IEEE STANDARDS ASSOCIATION IEEE

IEEE Standard for Information technology—
Telecommunications and information exchange between systems
Local and metropolitan area networks—
Specific requirements

Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications

IEEE Computer Society

Sponsored by the
LAN/MAN Standards Committee

IEEE
3 Park Avenue
New York, NY 10016-5997
USA

29 March 2012

IEEE Std 802.11™-2012
(Revision of
IEEE Std 802.11-2007)

IEEE Std 802.11™-2012
(Revision of
IEEE Std 802.11-2007)

IEEE Standard for Information technology— Telecommunications and information exchange between systems Local and metropolitan area networks— Specific requirements

Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications

Sponsor

**LAN/MAN Standards Committee
of the
IEEE Computer Society**

Approved 6 February 2012

IEEE-SA Standards Board

Abstract: This revision specifies technical corrections and clarifications to IEEE Std 802.11 for wireless local area networks (WLANS) as well as enhancements to the existing medium access control (MAC) and physical layer (PHY) functions. It also incorporates Amendments 1 to 10 published in 2008 to 2011.

Keywords: 2.4 GHz, 3650 MHz, 4.9 GHz, 5 GHz, 5.9 GHz, advanced encryption standard, AES, carrier sense multiple access/collision avoidance, CCMP, channel switching, Counter mode with Cipher-block chaining Message authentication code Protocol, confidentiality, CSMA/CA, DFS, direct link, dynamic frequency selection, E911, emergency alert system, emergency services, forwarding, generic advertisement service, high throughput, IEEE 802.11, interface, international roaming, interworking, interworking with external networks, LAN, local area network, MAC, measurement, medium access control, media-independent handover, medium access controller, mesh, MIH, MIMO, MIMO-OFDM, multi-hop, multiple input multiple output, network advertisement, network discovery, network management, network selection, off-channel direct link, path-selection, PHY, physical layer, power saving, QoS, quality of service, PHY, physical layer, QoS mapping, radio, radio frequency, RF, radio resource, radio management, SSP, SSPN, subscriber service provider, temporal key integrity protocol, TKIP, TPC, transmit power control, tunneled direct link setup, wireless access in vehicular environments, wireless LAN, wireless local area network, WLAN, wireless network management, zero-knowledge proof

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Print: **ISBN 978-0-7381-7211-8 STD97218**
PDF: **ISBN 978-0-7381-7245-3 STDPD97218**

Notice and Disclaimer of Liability Concerning the Use of IEEE Documents: IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

Use of an IEEE Standard is wholly voluntary. IEEE disclaims liability for any personal injury, property or other damage, of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, or reliance upon any IEEE Standard document.

IEEE does not warrant or represent the accuracy or content of the material contained in its standards, and expressly disclaims any express or implied warranty, including any implied warranty of merchantability or fitness for a specific purpose, or that the use of the material contained in its standards is free from patent infringement. IEEE Standards documents are supplied "**AS IS**."

The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE standard is subjected to review at least every ten years. When a document is more than ten years old and has not undergone a revision process, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE standard.

In publishing and making its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document, should rely upon his or her own independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

Translations: The IEEE consensus development process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English version published by IEEE should be considered the approved IEEE standard.

Official Statements: A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal position of IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position of IEEE.

Comments on Standards: Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE. However, IEEE does not provide consulting information or advice pertaining to IEEE Standards documents. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important to ensure that any responses to comments and questions also receive the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to comments or questions except in those cases where the matter has previously been addressed. Any person who would like to participate in evaluating comments or revisions to an IEEE standard is welcome to join the relevant IEEE working group at http://standards.ieee.org/develop/wg/.

Comments on standards should be submitted to the following address:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
Piscataway, NJ 08854
USA

Photocopies: Authorization to photocopy portions of any individual standard for internal or personal use is granted by The Institute of Electrical and Electronics Engineers, Inc., provided that the appropriate fee is paid to Copyright Clearance Center. To arrange for payment of licensing fee, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Notice to users

Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE standards document does not imply compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Copyrights

This document is copyrighted by the IEEE. It is made available for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making this document available for use and adoption by public authorities and private users, the IEEE does not waive any rights in copyright to this document.

Updating of IEEE documents

Users of IEEE standards should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect. In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit the IEEE-SA website or contact the IEEE at the address listed previously. For more information about the IEEE Standards Association or the IEEE standards development process, visit the IEEE-SA website.

Errata

Errata, if any, for this and all other standards can be accessed at the following URL: http://standards.ieee.org/findstds/errata/index.html. Users are encouraged to check this URL for errata periodically.

Patents

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. A patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE-SA website http://standards.ieee.org/about/sasb/patcom/patents.html. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or nondiscriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

Participants

At the time this revision was sent to sponsor ballot, the IEEE 802.11 Working Group had the following officers:

Bruce Kraemer, *Chair*
Jon Walter Rosdahl, *Vice-Chair and Treasurer*
Adrian P. Stephens, *Vice-Chair, Technical Editor and Assigned Number Authority*
Stephen McCann, *Secretary*
Peter Ecclesine, *Technical Editor*
Clint Chaplin, Chair, Wireless Next Generation Standing Committee
David Bagby, Chair, Architecture Standing Committee
Andrew Myles, Chair, JTC1 Ad hoc
Richard H. Kennedy, Chair, Regulatory Ad hoc and Task Group af
Hiroshi Mano, Chair, FIA Study Group
Dorothy Stanley, Chair, Task Group mb and Task Group v
Dee Denteneer, Chair, Task Group s
Menzo M. Wentink, Chair, Task Group z
Ganesh Venkatesan, Chair, Task Group aa
Osama S. Aboul-Magd, Chair, Task Group ac
Eldad Perahia, Chair, Task Group ad
Michael Montemurro, Chair, Task Group ae
Dave Halasz, Chair, Task Group ah

The officers and members of the Task Group mb Working Group ballot pool are as follows:

Matthew S. Gast, *Chair* (to March 2010)
Dorothy Stanley, *Chair* (from May 2010)
Michael Montemurro, *Vice Chair*
Jon Walter Rosdahl, *Secretary*
Adrian P. Stephens, *Editor*

Santosh P. Abraham
Tomoko Adachi
Alok Aggarwal
Carlos H. Aldana
Gary Anwyl
Lee R. Armstrong
Alex Ashley
Malik Audeh
Geert A. Awater
Michael Bahr
Fan Bai
Gabor Bajko
John R. Barr
Gal Basson
Tuncer Baykas
John L. Benko
Mathilde Benveniste
Daniel Borges
Anthony Braskich
Joseph Brennan
George Bumiller
Daniel Camps-Mur
Nancy Cam-Winget
Necati Canpolat
Javier Cardona
Philippe Chambelin
Douglas S. Chan
Jiunn-Tsair Chen
Lidong Chen
Minho Cheong
Woong Cho
Jee-Yon Choi
Nakjung Choi
Liwen Chu
Terry L. Cole
Charles I. Cook
Xavier P. Costa
David E. Cypher
Marc De Courville
Rolf J. deVegt
Jeremy deVries
Susan Dickey
Yoshiharu Doi
John Dorsey
Roger P. Durand
Srinivasa Duvvuri
Donald E. Eastlake
Michael Ellis
Stephen P. Emeott
Marc Emmelmann
Darwin Engwer
Vinko Erceg
Robert Fanfelle
Stefan Fechtel
Matthew J. Fischer
Wayne K. Fisher
Ryuhei Funada
James P. Gilb
Jeffrey Gilbert
Reinhard Gloger
Michelle Gong
David Goodall
Mark Grodzinsky
Jianlin Guo
Mark Hamilton
C. J. Hansen
Hiroshi Harada
Dan N. Harkins
Brian D. Hart
Amer A. Hassan
Vegard Hassel
Robert F. Heile
Guido R. Hiertz
Naoki Honma
Wendong Hu

Robert Y. Huang
Akio Iso
Wynona Jacobs
Junghoon Jee
Hongseok Jeon
Yeonkwon Jeong
Jorjeta G. Jetcheva
Lusheng Ji
Daniel Jiang
Padam Kafle
Carl W. Kain
Naveen K. Kakani
Masato Kato
Shuzo Kato
Douglas Kavner
John Kenney
Stuart J. Kerry
Joonsuk Kim
Kyeongpyo Kim
Yongsun Kim
Yunjoo Kim
Jarkko Kneckt
Mark M. Kobayashi
Fumihide Kojima
Tom Kolze
Johannes P. Kruys
Thomas Kuehnel
Thomas M. Kurihara
Joseph Kwak
Edwin Kwon
Hyoungjin Kwon
Ismail Lakkis
Paul Lambert
Zhou Lan
Jeremy A. Landt
Joseph P. Lauer
Tae H. Lee
Wooyong Lee
Yuro Lee
Sheung Li
Hang Liu
Michael Livshitz
Peter Loc
Daniel Lubar
Jakub Majkowski
Alastair Malarky
Jouni K. Malinen
Alexander Maltsev
Bill Marshall
Roman M. Maslennikov
Justin P. McNew

Sven Mesecke
Robert R. Miller
Rajendra T. Moorti
Hitoshi Morioka
Yuichi Morioka
Peter Murray
Yukimasa Nagai
Kengo Nagata
Hiroki Nakano
Chiu Ngo
Paul Nikolich
Eero Nikula
Richard H. Noens
Jisung Oh
Jong-Ee Oh
Chandra S. Olson
Youko Omori
Satoshi Oyama
Richard H. Paine
Arul D. Palanivelu
Changmin Park
Minyoung Park
Vijaykumar Patel
Bemini H. Peiris
James E. Petranovich
Albert Petrick
James D. Portaro
Henry S. Ptasinski
Rene Purnadi
Emily H. Qi
Luke Qian
Huyu Qu
Jim E. Raab
Mohammad Rahman
Vinuth Rai
Ali Raissinia
Harish Ramamurthy
Stephen G. Rayment
Ivan Reede
Alex Reznik
Randal Roebuck
Richard Roy
Alexander Safonov
Kazuyuki Sakoda
Hemanth Sampath
Katsuyoshi Sato
Hirokazu Sawada
Donald Schultz
Yongho Seok
Huairong Shao
Neeraj Sharma

Stephen J. Shellhammer
Ian Sherlock
Kai Shi
Shusaku Shimada
Francois Simon
Graham K. Smith
Matt Smith
Yoo-Seung Song
Kapil Sood
Vinay Sridhara
Robert Stacey
David S. Stephenson
Carl R. Stevenson
John Stine
Guenael T. Strutt
Chin-Sean Sum
Arash Tabibiazar
Eiji Takagi
Mineo Takai
Teik-Kheong Tan
Allan Thomson
Jerry Thrasher
Eric Tokubo
Ichihiko Toyoda
Jason Trachewsky
Solomon B. Trainin
Richard D. Van Nee
Allert Van Zelst
Mathieu Varlet-Andre
Prabodh Varshney
Dalton T. Victor
George A. Vlantis
Jesse R. Walker
Junyi Wang
Qi Wang
Craig D. Warren
Fujio Watanabe
Patrick Waye
Frank Whetten
James Worsham
Harry R. Worstell
Takeshi Yamamoto
James Yee
Peter Yee
Su K. Yong
Seiji Yoshida
Christopher Young
Artur Zaks
Hongyuan Zhang
Huimin Zhang
Shiwei Zhao

Major contributions were received from the following individuals:

Peter Ecclesine
Matthew S. Gast
Michelle Gong
Mark Hamilton
Dan Harkins

Bill Marshall
Michael Montemurro
Henry Ptasinski
Mark Rison
Jon Walter Rosdahl
Ashish Shukla

Robert Stacey
Dorothy Stanley
Adrian P. Stephens
Dave Stephenson
Shi Yang

The following members of the individual balloting committee voted on this revision. Balloters may have voted for approval, disapproval, or abstention.

Tomoko Adachi
Roberto Aiello
Thomas Alexander
Richard Alfvin
Mark Anderson
Peter Anslow
Lee Armstrong
Torrey Atcitty
Tuncer Baykas
Harry Bims
Gennaro Boggia
Nancy Bravin
William Byrd
Ruben Salazar Cardozo
James Carlo
Yi-Ming Chen
Keith Chow
Charles Cook
Theodorus Denteneer
Wael Diab
Patrick Diamond
Russell Dietz
Thomas Dineen
Roger Durand
Sourav Dutta
Donald Eastlake
Peter Ecclesine
Richard Eckard
Marc Emmelmann
Matthew Fischer
Prince Francis
Avraham Freedman
Devon Gayle
Pieter-Paul Giesberts
James Gilb
Stephen Glass
Reinhard Gloger
Tim Godfrey
Patrick Gonia
Sudheer Grandhi
Randall Groves
Michael Gundlach
C. Guy
Rainer Hach
David Halasz
Mark Hamilton
Christopher Hansen
Marco Hernandez
Guido Hiertz
Ronald Hochnadel
Oliver Hoffmann
David Hunter
Yasuhiko Inoue
Sergiu Iordanescu
Akio Iso

Atsushi Ito
Raj Jain
Junghoon Jee
Vincent Jones
Bobby Jose
Tal Kaitz
Naveen Kakani
Shinkyo Kaku
Masahiko Kaneko
Tae-Gyu Kang
Piotr Karocki
John Kenney
Stuart J. Kerry
Yongbum Kim
Youhan Kim
Patrick Kinney
Bruce Kraemer
Thomas Kurihara
David Landry
Jeremy Landt
Michael Lerer
Daniel Levesque
Jan-Ray Liao
Arthur Light
Ru Lin
Lu Liru
William Lumpkins
Greg Luri
Bradley Lynch
Chris Lyttle
Elvis Maculuba
Alastair Malarky
Jouni Malinen
Roger Marks
Jeffery Masters
Stephen McCann
Michael McInnis
Justin McNew
Steven Methley
David Mitton
Emmanuel Monnerie
Michael Montemurro
Matthew Mora
Jose Morales
Ronald Murias
Rick Murphy
Peter Murray
Andrew Myles
Michael S. Newman
Charles Ngethe
Paul Nikolich
Kevin Noll
Satoshi Obara
Knut Odman

Robert O’Hara
Chris Osterloh
Satoshi Oyama
Glenn Parsons
Eldad Perahia
James Petranovich
Venkatesha Prasad
Michael Probasco
Henry Ptasinski
Sridhar Rajagopal
Jayaram Ramasastry
Maximilian Riegel
Robert Robinson
Randal Roebuck
Jon Walter Rosdahl
Herbert Ruck
Randall Safier
Kazuyuki Sakoda
Naotaka Sato
Bartien Sayogo
Cristina Seibert
Rich Seifert
Yang Shi
Shusaku Shimada
Gil Shultz
Di Dieter Smely
Jae-Hyung Song
Kapil Sood
Amjad Soomro
Manikantan Srinivasan
Dorothy Stanley
Kenneth Stanwood
Thomas Starai
Adrian Stephens
Rene Struik
Walter Struppler
Mark Sturza
Bo Sun
Jun Ichi Takada
David Thompson
Solomon Trainin
Mark-Rene Uchida
Prabodh Varshney
Bhupender Virk
George Vlantis
Stanley Wang
Stephen Webb
Hung-Yu Wei
Menzo Wentink
James Worsham
Harry Worstell
Forrest Wright
Tan Pek Yew
Oren Yuen
Janusz Zalewski

When the IEEE-SA Standards Board approved this revision on 6 February 2012, it had the following membership:

Richard H. Hulett, *Chair*
John Kulick, *Vice Chair*
Robert M. Grow, *Past President*
Judith Gorman, *Secretary*

Masayuki Ariyoshi
William Bartley
Ted Burse
Clint Chaplin
Wael Diab
Jean-Philippe Faure
Alexander Gelman
Paul Houzé
Jim Hughes
Joseph L. Koepfinger*
David J. Law
Thomas Lee
Hung Ling
Oleg Logvinov
Ted Olsen
Gary Robinson
Jon Walter Rosdahl
Sam Sciacca
Mike Seavey
Curtis Siller
Phil Winston
Howard Wolfman
Don Wright

*Member Emeritus

Also included are the following nonvoting IEEE-SA Standards Board liaisons:

Satish K. Aggarwal, *NRC Representative*
Richard DeBlasio, *DOE Representative*
Michael H. Kelly, *NIST Representative*

Michelle D. Turner
IEEE Standards Program Manager, Document Development

Patricia Gerdon
IEEE Standards Program Managers, Technical Program Development

Introduction

This introduction is not part of IEEE Std 802.11-2012, IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area network—Specific requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications.

This revision gives users, in one document, the IEEE 802.11 standard for wireless local area networks (WLANS) with all the amendments that have been published to date.

Incorporating published amendments

The original standard was published in 1999 and reaffirmed in 2003. A revision was published in 2007, which incorporated into the 1999 edition the following amendments: IEEE Std 802.11a™-1999, IEEE Std 802.11b™-1999, IEEE Std 802.11b-1999/Corrigendum 1-2001, IEEE Std 802.11d™-2001, IEEE Std 802.11g™-2003, IEEE Std 802.11h™-2003, IEEE Std 802.11i™-2004, IEEE Std 802.11j™-2004 and IEEE Std 802.11e™-2005.

The current revision, IEEE Std 802.11-2012, incorporates the following amendments into the 2007 revision:

— IEEE Std 802.11k™-2008: Radio Resource Measurement of Wireless LANs (Amendment 1)

— IEEE Std 802.11r™-2008: Fast Basic Service Set (BSS) Transition (Amendment 2)

— IEEE Std 802.11y™-2008: 3650–3700 MHz Operation in USA (Amendment 3)

— IEEE Std 802.11w™-2009: Protected Management Frames (Amendment 4)

— IEEE Std 802.11n™-2009: Enhancements for Higher Throughput (Amendment 5)

— IEEE Std 802.11p™-2010: Wireless Access in Vehicular Environments (Amendment 6)

— IEEE Std 802.11z™-2010: Extensions to Direct-Link Setup (DLS) (Amendment 7)

— IEEE Std 802.11v™-2011: IEEE 802.11 Wireless Network Management (Amendment 8)

— IEEE Std 802.11u™-2011: Interworking with External Networks (Amendment 9)

— IEEE Std 802.11s™-2011: Mesh Networking (Amendment 10)

As a result of publishing this revision, all of the previously published amendments and revisions are now retired.

Technical corrections, clarifications, and enhancements

In addition, this revision specifies technical corrections and clarifications to IEEE Std 802.11 as well as enhancements to the existing medium access control (MAC) and physical layer (PHY) functions. Such enhancements include incorporated interpretation requests.

Revised clause and annex numbering

In IEEE Std 802.11-2012, the order of clauses and annexes has also been revised. The result of this revised order on the numbering of clauses and annexes is summarized in Figure A.

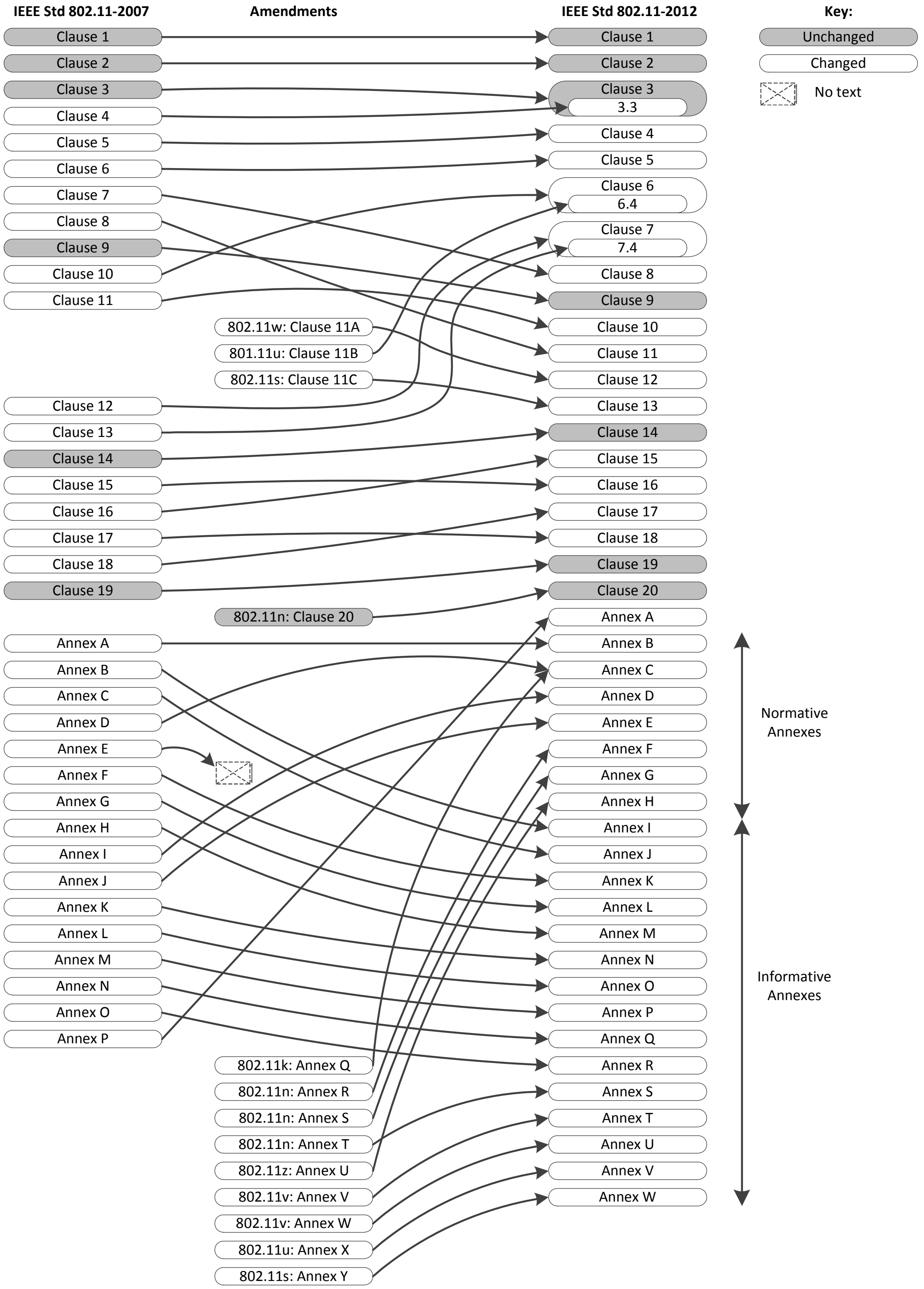


Figure A—Changes in clause numbers and annex letters from 2007 revision to 2012 revision

Contents

Tables

Figures

IEEE Standard for Information technology— Telecommunications and information exchange between systems Local and metropolitan area networks— Specific requirements

Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications

IMPORTANT NOTICE: This standard is not intended to assure safety, security, health, or environmental protection. Implementers of the standard are responsible for determining appropriate safety, security, environmental, and health practices or regulatory requirements.

This IEEE document is made available for use subject to important notices and legal disclaimers. These notices and disclaimers appear in all publications containing this document and may be found under the heading "Important Notice" or "Important Notices and Disclaimers Concerning IEEE Documents." They can also be obtained on request from IEEE or viewed at http://standards.ieee.org/IPR/disclaimers.html.

1. Overview

1.1 Scope

The scope of this standard is to define one medium access control (MAC) and several physical layer (PHY) specifications for wireless connectivity for fixed, portable, and moving stations (STAs) within a local area.

1.2 Purpose

The purpose of this standard is to provide wireless connectivity for fixed, portable, and moving stations within a local area. This standard also offers regulatory bodies a means of standardizing access to one or more frequency bands for the purpose of local area communication.

1.3 Supplementary information on purpose

Specifically, this standard

— Describes the functions and services required by an IEEE 802.11™-compliant device to operate within independent and infrastructure networks as well as the aspects of STA mobility (transition) within those networks.

— Describes the functions and services that allow an IEEE 802.11-compliant device to communicate directly with another such device outside of an independent or infrastructure network.

- Defines the MAC procedures to support the MAC service data unit (MSDU) delivery services.
- Defines several PHY signaling techniques and interface functions that are controlled by the IEEE 802.11 MAC.
- Permits the operation of an IEEE 802.11-conformant device within a wireless local area network (WLAN) that may coexist with multiple overlapping IEEE 802.11 WLANs.
- Describes the requirements and procedures to provide data confidentiality of user information and MAC management information being transferred over the wireless medium (WM) and authentication of IEEE 802.11-conformant devices.
- Defines mechanisms for dynamic frequency selection (DFS) and transmit power control (TPC) that may be used to satisfy regulatory requirements for operation in any band.
- Defines the MAC procedures to support local area network (LAN) applications with quality-of-service (QoS) requirements, including the transport of voice, audio, and video.
- Defines mechanisms and services for wireless network management of STAs that include BSS transition management, channel usage and coexistence, collocated interference reporting, diagnostic, multicast diagnostic and event reporting, flexible multicast, efficient beacon mechanisms, proxy ARP advertisement, location, timing measurement, directed multicast, extended sleep modes, traffic filtering, and management notification.
- Defines functions and procedures aiding network discovery and selection by STAs, information transfer from external networks using QoS mapping, and a general mechanism for the provision of emergency services.
- Defines the MAC procedures that are necessary for wireless multi-hop communication to support wireless LAN mesh topologies.

1.4 Word Usage

In this document, the word *shall* is used to indicate a mandatory requirement. The word *should* is used to indicate a recommendation. The word *may* is used to indicate a permissible action. The word *can* is used for statements of possibility and capability.

2. Normative references

The following referenced documents are indispensable for the application of this standard (i.e., they must be understood and used; therefore, each referenced document is cited in the text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

3GPP TS 24.234, 3GPP System to Wireless Local Area Network (WLAN) interworking; WLAN User Equipment (WLAN UE) to network protocols; Stage 3.[1]

ETSI EN 301 893, Broadband Radio Access Networks (BRAN); 5 GHz high performance RLAN; Part 2: Harmonized EN covering essential requirements of article 3.2 of the R&TTE Directive.[2]

FIPS PUB 180-3-2008, Secure Hash Standard.[3]

FIPS PUB 197-2001, Advanced Encryption Standard (AES).

[1] 3GPP™ documents are available from the 3rd Generation Partnership Project Web site (http://www.3gpp.org).

[2] ETSI documents are available from the European Telecommunications Standards Institute (http://www.etsi.org).

[3] FIPS publications are available from the National Technical Information Service (NTIS) (http://www.ntis.org/).

FIPS SP800-38B, “Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication, Dworkin, M.”

IANA EAP Method Type Numbers, http://www.iana.org/assignments/eap-numbers.

IEEE Std 754™-2008, IEEE Standard for Binary Floating-Point Arithmetic.[4,5]

IEEE Std 802®-2001, IEEE Standards for Local and Metropolitan Area Networks: Overview and Architecture.

IEEE Std 802.1AS™, IEEE Standard for Local and Metropolitan Area Networks—Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks.

IEEE Std 802.1X™-2004, IEEE Standard for Local and Metropolitan Area Networks: Port-Based Network Access Control.

IEEE Std 802.21™-2008, IEEE Standard for Local and Metropolitan Area Networks: Media Independent Handover Services.

IEEE Std C95.1™, IEEE Standard Safety Levels with Respect to Human Exposure to Radio Frequency Electromagnetic Fields, 3 kHz to 300 GHz.

IETF RFC 791, Internet Protocol, Sept. 1981.[6]

IETF RFC 925, Multi-LAN Address Resolution, J. Postel, Oct. 1984.

IETF RFC 1035, Domain Names — Implementation and Specification, P. Mockapetris, Nov. 1987.

IETF RFC 1042, A Standard for the Transmission of IP Datagrams over IEEE 802® Networks, J. Postel, J. Reynolds, Feb. 1988.

IETF RFC 1321, The MD5 Message-Digest Algorithm, Apr. 1992 (status: informational).

IETF RFC 2104, HMAC: Keyed-Hashing for Message Authentication, H. Krawczyk, M. Bellare, R. Canetti, Feb. 1997 (status: informational).

IETF RFC 2409, The Internet Key Exchange (IKE), D. Harkins, D. Carrel, Nov. 1998 (status: Standards Track).

IETF RFC 2460, Internet Protocol, Version 6 (IPv6), S. Deering, R. Hinden, Dec. 1998.

IETF RFC 3164, The BSD Syslog Protocol, Aug. 2001.

IETF RFC 3394, Advanced Encryption Standard (AES) Key Wrap Algorithm, J. Schaad, R. Housley, Sept. 2002 (status: informational).

IETF RFC 3610, Counter with CBC-MAC (CCM), D. Whiting, R. Housley, N. Ferguson, Sept. 2003 (status: informational).

[4] The IEEE standards or products referred to in this clause are trademarks owned by The Institute of Electrical and Electronics Engineers, Inc.

[5] IEEE publications are available from The Institute of Electrical and Electronics Engineers (http://standards.ieee.org/).

[6] IETF documents (i.e., RFCs) are available for download at http://www.rfc-archive.org/.

IETF RFC 3629, UTF-8, a transformation format of ISO 10646, F. Yergeau, Nov. 2003.

IETF RFC 3748, Extensible Authentication Protocol (EAP), B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, H. Levkowetz, June 2004.

IETF RFC 3825, Dynamic Host Configuration Protocol Option for Coordinate-based Location Configuration Information, Polk, J., Schnizlein, J., Linsner, M., July 2004.

IETF RFC 3986, Uniform Resource Identifier (URI): Generic Syntax, Jan. 2005.

IETF RFC 4017, Extensible Authentication Protocol (EAP) Method Requirements for Wireless LANs, D. Stanley, J. Walker, B. Aboba, Mar. 2005 (status: informational).

IETF RFC 4282, The Network Access Identifier, Dec. 2005.

IETF RFC 4776, Dynamic Host Configuration Protocol (DHCPv4 and DHCPv6) Option for Civic Addresses Configuration Information, Nov. 2006.

IETF RFC 4861, Neighbor Discovery for IP version 6 (IPv6), T. Narten, E. Nordmark, W. Simpson, H. Soliman, Sept. 2007.

IETF RFC 5216, The EAP-TLS Authentication Protocol, D. Simon, B. Aboba, R. Hurst, March 2008.

IETF RFC 5297, Synthetic Initialization Vector (SIV) Authenticated Encryption Using the Advanced Encryption Standard (AES), D. Harkins, October 2008 (status: informational).

ISO/IEC 3166-1, Codes for the representation of names of countries and their subdivisions—Part 1: Country codes.[7]

ISO/IEC 7498-1:1994, Information technology—Open Systems Interconnection—Basic Reference Model: The Basic Model.

ISO/IEC 8802-2:1998, Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 2: Logical link control.

ISO/IEC 8824-1:1995, Information technology—Abstract Syntax Notation One (ASN.1): Specification of basic notation.

ISO/IEC 8824-2:1995, Information technology—Abstract Syntax Notation One (ASN.1): Information object specification.

ISO/IEC 8824-3:1995, Information technology—Abstract Syntax Notation One (ASN.1): Constraint specification.

ISO/IEC 8824-4:1995, Information technology—Abstract Syntax Notation One (ASN.1): Parameterization of ASN.1 specifications.

ISO/IEC 8825-1:1995, Information technology—ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER).

[7] ISO/IEC publications are available from the ISO Central Secretariat (http://www.iso.ch/). ISO/IEC publications are also available in the United States from the American National Standards Institute (http://www.ansi.org/).

ISO/IEC 8825-2:1996, Information technology—ASN.1 encoding rules: Specification of Packed Encoding Rules (PER).

ISO/IEC 11802-5:1997(E), Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Technical reports and guidelines—Part 5: Medium Access Control (MAC) Bridging of Ethernet V2.0 in Local Area Networks (previously known as IEEE Std 802.1H-1997 [B21][8]).

ISO/IEC 15802-1:1995, Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Common specifications—Part 1: Medium Access Control (MAC) service definition.

ISO/IEC 15802-3, Information Technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Common specifications—Part 3: Media Access Control (MAC) Bridges.

ITU-R Recommendation TF.460-4(2002), Standard-frequency and time-signal emissions.[9]

ITU-T Recommendation Z.100 (03/93), CCITT specification and description language (SDL).

ITU-T Recommendation Z.105 (03/95), SDL combined with ASN.1 (SDL/ASN.1).

ITU-T Recommendation Z.120 (2004), Programming Languages—Formal Description Techniques (FDT)—Message Sequence Chart (MSC).

OASIS Emergency Management Technical Committee, "Emergency Data Exchange Language (EDXL) Distribution Element, v. 1.0." OASIS Standard EDXL-DE v1.0, May 2006.

3. Definitions, acronyms, and abbreviations

3.1 Definitions

For the purposes of this standard, the following terms and definitions apply. The *IEEE Standards Dictionary: Glossary of Terms & Definitions* should be referenced for terms not defined in this clause.[10]

access category (AC): A label for the common set of enhanced distributed channel access (EDCA) parameters that are used by a quality-of-service (QoS) station (STA) to contend for the channel in order to transmit medium access control (MAC) service data units (MSDUs) with certain priorities.

access control: The prevention of unauthorized usage of resources.

access point (AP): An entity that contains one station (STA) and provides access to the distribution services, via the wireless medium (WM) for associated STAs.

access point (AP) path: Path between two tunneled direct-link setup (TDLS) peer stations (STAs) via the AP with which the STAs are currently associated.

[8] The numbers in brackets correspond to the numbers of the bibliography in Annex A.

[9] ITU publications are available from the International Telecommunications Union (http://www.itu.int/).

[10] The *IEEE Standards Dictionary: Glossary of Terms & Definitions* is available at http://shop.ieee.org/.

access point (AP) reachability: An AP is reachable by a station (STA) if preauthentication messages can be exchanged between the STA and the target AP via the distribution system (DS).

NOTE—Preauthentication is defined in 11.5.9.2.[11]

active mode: A mesh power mode in which the mesh station (STA) operates in the Awake state towards a neighbor mesh STA.

additional authentication data (AAD): Data that are not encrypted, but are cryptographically protected.

ad hoc network: Often used as a venacular term for an independent basic service set (IBSS).

admission control: An algorithm to ensure that admittance of a new flow into a resource constrained network does not violate parameterized service commitments made by the network to admitted flows.

aggregate medium access control (MAC) protocol data unit (A-MPDU): A structure containing multiple MPDUs, transported as a single physical layer convergence procedure (PLCP) service data unit (PSDU) by the physical layer (PHY).

aggregate medium access control (MAC) service data unit (A-MSDU): A structure containing multiple MSDUs, transported within a single (unfragmented) data medium access control (MAC) protocol data unit (MPDU).

aggregate medium access control (MAC) service data unit (A-MSDU) subframe: A portion of an A-MSDU containing a header and associated MSDU.

aggregated schedule: The aggregation of delivery and/or poll schedules by the quality-of-service (QoS) access point (AP) for a particular QoS station (STA) into a single service period (SP).

antenna connector: The measurement point of reference for radio frequency (RF) measurements in a station (STA). The antenna connector is the point in the STA architecture representing the input of the receiver (output of the antenna) for radio reception and the input of the antenna (output of the transmitter) for radio transmission. In systems using multiple antennas or antenna arrays, the antenna connector is a virtual point representing the aggregate output of (or input to) the multiple antennas. In systems using active antenna arrays with processing, the antenna connector is the output of the active array, which includes any processing gain of the active antenna subsystem.

antenna selection (ASEL) receiver: A station (STA) that performs receive ASEL.

antenna selection (ASEL) transmitter: A station (STA) that performs transmit ASEL.

association: The service used to establish access point/station (AP/STA) mapping and enable STA invocation of the distribution system services (DSSs).

authentication: The service used to establish the identity of one station (STA) as a member of the set of STAs authorized to associate with another STA.

authentication and key management (AKM) suite: A set of one or more algorithms designed to provide authentication and key management, either individually or in combination with higher layer authentication and key management algorithms outside the scope of this standard.

[11] Notes in text, tables, and figures of a standard are given for information only and do not contain requirements needed to implement the standard.

Authentication Server (AS): An entity that provides an authentication service to an Authenticator. This service determines, from the credentials provided by the Supplicant, whether the Supplicant is authorized to access the services provided by the Authenticator. (IEEE Std 802.1X-2004[12])

Authenticator: An entity at one end of a point-to-point LAN segment that facilitates authentication of the entity attached to the other end of that link. (IEEE Std 802.1X-2004)

Authenticator address (AA): The medium access control (MAC) address of the IEEE 802.1X Authenticator.

authorization: The act of determining whether a particular right, such as access to a resource, is granted to an entity.
NOTE—See IETF RFC 2903 [B32].[13]

authorized: To be explicitly allowed.

average noise power indicator (ANPI): A medium access control (MAC) indication of the average noise plus interference power measured when the channel is idle as defined by three simultaneous conditions: 1) the Virtual Carrier Sense (CS) mechanism indicates idle channel, 2) the station (STA) is not transmitting a frame, and 3) the STA is not receiving a frame.

azimuth: The horizontal orientation of the front surface of a station or of a radio antenna system’s main lobe measured clockwise from true north.

base channel: Channel on which the tunneled direct-link setup (TDLS) peer station (STA) is associated with an access point (AP).

basic service area (BSA): The area containing the members of a basic service set (BSS). It might contain members of other BSSs.

basic service set (BSS): A set of stations (STAs) that have successfully synchronized using the JOIN service primitives[14] and one STA that has used the START primitive. Alternatively, a set of STAs that have used the START primitive specifying matching mesh profiles where the match of the mesh profiles has been verified via the scanning procedure. Membership in a BSS does not imply that wireless communication with all other members of the BSS is possible.

basic service set (BSS) transition: A station (STA) movement from one BSS to another BSS in the same extended service set (ESS).

beamformee: A station (STA) that receives a physical layer convergence procedure (PLCP) protocol data unit (PPDU) that was transmitted using a beamforming steering matrix.

beamformer: A station (STA) that transmits a physical layer convergence procedure (PLCP) protocol data unit (PPDU) using a beamforming steering matrix.

beamforming: A spatial filtering mechanism used at a transmitter to improve the received signal power or signal-to-noise ratio (SNR) at an intended receiver. *Syn:* **beam steering**.

big endian: The concept that, for a given multi-octet numeric representation, the most significant octet has the lowest address.

[12] Information on references can be found in Clause 2.

[13] The numbers in brackets correspond to the numbers of the bibliography in Annex A.

[14] Description of these primitives can be found in 6.3.4.

broadcast address: A unique group address that specifies all stations (STAs).

BSS Max idle period: A time period during which the access point (AP) does not disassociate a station (STA) due to nonreceipt of frames from that STA.

calibration initiator: A station (STA) that initiates a calibration sequence.

calibration responder: A station (STA) that transmits during a calibration sequence in response to a transmission by a calibration initiator.

candidate peer mesh station (STA): A neighbor mesh STA to which a mesh peering has not been established but meets eligibility requirements to become a peer mesh STA.

channel: An instance of communications medium use for the purpose of passing protocol data units (PDUs) between two or more stations (STAs).

channel spacing: The difference between the center frequencies of two nonoverlapping and adjacent channels of the radio transmitter.

cipher suite: A set of one or more algorithms, designed to provide data confidentiality, data authenticity or integrity, and/or replay protection.

clear channel assessment (CCA) function: That logical function in the physical layer (PHY) that determines the current state of use of the wireless medium (WM).

collocated interference: Interference that is caused by another radio or station (STA) emitting radio energy located in the same physical device as the reporting STA, where the reported characteristics of the interference are known a priori without interference detection, measurement, or characterization by the reporting STA.

collocated radio: A radio capable of emitting radio-frequency energy located in the same physical device as the reporting station (STA), where the radio’s type and some link characteristics are known without signal detection or measurement by the reporting STA.

configuration profile: A collection of parameters identified by a profile identifier (ID) that represent a current or available configuration of a station (STA).

contention-free period (CFP): The time period during the operation of a point coordination function (PCF) when the right to transmit is assigned to stations (STAs) solely by a point coordinator (PC), allowing frame exchanges to occur between members of the basic service set (BSS) without contention for the wireless medium (WM).

contention period (CP): The time period outside of the contention-free period (CFP) in a point-coordinated basic service set (BSS). In a BSS where there is no point coordinator (PC), this corresponds to the entire time of operation of the BSS.

controlled access phase (CAP): A time period when the hybrid coordinator (HC) maintains control of the medium, after gaining medium access by sensing the channel to be idle for a point coordination function (PCF) interframe space (PIFS) duration. It might span multiple consecutive transmission opportunities (TXOPs) and can contain polled TXOPs.

coordination function: The logical function that determines when a station (STA) operating within a basic service set (BSS) is permitted to transmit protocol data units (PDUs) via the wireless medium (WM). The coordination function within a BSS might have one hybrid coordination function (HCF), or it might have one HCF and one point coordination function (PCF) and has one distributed coordination function (DCF). A quality-of-service (QoS) BSS has one DCF and one HCF.

contention-free (CF) pollable: A station (STA) that is able to respond to a CF poll with a data frame if such a frame is queued and able to be generated.

Counter mode with Cipher-block chaining Message authentication code (CCM): A symmetric key block cipher mode providing confidentiality using counter mode (CTR) and data origin authenticity using cipher-block chaining message authentication code (CBC-MAC).

NOTE—See IETF RFC 3610.

cryptographic encapsulation: The process of generating the cryptographic payload from the plaintext data. This comprises the cipher text as well as any associated cryptographic state required by the receiver of the data, e.g., initialization vectors (IVs), sequence numbers, message integrity codes (MICs), key identifiers.

data confidentiality: A property of information that prevents disclosure to unauthorized individuals, entities, or processes.

deauthentication service: The service that voids an existing authentication relationship.

decapsulate: To recover an unprotected frame from a protected one.

decapsulation: The process of generating plaintext data by decapsulating an encapsulated frame.

deep sleep mode: A mesh power mode in which the mesh station (STA) operates either in the Awake state or in the Doze state towards a neighbor mesh STA, and is not expected to receive beacons from this neighbor mesh STA.

delivery-enabled access category (AC): A quality-of-service (QoS) access point (AP) AC where the AP is allowed to use enhanced distributed channel access (EDCA) to deliver traffic from the AC to a QoS station (STA) in an unscheduled service period (SP) triggered by the STA.

dependent station (STA): A STA that is not registered and whose operational parameters are dictated by messages it receives from an enabling STA. Once enabled by the dynamic STA enablement (DSE) process, a dependent STA's continued operation becomes contingent upon being able to receive messages from its enabling STA over the air.

destination mesh station (STA): A mesh STA that is the final destination of a MAC service data unit (MSDU). This mesh STA might reside in a proxy mesh gate that might forward the MSDU to a STA outside of the MBSS. A destination mesh STA might be an end station as defined in IEEE Std 802.1.

directed frame: *See:* **individually addressed**.

direct link: A bidirectional link from one quality-of-service (QoS) station (STA) to another QoS STA operating in the same infrastructure QoS basic service set (BSS) that does not pass through a QoS access point (AP). Once a direct link has been set up, all frames between the two QoS STAs are exchanged directly.

directed multicast service (DMS): A service in which the access point (AP) transmits group addressed frames as individually addressed frames to the requesting non-AP station (STA).

disassociation service: The service that removes an existing association.

distributed coordination function (DCF): A class of coordination function where the same coordination function logic is active in every station (STA) in the basic service set (BSS) whenever the network is in operation.

distribution service: The service that, by using association information, delivers medium access control (MAC) service data units (MSDUs) within the distribution system (DS).

distribution system (DS): A system used to interconnect a set of basic service sets (BSSs) and integrated local area networks (LANs) to create an extended service set (ESS).

distribution system medium (DSM): The medium or set of media used by a distribution system (DS) for communications between access points (APs), mesh gates, and portals of an extended service set (ESS).

distribution system service (DSS): The set of services provided by the distribution system (DS) that enable the medium access control (MAC) to transport MAC service data units (MSDUs) between stations (STAs) that are not in direct communication with each other over a single instance of the wireless medium (WM). These services include transport of MSDUs between the access points (APs) of basic service sets (BSSs) within an extended service set (ESS), transport of MSDUs between portals and BSSs within an ESS, transport of MSDUs between mesh gates in the same or different mesh basic service sets (MBSSs), transport of MSDUs between mesh gates and APs, transport of MSDUs between mesh gates and portals, and transport of MSDUs between STAs in the same BSS in cases where the MSDU has a group destination address or where the destination is an individual address and the STA is associated with an AP.
NOTE—DSSs are provided between pairs of MACs not on the same instance of the WM.

downlink: A unidirectional link from an access point (AP) to one or more non-AP stations (STAs).

dynamic frequency selection (DFS): Facilities mandated to satisfy requirements in some regulatory domains for radar detection and uniform channel spreading in the 5 GHz band. These facilities might also be used for other purposes, such as automatic frequency planning.

dynamic frequency selection (DFS) owner: A station (STA) in an independent basic service set (IBSS) or off-channel TDLS direct link that takes responsibility for selecting the next channel after radar is detected operating in a channel. Due to the nature of IBSSs, it cannot be guaranteed that there is a single DFS owner at any particular time and the protocol is robust to this situation.

dynamic station (STA) enablement (DSE): The process by which an enabling STA grants permission and dictates operational procedures to STAs that are subject to its control.

effective isotropic radiated power (EIRP): The equivalent power of a transmitted signal in terms of an isotropic (omnidirectional) radiator. The EIRP equals the product of the transmitter power and the antenna gain (reduced by any coupling losses between the transmitter and antenna).

emergency alert system (EAS): A U.S. national public warning system.

enabling station (STA): A registered STA that has the authority to control when and how a dependent STA can operate. An enabling STA communicates an enabling signal to its dependents over the air. An enabling STA chooses whether other dynamic STA enablement (DSE) messages are exchanged over the air, over the distribution system (DS), or by mechanisms that rely on transport via higher layers.

encapsulate: To construct a protected frame from an unprotected frame.

encapsulation: The process of generating a protected frame by encapsulating plaintext data.

enhanced distributed channel access (EDCA): The prioritized carrier sense multiple access with collision avoidance (CSMA/CA) access mechanism used by quality-of-service (QoS) stations (STAs) in a QoS basic service set (BSS). This access mechanism is also used by the QoS access point (AP) and operates concurrently with hybrid coordination function (HCF) controlled channel access (HCCA).

enhanced distributed channel access function (EDCAF): A logical function in a quality-of-service (QoS) station (STA) that determines, using enhanced distributed channel access (EDCA), when a frame in the transmit queue with the associated access category (AC) is permitted to be transmitted via the wireless medium (WM). There is one EDCAF per AC.

extended service area (ESA): The area within which members of an extended service set (ESS) can communicate. An ESA is larger than or equal to a basic service area (BSA) and might involve several basic service sets (BSSs) in overlapping, disjointed, or both configurations.

extended service set (ESS): A set of one or more interconnected basic service sets (BSSs) that appears as a single BSS to the logical link control (LLC) layer at any station (STA) associated with one of those BSSs.

extended service set (ESS) transition: A station (STA) movement from one basic service set (BSS) in one ESS to another BSS in a different ESS.

fast basic service set (BSS) transition: A station (STA) movement that is from one BSS in one extended service set (ESS) to another BSS within the same ESS and that minimizes the amount of time that data connectivity is lost between the STA and the distribution system (DS).

fast basic service set (BSS) transition (FT) 4-Way Handshake: A pairwise key management protocol used during FT initial mobility domain association. This handshake confirms mutual possession of a pairwise master key, the PMK-R1, by two parties and distributes a group temporal key (GTK).

fast basic service set (BSS) transition (FT) initial mobility domain association: The first association or first reassociation procedure within a mobility domain, during which a station (STA) indicates its intention to use the FT procedures.

fixed station (STA): A STA that is physically attached to a specific location. In licensed bands, a fixed STA might be authorized to operate only at a specific location.

flexible multicast service (FMS): A service that enables a non-access-point (non-AP) station (STA) to request a multicast delivery interval longer than the delivery traffic indication map (DTIM) interval for the purposes of lengthening the period of time a STA can be in a power save state.

flexible multicast stream identifier (FMSID): An identifier assigned by the access point (AP) to a particular group addressed stream subsequent to a successful FMS Request.

flexible multicast service (FMS) stream: A succession of frames transmitted by the access point (AP) that correspond to a single flexible multicast stream identifier (FMSID).

flexible multicast service (FMS) stream set: A collection of FMS streams identified by an FMS Token, used during the FMS Request procedure.

forwarding information: The information maintained by a mesh station (STA) that allows the mesh STA to perform its path selection and forwarding functions.

fragmentation: The process of segmenting a medium access control (MAC) service data unit (MSDU) or MAC management protocol data unit (MMPDU) into a sequence of smaller MAC protocol data units

(MPDUs) prior to transmission. The process of recombining a set of fragment MPDUs into an MSDU or MMPDU is known as defragmentation. These processes are described in 5.8.1.9 of ISO/IEC 7498-1:1994.

Gaussian frequency shift keying (GFSK): A modulation scheme in which the data are first filtered by a Gaussian filter in the baseband and then modulated with a simple frequency modulation.

group: The entities in a wireless network, e.g., an access point (AP) and its associated stations (STAs), or all the STAs in an independent basic service set (IBSS) network.

group address: A medium access control (MAC) address that has the group bit equal to 1. *Syn*: multicast address.

group addressed: When applied to a medium access control (MAC) service data unit (MSDU), it is an MSDU with a group address as the destination address (DA). When applied to a MAC protocol data unit (MPDU), it is an MPDU with a group address in the Address 1 field. *Syn*: **multicast**.

Group Key Handshake: A group key management protocol defined by this standard. It is used only to issue a new group temporal key (GTK) to peers with whom the local station (STA) has already formed security associations.

group master key (GMK): An auxiliary key that might be used to derive a group temporal key (GTK).

group temporal key (GTK): A random value, assigned by the group source, which is used to protect group addressed medium access control (MAC) protocol data units (MPDUs) from that source. The GTK might be derived from a group master key (GMK).

hidden station (STA): A STA whose transmissions are not detected using carrier sense (CS) by a second STA, but whose transmissions interfere with transmissions from the second STA to a third STA

homogenous extended service set (ESS): A collection of basic service sets (BSSs), within the same extended service set (ESS), in which every subscription service provider network (SSPN) or other external network reachable at one BSS is reachable at all of them.

hybrid coordination function (HCF): A coordination function that combines and enhances aspects of the contention-based and contention-free access methods to provide quality-of-service (QoS) stations (STAs) with prioritized and parameterized QoS access to the wireless medium (WM), while continuing to support non-QoS STAs for best-effort transfer. The HCF includes the functionality provided by both enhanced distributed channel access (EDCA) and HCF controlled channel access (HCCA). The HCF is compatible with the distributed coordination function (DCF) and the point coordination function (PCF). It supports a uniform set of frame formats and exchange sequences that STAs might use during both the contention period (CP) and the contention-free period (CFP).

hybrid coordinator (HC): A type of coordinator, defined as part of the quality-of-service (QoS) facility, that implements the frame exchange sequences and medium access control (MAC) service data unit (MSDU) handling rules defined by the hybrid coordination function (HCF). The HC operates during both the contention period (CP) and contention-free period (CFP). The HC performs bandwidth management including the allocation of transmission opportunities (TXOPs) to QoS stations (STAs). The HC is collocated with a QoS access point (AP).

hybrid coordination function (HCF) controlled channel access (HCCA): The channel access mechanism utilized by the hybrid coordinator (HC) to coordinate contention-free media use by quality-of-service (QoS) stations (STAs) for downlink individually addressed, uplink, and direct-link transmissions.

idle power indicator (IPI): A physical layer (PHY) indication of the total channel power (noise and interference) as measured in the channel at the receiving antenna connector while the station (STA) is idle, i.e., neither transmitting nor receiving a frame.

IEEE 802.1X authentication: Extensible Authentication Protocol (EAP) authentication transported by the IEEE 802.1X protocol.

independent basic service set (IBSS): A basic service set (BSS) that forms a self-contained network, and in which no access to a distribution system (DS) is available.

individual address: A medium access control (MAC) address in which the group bit is 0. *Syn:* **directed address**, **unicast address**.

individually addressed: When applied to a medium access control (MAC) service data unit (MSDU), it is an MSDU with an individual address as the destination address (DA). When applied to a MAC protocol data unit (MPDU), it is an MPDU with an individual address in the Address 1 field. *Syn:* **directed, unicast.**

infrastructure: The infrastructure includes the distribution system medium (DSM), access point (AP), and portal entities. It is also the logical location of distribution and integration service functions of an extended service set (ESS). An infrastructure contains one or more APs and zero or more portals in addition to the distribution system (DS).

infrastructure authorization information: The information that specifies the access rights of the user of a non-access-point (non-AP) station (STA). This information might include the rules for routing the user traffic, a set of permissions about services that a user is allowed to access, quality-of-service (QoS) configuration information, or the accounting policy to be applied by the infrastructure.

integration service: The service that enables delivery of medium access control (MAC) service data units (MSDUs) between the distribution system (DS) and a local area network (LAN) (via a portal).

integrity GTK (IGTK): A random value, assigned by the broadcast/multicast source STA, which is used to protect group addressed medium access control (MAC) management protocol data units (MMPDUs) from that source STA.

light sleep mode: A mesh power mode in which the mesh station (STA) operates either in the Awake state or in the Doze state towards a neighbor mesh STA, and is expected to receive beacons from this neighbor peer mesh STA.

link: In the context of an IEEE 802.11 medium access control (MAC) entity, a physical path consisting of exactly one traversal of the wireless medium (WM) that is used to transfer an MAC service data unit (MSDU) between two stations (STAs).

link margin: Ratio of the received signal power to the minimum required by the station (STA). The STA might incorporate rate information and channel conditions, including interference, into its computation of link margin. The specific algorithm for computing the link margin is implementation dependent.

link metric: A criterion used to characterize the performance, quality, and eligibility of a link.

little endian: The concept that, for a given multi-octet numeric representation, the least significant octet has the lowest address.

liveness: A demonstration that the peer is actually participating in this instance of communication.

location configuration information (LCI): As defined in IETF RFC 3825: includes latitude, longitude, and altitude, with resolution indicators for each.

location subject local: The term used when a location request is for the location of the requesting STA, i.e., when the requesting STA asks, "Where am I?"

location subject remote: The term used when a location request is for the location of the reporting STA, i.e., when the requesting STA asks, "Where are you?"

location subject third party: The term used when the location request is for the location of a station (STA) other than the requesting STA or the requested STA, (i.e., when the requesting STA asks, "Where is he/she?")

master session key (MSK): Keying material that is derived between the Extensible Authentication Protocol (EAP) peer and exported by the EAP method to the Authentication Server (AS).
NOTE—In this standard, this key is at least 64 octets in length.

medium access control (MAC) management protocol data unit (MMPDU): The unit of data exchanged between two peer MAC entities, using services of the physical layer (PHY), to implement the MAC management protocol.

medium access control (MAC) protocol data unit (MPDU): The unit of data exchanged between two peer MAC entities using the services of the physical layer (PHY). *Syn:* **frame**.

medium access control (MAC) service data unit (MSDU): Information that is delivered as a unit between MAC service access points (SAPs).

mesh basic service set (MBSS): A basic service set (BSS) that forms a self-contained network of mesh stations (STAs) that use the same mesh profile. An MBSS contains zero or more mesh gates, and can be formed from mesh STAs that are not in direct communication.

mesh facility: The set of enhanced functions, channel access rules, frame formats, mutual authentication methods, and managed objects used to provide data transfer among autonomously operating stations (STAs) that might not be in direct communication with each other over a single instance of the wireless medium. Communication between STAs using the mesh facility takes place using only the wireless medium. The mesh facility transports an MSDU between source and destination STAs over potentially multiple hops of the wireless medium without transiting the MAC_SAP at intermediate STAs.

mesh gate: Any entity that has mesh station (STA) functionality and provides access to one or more distribution systems, via the wireless medium (WM) for the mesh basic service set (MBSS).

mesh link: A link from one mesh station (STA) to a neighbor mesh STA that have a mesh peering with each other.

mesh neighborhood: The set of all neighbor mesh stations (STAs) relative to a particular mesh STA.

mesh path: A concatenated set of mesh links from a source mesh station (STA) to a destination mesh STA.

mesh path selection: The process of selecting a mesh path.

mesh peer service period (MPSP): A contiguous period of time during which one or more individually addressed frames are transmitted between two peer mesh stations (STAs) with at least one of those mesh STAs operating in light sleep or deep sleep mode. A mesh peer service period is directional and may contain one or more transmission opportunities (TXOPs). A mesh STA may have multiple mesh peer service periods ongoing in parallel. No more than one mesh peer service period may be set up in each direction with each peer mesh STA.

mesh peer service period (MPSP) owner: A mesh station (STA) that obtains transmission opportunities (TXOPs), transmits individually addressed frames to the recipient mesh STA in the mesh peer service period, and terminates the mesh peer service period.

mesh peering: A relationship between two mesh stations (STAs) that is required for direct communication over a single instance of the wireless medium (WM). A mesh peering is established with a mesh peering protocol.

mesh peering management: A group of protocols to facilitate the mesh peering establishment and closure of the mesh peerings.

mesh power mode: The activity level identifier of a mesh station (STA) set per mesh peering or for nonpeer neighbor STAs. A lower activity level enables a mesh STA to reduce its power consumption.

mesh power mode tracking: Operation to observe the peering-specific mesh power modes from the peer mesh STAs and to maintain the peering-specific mesh power modes for each peer mesh STA.

mesh profile: A set of values of parameters that identifies the attributes of the mesh basic service set (MBSS) and that is used in a single mesh BSS. The mesh profile consists of the identifiers that are the values for the parameters: mesh ID, active path selection protocol, active path selection metric, congestion control mode, synchronization method, and authentication protocol.

mesh services: The set of services that enable the creation and operation of a mesh basic service set (MBSS).

mesh station (STA): A quality-of-service (QoS) STA that implements the mesh facility.

message integrity code (MIC): A value generated by a cryptographic function. If the input data are changed, a new value cannot be correctly computed without knowledge of the cryptographic key(s) used by the cryptographic function.

NOTE—This is traditionally called a *message authentication code* (MAC), but the acronym MAC is already reserved for another meaning in this standard.

mobile station (STA): A type of STA that uses network communications while in motion.

mobility domain: A set of basic service sets (BSSs), within the same extended service set (ESS), that support fast BSS transitions between themselves and that are identified by the set's mobility domain identifier (MDID).

mobility domain identifier (MDID): An identifier that names a mobility domain.

multi-level precedence and preemption (MLPP): A framework used with admission control for the treatment of traffic streams based on precedence, which supports the preemption of an active traffic stream by a higher precedence traffic stream when resources are limited. Preemption is the act of forcibly removing a traffic stream in progress in order to free up resources for another higher precedence traffic stream.

multicast: *See:* **group addressed**.

multicast address: *See:* **group address**.

multicast-group address: A medium access control (MAC) address associated by higher level convention with a group of logically related stations (STAs).

multiple BSSID capability: The capability to advertise information for multiple basic service set identifiers (BSSIDs) using a single Beacon or Probe Response frame instead of using multiple Beacon or Probe Response frames, each corresponding to a single BSSID, and the capability to indicate buffered frames for these multiple BSSIDs using a single traffic indication map (TIM) element in a single Beacon.

multiple input, multiple output (MIMO): A physical layer (PHY) configuration in which both transmitter and receiver use multiple antennas.

neighbor access point (AP): Any AP that is a potential service set transition candidate.

neighbor station (STA): A STA in the following relationship: STA A is a neighbor to STA B if STA A can both directly transmit to and receive from STA B over the wireless medium.

network access identifier (NAI): The user identity submitted by the Supplicant during IEEE 802.1X authentication.
NOTE—See IETF RFC 4282.[15]

network access server (NAS) client: The client component of a NAS that communicates with the Authentication Server (AS).

network allocation vector (NAV): An indicator, maintained by each station (STA), of time periods when transmission onto the wireless medium (WM) is not initiated by the STA regardless of whether the STA's clear channel assessment (CCA) function senses that the WM is busy.

next-hop mesh station (STA): The next peer mesh STA on the mesh path to the destination mesh STA.

nonce: A numerical value, used in cryptographic operations associated with a given cryptographic key, that is not to be reused with that key, including over all reinitializations of the system through all time.

non-access-point (non-AP) station (STA): A STA that is not contained within an AP.

nonoperating channel: A channel that is not the operating channel of the basic service set (BSS) of which the station (STA) is a member.

nonpeer mesh power mode: The activity level identifier of a mesh station (STA) towards nonpeer neighbor mesh STAs. Two nonpeer mesh power modes are defined: active mode and deep sleep mode.

non-quality-of-service (non-QoS) access point (AP): An AP that does not support the quality-of-service (QoS) facility.

non-quality-of-service (non-QoS) basic service set (BSS): A BSS that does not support the quality-of-service (QoS) facility.

non-quality-of-service (non-QoS) station (STA): A STA that does not support the quality-of-service (QoS) facility.

nontransmitted BSSID: A basic service set identifier (BSSID) corresponding to one of the basic service sets (BSSs) when the multiple BSSID capability is supported, where the BSSID is not announced explicitly but can be derived from the information encoded in the transmitted beacon frames.

[15] Information on references can be found in Clause 2.

null data packet (NDP): A physical layer convergence procedure (PLCP) protocol data unit (PPDU) that carries no Data field.

off-channel: Channel that is not the base channel.

operating channel: The operating channel is the channel used by the serving AP of the BSS to transmit beacons. In an IBSS the operating channel is the channel used by the IBSS DFS owner to transmit beacons.

operating channel width: The channel width in which the station (STA) is currently able to receive.

overlapping basic service set (OBSS): A basic service set (BSS) operating on the same channel as the station's (STA's) BSS and within (either partly or wholly) its basic service area (BSA).

over-the-air fast basic service set (BSS) transition (FT): An FT method in which the station (STA) communicates over a WM link to the target access point (AP).

over-the-DS (distribution system) fast basic service set (BSS) transition (FT): An FT method in which the station (STA) communicates with the target access point (AP) via the current AP.

pairwise: Referring to, or an attribute of, two entities that are associated with each other, e.g., an access point (AP) and an associated station (STA), or two STAs in an independent basic service set (IBSS) network. This term is used to refer to a type of encryption key hierarchy pertaining to keys shared by only two entities.

pairwise master key (PMK): The key derived from a key generated by an Extensible Authentication Protocol (EAP) method or obtained directly from a preshared key (PSK).

pairwise master key R0 (PMK-R0): The key at the first level of the fast basic service set (BSS) transition (FT) key hierarchy.

pairwise master key (PMK) R0 name (PMKR0Name): An identifier that names the PMK-R0.

pairwise master key (PMK) R0 key holder identifier (R0KH-ID): An identifier that names the holder of the PMK-R0 in the Authenticator.

pairwise master key R1 (PMK-R1): A key at the second level of the fast basic service set (BSS) transition (FT) key hierarchy.

pairwise master key (PMK) R1 name (PMKR1Name): An identifier that names a PMK-R1.

pairwise master key (PMK) R1 key holder identifier (R1KH-ID): An identifier that names the holder of a PMK-R1 in the Authenticator.

pairwise master key (PMK) S0 key holder identifier (S0KH-ID): An identifier that names the holder of the PMK-R0 in the Supplicant.

pairwise master key (PMK) S1 key holder identifier (S1KH-ID): An identifier that names the holder of the PMK-R1 in the Supplicant.

pairwise transient key (PTK): A concatenation of session keys derived from the pairwise master key (PMK) or from the PMK-R1. Its components include a key confirmation key (KCK), a key encryption key (KEK), and one or more temporal keys that are used to protect information exchanged over the link.

pairwise transient key (PTK) name (PTKName): An identifier that names the PTK.

parameterized quality of service (QoS): The treatment of the medium access control (MAC) protocol data units (MPDUs) depends on the parameters associated with the MPDU. Parameterized QoS is primarily provided through the hybrid coordination function (HCF) controlled channel access (HCCA) mechanism, but is also provided by the enhanced distributed channel access (EDCA) mechanism when used with a traffic specification (TSPEC) for admission control.

pass-phrase: A secret text string employed to corroborate the user's identity.

password: A shared, secret, and potentially low-entropy word, phrase, code, or key used as a credential for authentication purposes.
NOTE—The method of distribution of a password to the units in the system is outside the scope of this standard.

path metric: An aggregate multi-hop criterion used to characterize the performance, quality, and eligibility of a mesh path.

peer mesh station (STA): A mesh STA to which a mesh peering has been established.

peer trigger frame: A Mesh Data or quality-of-service (QoS) Null frame that initiates a mesh peer service period.

peer-specific mesh power mode: The activity level identifier of a mesh station (STA) set per mesh peering. Three peer-specific mesh power modes are defined: active mode, light sleep mode, and deep sleep mode.

peer-to-peer link: A direct link within a quality-of-service (QoS) basic service set (BSS), a tunnelled direct-link setup (TDLS) link, or a STA-to-STA communication in an independent basic service set (IBSS).

PeerKey Handshake: A key management protocol composed of the station-to-station link (STSL) master key (SMK) Handshake and the 4-Way STSL transient key (STK) Handshake. This is used to create new SMK security associations (SMKSAs) and STK security associations (STKSAs) to secure the STSLs.

per-frame encryption key: A unique encryption key constructed for each medium access control (MAC) protocol data unit (MPDU).
NOTE—A per-frame encryption key is employed by some security protocols defined in this standard.

piggyback: The overloading of a data frame with an acknowledgment of a previously received medium access control (MAC) protocol data unit (MPDU) and/or a poll to the station (STA) to which the frame is directed.

point coordinator (PC): The entity within the STA in an AP that performs the point coordination function.

point coordination function (PCF): A class of possible coordination functions in which the coordination function logic is active in only one station (STA) in a basic service set (BSS) at any given time that the network is in operation.

portable station (STA): A type of station (STA) that might be moved from location to location, but that only uses network communications while at a fixed location.

portal: The logical point at which the integration service is provided.

precursor mesh station (STA): A neighbor peer mesh STA on the mesh path to the destination mesh STA, that identifies the mesh STA as the next-hop mesh STA.

preshared key (PSK): A static key that is distributed to the units in the system by some out-of-band means.

primary channel: The common channel of operation for all stations (STAs) that are members of the basic service set (BSS).

prioritized quality of service (QoS): The provisioning of service in which the medium access control (MAC) protocol data units (MPDUs) with higher priority are given a preferential treatment over MPDUs with a lower priority. Prioritized QoS is provided through the enhanced distributed channel access (EDCA) mechanism.

protection mechanism: Any procedure that attempts to update the network allocation vector (NAV) of all receiving stations (STAs) prior to the transmission of a frame that might or might not be detected as valid network activity by the PHY entities at those receiving STAs.

protection mechanism frame: Any frame that is sent as part of a protection mechanism procedure.

protocol instance: An execution of a particular protocol that consists of the state of the communicating parties as well as the messages exchanged.

proxy mesh gate: A mesh gate acting as an intermediary for IEEE 802 stations (STAs) outside the mesh basic service set (MBSS).

pseudorandom function (PRF): A function that hashes various inputs to derive a pseudorandom value. In order to ensure liveness of a communication in which a pseudorandom value is used, a nonce is used as one of the inputs to the function.

public safety answering point (PSAP): A physical location where emergency calls are received and routed to the appropriate emergency service dispatch center.
NOTE—See NENA 08-002 [B52].

quadrature binary phase shift keying (QBPSK): A binary phase shift keying modulation in which the binary data is mapped onto the imaginary (Q) axis.

quality-of-service (QoS) access point (AP): An AP that supports the QoS facility. The functions of a QoS AP are a superset of the functions of a non-QoS AP, and thus a QoS AP is able to function as a non-QoS AP to non-QoS stations (STAs).

quality-of-service (QoS) basic service set (BSS): A BSS that provides the QoS facility. An infrastructure QoS BSS contains a QoS access point (AP).

quality-of-service (QoS) facility: The set of enhanced functions, channel access rules, frame formats, frame exchange sequences and managed objects used to provide parameterized and prioritized QoS.

quality-of-service (QoS) independent basic service set (IBSS): An IBSS in which one or more of its stations (STAs) support the QoS facility.

quality-of-service (QoS) station (STA): A STA that implements the QoS facility. A QoS STA acts as a non-QoS STA when associated in a non-QoS basic service set (BSS).

reassociation service: The service that enables an established association [between access point (AP) and station (STA)] to be transferred from one AP to another (or the same) AP.

receive chain: The physical entity that implements any necessary signal processing to provide the received signal to the digital baseband. Such signal processing includes filtering, amplification, down-conversion, and sampling.

receive power: Mean power measured at the antenna connector.

received channel power indicator (RCPI): An indication of the total channel power (signal, noise, and interference) of a received frame measured on the channel and at the antenna connector used to receive the frame.

received power indicator (RPI): A quantized measure of the received power level as seen at the antenna connector.

received signal to noise indicator (RSNI): An indication of the signal to noise plus interference ratio of a received frame. RSNI is defined by the ratio of the received signal power (RCPI-ANPI) to the noise plus interference power (ANPI) as measured on the channel and at the antenna connector used to receive the frame.

NOTE—RCPI and ANPI might not be measured simultaneously; see 10.11.9.4 for details.

registered station (STA): A STA for which information needs to be submitted to an appropriate regulatory or coordination authority before it is allowed to transmit.

remote request broker (RRB): The component of the station management entity (SME) of an access point (AP) that supports fast basic service set (BSS) transitions over the distribution system (DS).

resource information container (RIC): A sequence of elements that include resource request and response parameters.

restricted channel: A radio channel in which transmission is restricted to stations (STAs) that operate under the control of licensed operators.

roaming consortium: A group of subscription service providers (SSPs) having inter-SSP roaming agreements.

Robust Management frame: A management frame that is eligible for protection.

scheduled service period (SP): The SP that is scheduled by the quality-of-service (QoS) access point (AP). Scheduled SPs start at fixed intervals of time.

service interval (SI): The interval between the start of two successive scheduled service periods (SPs).

service period (SP): A contiguous time during which one or more downlink individually addressed frames are transmitted to a quality-of-service (QoS) station (STA) and/or one or more transmission opportunities (TXOPs) are granted to the same STA. SPs are either scheduled or unscheduled. For a non-access-point (non-AP) STA, there can be at most one SP active at any time.

service set transition: A STA movement from one BSS to another BSS, i.e., either a BSS transition or an ESS transition.

serving AP: The AP to which the STA is associated.

sounding: The use of preamble training fields to measure the channel for purposes other than demodulation of the Data portion of the physical layer convergence procedure (PLCP) protocol data unit (PPDU) containing the training fields.

NOTE—These uses include calculation of transmit steering, calculation of recommended MCS, and calculation of calibration parameters.

source mesh station (STA): A mesh STA from which a MAC service data unit (MSDU) enters the mesh basic service set (MBSS). A source mesh STA may be a mesh STA that is the source of an MSDU or a proxy mesh gate that receives an MSDU from a STA outside of the MBSS and forwards the MSDU on a mesh path.

space-time block coding/spatial multiplexing (STBC/SM): A combination of STBC and SM where one spatial stream is transmitted using STBC and one or two additional spatial streams are transmitted using SM.

space-time streams: Streams of modulation symbols created by applying a combination of spatial and temporal processing to one or more spatial streams of modulation symbols.

spatial multiplexing (SM): A transmission technique in which data streams are transmitted on multiple spatial channels that are provided through the use of multiple antennas at the transmitter and the receiver.

spatial stream: One of several streams of bits or modulation symbols that might be transmitted over multiple spatial dimensions that are created by the use of multiple antennas at both ends of a communications link.

station (STA): A logical entity that is a singly addressable instance of a medium access control (MAC) and physical layer (PHY) interface to the wireless medium (WM).

station service (SS): The set of services that support transport of medium access control (MAC) service data units (MSDUs) between stations (STAs) within a basic service set (BSS).

subscription service provider (SSP): An organization (operator) offering connection to network services, perhaps for a fee.

subscription service provider network (SSPN): The network controlled by a subscription service provider (SSP). The network maintains user subscription information.

Supplicant: An entity at one end of a point-to-point LAN segment that is being authenticated by an Authenticator attached to the other end of that link. (IEEE Std 802.1X-2004)

Supplicant address (SPA): The medium access control (MAC) address of the IEEE 802.1X Supplicant.

temporal encryption key: The portion of a pairwise transient key (PTK) or group temporal key (GTK) used directly or indirectly to encrypt data in medium access control (MAC) protocol data units (MPDUs).

temporal key (TK): The combination of temporal encryption key and temporal message integrity code (MIC) key.

temporal message integrity code (MIC) key: The portion of a transient key used to ensure the integrity of medium access control (MAC) service data units (MSDUs) or MAC protocol data units (MPDUs).

time unit (TU): A measurement of time equal to 1024 µs.

traffic category (TC): A label for medium access control (MAC) service data units (MSDUs) that have a distinct user priority (UP), as viewed by higher layer entities, relative to other MSDUs provided for delivery over the same link. Traffic categories are meaningful only to MAC entities that support quality of service (QoS) within the MAC data service. These MAC entities determine the UP for MSDUs belonging to a particular traffic category using the priority value provided with those MSDUs at the MAC service access point (MAC_SAP).

traffic classification (TCLAS): The specification of certain parameter values to identify the medium access control (MAC) service data units (MSDUs) belonging to a particular traffic stream (TS). The classification process, performed above the MAC service access point (MAC_SAP) at a quality-of-service (QoS) access point (AP), uses the parameter values for a given TS to examine each incoming MSDU and determine whether this MSDU belongs to that TS. TCLAS might also occur at non-access-point (non-AP) QoS station (STA) with multiple streams.
NOTE—However, such classification is beyond the scope of this standard.

traffic filter: A set of traffic specifications defined by the use of traffic classification (TCLAS) elements that are utilized by the traffic filtering service (TFS) to identify specific allowed frames.

traffic filtering service (TFS): A service provided by an access point (AP) to a non-AP station (STA) to reduce the number of frames sent to the non-AP STA by not forwarding individually addressed frames addressed to the non-AP STA that do not match traffic filters specified by the non-AP STA.

traffic identifier (TID): Any of the identifiers usable by higher layer entities to distinguish medium access control (MAC) service data units (MSDUs) to MAC entities that support quality of service (QoS) within the MAC data service. There are 16 possible TID values; eight identify TCs, and the other eight identify parameterized TSs. The TID is assigned to an MSDU in the layers above the MAC.

traffic indication map (TIM) broadcast: A service that enables a non-access-point (non-AP) station (STA) to request periodic transmission of a TIM frame by the AP. TIM frames have shorter duration than Beacon frames and can be transmitted at a higher physical layer (PHY) rate, which allows the STA to save additional power while periodically checking for buffered traffic in standby mode, relative to the power consumed if the station (STA) were to periodically wake up to receive a beacon frame.

traffic specification (TSPEC): The quality-of-service (QoS) characteristics of a data flow to and from a QoS station (STA).

traffic stream (TS): A set of medium access control (MAC) service data units (MSDUs) to be delivered subject to the quality-of-service (QoS) parameter values provided to the MAC in a particular traffic specification (TSPEC). TSs are meaningful only to MAC entities that support QoS within the MAC data service. These MAC entities determine the TSPEC applicable for delivery of MSDUs belonging to a particular TS using the priority parameter provided with those MSDUs at the MAC service access point (MAC_SAP).

traffic stream identifier (TSID): Any of the identifiers usable by higher layer entities to distinguish medium access control (MAC) service data units (MSDUs) to MAC entities for parameterized quality of service (QoS) [i.e., the traffic stream (TS) with a particular traffic specification (TSPEC)] within the MAC data service. The TSID is assigned to an MSDU in the layers above the MAC.

transmission opportunity (TXOP): An interval of time when a particular quality-of-service (QoS) station (STA) has the right to initiate frame exchange sequences onto the wireless medium (WM). A TXOP is defined by a starting time and a maximum duration. The TXOP is either obtained by the STA by successfully contending for the channel or assigned by the hybrid coordinator (HC).

transmission opportunity (TXOP) holder: A quality-of-service (QoS) station (STA) that has either been granted a TXOP by the hybrid coordinator (HC) or successfully contended for a TXOP.

transmission opportunity (TXOP) responder: A station (STA) that transmits a frame in response to a frame received from a TXOP holder during a frame exchange sequence, but that does not acquire a TXOP in the process.

transmitted basic service set identifier (BSSID): The BSSID included in the MAC Header transmitter address field of a Beacon frame when the multiple BSSID capability is supported.

tunneled direct-link setup (TDLS): A protocol that uses a specific Ethertype encapsulation to TDLS frames through an access point (AP) to establish a TDLS direct link.

tunneled direct-link setup (TDLS) direct link: Direct link between two non-AP stations (STAs) that has been established using the TDLS protocol.

tunneled direct-link setup (TDLS) initiator station (STA): A STA that transmits a TDLS Setup Request frame or a TDLS Discovery Request frame.

tunneled direct-link setup (TDLS) peer power save mode (PSM): A power save mode that is based on periodically scheduled service periods, which can be used between two stations (STAs) that have set up a TDLS direct link.

tunneled direct-link setup (TDLS) peer power save mode (PSM) initiator: A station (STA) that transmits a TDLS Peer PSM request frame.

tunneled direct-link setup (TDLS) peer power save mode (PSM) responder: A station (STA) that transmits a TDLS Peer PSM response frame.

tunneled direct-link setup (TDLS) peer station (STA): A STA with a TDLS direct link.

tunneled direct-link setup (TDLS) peer unscheduled automatic power save delivery (U-APSD) [TDLS peer U-APSD (TPU)]: A power save mode based on unscheduled service periods that can be used between two stations (STAs) that have set up a TDLS direct link.

tunneled direct-link setup (TDLS) peer unscheduled automatic power save delivery (U-APSD) [TDLS peer U-APSD (TPU)] buffer station (STA): A TDLS peer STA that buffers traffic for a TPU sleep STA.

tunneled direct-link setup (TDLS) peer unscheduled automatic power save delivery (U-APSD) [TDLS peer U-APSD (TPU)] sleep station (STA): A TDLS STA that entered power save mode on a TDLS direct link and that is using TPU for the delivery of buffered traffic.

tunneled direct-link setup (TDLS) power save mode (PSM): TDLS peer PSM or peer unscheduled automatic power save delivery (U-APSD).

tunneled direct-link setup (TDLS) responder station (STA): A STA that receives or is the intended recipient of a TDLS Setup Request frame or TDLS Discovery Request frame.

unauthorized disclosure: The process of making information available to unauthorized individuals, entities, or processes.

unauthorized resource use: Use of a resource not consistent with the defined security policy.

unicast: *See:* **individually addressed**.

unicast address: *See:* **individual address**.

uniform spreading: A regulatory requirement for a channel selection mechanism that provides uniform usage across a minimum set of channels in the regulatory domain.

unreachable destination: A destination mesh station (STA) for which the link to the next hop of the mesh path to this destination mesh STA is no longer usable.

unreachable star: A star that cannot be reached. Often associated with impossible dreams.

unscheduled service period (SP): The period that is started when a quality-of-service (QoS) station (STA) transmits a trigger frame to the QoS access point (AP).

uplink: A unidirectional link from a non-access-point (non-AP) station (STA) to an access point (AP).

user priority (UP): A value associated with an medium access control (MAC) service data unit (MSDU) that indicates how the MSDU is to be handled. The UP is assigned to an MSDU in the layers above the MAC.

validated AP: An AP that has either been explicitly configured as a Neighbor or learned through a mechanism like the Beacon Report.

wildcard BSSID: A BSSID value (all binary 1s) used to represent all BSSIDs.

wildcard SSID: A SSID value (null) used to represent all SSIDs.

wireless distribution system (WDS): Often used as a vernacular term for a mechanism for wireless communication among nonmesh stations (STAs) using a four address frame format.

NOTE—This standard specifies such a frame format and its use only for a mesh basic service set (MBSS). Because of this, the term WDS is obsolete and subject to removal in a subsequent revision of this standard.

wireless local area network (WLAN) system: A system that includes the distribution system (DS), access points (APs), and portal entities. It is also the logical location of distribution and integration service functions of an extended service set (ESS). A WLAN system contains one or more APs and zero or more portals in addition to the DS.

wireless medium (WM): The medium used to implement the transfer of protocol data units (PDUs) between peer physical layer (PHY) entities of a wireless local area network (LAN).

3.2 Definitions specific to IEEE 802.11

The following terms and definitions are specific to terms or references in this standard and are not appropriate for inclusion in the *IEEE Standards Dictionary: Glossary of Terms & Definitions*.

20 MHz basic service set (BSS): A BSS in which the Secondary Channel Offset field is equal to SCN.

20 MHz high-throughput (HT): A Clause 20 transmission using FORMAT=HT_MF or HT_GF and CH_BANDWIDTH=HT_CBW20.

20 MHz mask physical layer convergence procedure (PLCP) protocol data unit (PPDU): A Clause 18 PPDU, a Clause 19 orthogonal frequency division multiplexing (OFDM) PPDU, or a Clause 20 20 MHz high-throughput (HT) PPDU with the TXVECTOR parameter CH_BANDWIDTH equal to HT_CBW20 and the CH_OFFSET parameter equal to CH_OFF_20. The PPDU is transmitted using a 20 MHz transmit spectral mask defined in Clause 18, Clause 19, or Clause 20, respectively.

20 MHz physical layer convergence procedure (PLCP) protocol data unit (PPDU): A Clause 16 PPDU, Clause 18 PPDU, Clause 17 PPDU, Clause 19 orthogonal frequency division multiplexing (OFDM) PPDU, or Clause 20 20 MHz high-throughput (HT) PPDU with the TXVECTOR parameter CH_BANDWIDTH equal to HT_CBW20.

20/40 MHz basic service set (BSS): A BSS in which the supported channel width of the access point (AP) or dynamic frequency selection (DFS) owner (DO) station (STA) is 20 MHz and 40 MHz (Channel Width field is equal to 1) and the Secondary Channel Offset field is equal to a value of SCA or SCB.

40-MHz-capable (FC) high-throughput (HT) access point (AP): An HT AP that included a value of 1 in the Supported Channel Width Set subfield (indicating its capability to operate on a 40 MHz channel) of its most recent transmission of a frame containing an HT Capabilities element.

40-MHz-capable (FC) high-throughput (HT) access point (AP) 2G4: An HT AP 2G4 that is also an FC HT AP.

40-MHz-capable (FC) high-throughput (HT) access point (AP) 5G: An HT AP 5G that is also an FC HT AP.

40-MHz-capable (FC) high-throughput (HT) station (STA): An HT STA that included a value of 1 in the Supported Channel Width Set subfield (indicating its capability to operate on a 40 MHz channel) of its most recent transmission of a frame containing an HT Capabilities element.

40-MHz-capable (FC) high-throughput (HT) station (STA) 2G4: An HT STA 2G4 that is also an FC HT STA.

40-MHz-capable (FC) high-throughput (HT) station (STA) 5G: An HT STA 5G that is also an FC HT STA.

40 MHz high throughput (HT): A Clause 20 transmission using FORMAT=HT_MF or HT_GF and CH_BANDWIDTH=HT_CBW40.

40 MHz mask physical layer convergence procedure (PLCP) protocol data unit (PPDU): One of the following PPDUs: 1) a 40 MHz high-throughput (HT) PPDU (TXVECTOR parameter CH_BANDWIDTH equal to HT_CBW40); 2) a 40 MHz non-HT duplicate PPDU (TXVECTOR parameter CH_BANDWIDTH equal to NON_HT_CBW40); or 3) a Clause 20 20 MHz HT PPDU with the TXVECTOR parameter CH_BANDWIDTH equal to HT_CBW20 and the CH_OFFSET parameter equal to either CH_OFF_20U or CH_OFF_20L. The PPDU is transmitted using a 40 MHz transmit spectral mask defined in Clause 20.

40 MHz physical layer convergence procedure (PLCP) protocol data unit (PPDU): A 40 MHz high-throughput (HT) PPDU (TXVECTOR parameter CH_BANDWIDTH equal to HT_CBW40) or a 40 MHz non-HT duplicate PPDU (TXVECTOR parameter CH_BANDWIDTH equal to NON_HT_CBW40) as defined in Clause 20.

4-Way Handshake: A pairwise key management protocol defined by this standard. This handshake confirms mutual possession of a pairwise master key (PMK) by two parties and distributes a group temporal key (GTK).

4-Way station-to-station link (STSL) transient key (STK) Handshake: A key management protocol between two parties that confirms mutual possession of an STSL master key (SMK) and distributes an STK.

Access Network Query Protocol (ANQP): The query protocol for access network information retrieval transported by generic advertisement service (GAS) Public Action frames.

advertisement protocol: Access Network Query Protocol (ANQP) and higher layer protocols defined external to this standard that are used for network and service discovery.

advertisement server: A logical server that provides the information repository for a specific advertisement protocol. The location of the physical server that instantiates the advertisement server is outside the scope of this specification.

basic space-time block coding (STBC) modulation and coding scheme (MCS): An MCS value and STBC encoder specification used in the transmission of STBC-encoded control frames and STBC-encoded group addressed frames. The value is defined in 9.7.3.

BSSBasicMCSSet: The set of modulation and coding scheme (MCS) values that are supported by all high-throughput (HT) stations (STAs) that are members of an HT basic service set (BSS).

bufferable management frame: Either an individually addressed Probe Response frame that is sent in an IBSS in response to an individually addressed Probe Request frame, or an Action, Disassociation, or Deauthentication frame.

bufferable medium access control (MAC) management protocol data unit (MMPDU): An MMPDU that is transmitted using one or more bufferable management frames.

bufferable unit (BU): An MSDU, A-MSDU (HT STAs only) or bufferable MMPDU that is buffered to operate the power saving protocol.

delivery traffic indication message (DTIM) interval: The interval between the consecutive TBTTs of beacons containing a DTIM. The value, expressed in time units, is equal to the product of the value in the Beacon Interval field and the value in the DTIM Period subfield in the TIM element in Beacon frames.

direct sequence spread spectrum orthogonal frequency division multiplexing (DSSS-OFDM): A PHY using DSSS-OFDM modulation under 19.7 rules.

direct sequence spread spectrum/complementary code keying (DSSS/CCK): A Clause 16 or Clause 17 transmission.

dynamic frequency selection (DFS) owner (DO) station (STA): A STA that is the DFS Owner of an IBSS or off-channel tunneled direct-link setup (TDLS) direct link that is operating on a channel within an operating class that has a value of 20 or 40 for the entry in the “Channel spacing” column and that has a value of 5 for the entry in the “Channel starting frequency” column of any of the tables found in E.1.

EAPOL-Key confirmation key (KCK): A key used to integrity-check an EAPOL-Key frame.

EAPOL-Key encryption key (KEK): A key used to encrypt the Key Data field in an EAPOL-Key frame.

emergency services association: A robust security network association (RSNA) between an access point (AP) and a non-AP station (STA) without security credentials; the non-AP STA is granted access to emergency services using unprotected frames via this association.

extended channel switching (ECS): A procedure that is used to announce a pending change of operating channel, operating class, or both.

extended rate PHY (ERP): A PHY conforming to Clause 19

extended rate PHY using CCK modulation (ERP-CCK): A PHY operating under Clause 19 rules.

extended rate PHY using DSSS modulation (ERP-DSSS): A PHY operating under Clause 19 rules.

extended rate PHY using DSSS or CCK modulation (ERP-DSSS/CCK): A PHY operating under Clause 19 rules.

extended rate PHY using OFDM modulation (ERP-OFDM): A PHY operating under 19.5 rules.

extended rate PHY using extended rate PBCC modulation (ERP-PBCC): A PHY operating under 19.6 rules.

extended service set (ESS) link: In the context of an IEEE 802.11 medium access control (MAC) entity, a connection path through the wireless medium between a non-access-point (non-AP) station (STA) and one of the APs that is a member of the ESS.

fast BSS transition (FT) originator: A STA that initiates the FT protocol by sending an FT Request frame or an Authentication frame with Authentication Algorithm equal to Fast BSS Transition.

IEEE 802.11 station (STA): Any station that is conformant to IEEE 802.11. Any reference to the term station (STA) in this standard where not qualified by the term IEEE 802.11 implicitly refers to an IEEE 802.11 station.

generic advertisement service (GAS): An over-the-air transportation service that provides over-the-air transportation for frames of higher layer advertisements between stations (STAs) or between an advertisement server and a non-access-point (non-AP) STA. The protocol(s) used to relay frames between an AP, portal, and advertisement server is outside the scope of this standard. GAS supports higher layer protocols that employ a query/response mechanism.

group addressed bufferable unit (BU): A group addressed MSDU or group addressed bufferable MMPDU.

group temporal key security association (GTKSA): The context resulting from a successful group temporal key (GTK) distribution exchange via either a Group Key Handshake or a 4-Way Handshake.

high-throughput (HT) basic service set (BSS): A BSS in which Beacon frames transmitted by an HT station (STA) include the HT Capabilities element.

high-throughput-delayed (HT-delayed) block acknowledgment (Ack): A Delayed Block Ack mechanism that requires the use of the compressed BlockAck frame and the No Acknowledgement ack policy setting within both BlockAckReq and BlockAck frames. This Block Ack scheme is negotiated between two HT stations (STAs) that both support HT Delayed Block Ack.

high-throughput-immediate (HT-immediate) block acknowledgment (Ack): An Immediate Block Ack mechanism that requires the use of the compressed BlockAck frame and an implicit Block Ack request and allows partial-state operation at the recipient. This Block Ack scheme is negotiated between two HT stations (STAs).

high-throughput-greenfield (HT-greenfield) format: A physical layer convergence procedure (PLCP) protocol data unit (PPDU) format of the HT physical layer (PHY) using the HT-greenfield format preamble. This format is represented at the PHY data service access point (SAP) by the TXVECTOR/RXVECTOR FORMAT parameter being equal to HT_GF.

high-throughput-mixed (HT-mixed) format: A physical layer convergence procedure (PLCP) protocol data unit (PPDU) format of the HT physical layer (PHY) using the HT-mixed format preamble. This format is represented at the PHY data service access point (SAP) by the TXVECTOR/RXVECTOR FORMAT parameter being equal to HT_MF.

high-throughput (HT) physical layer convergence procedure (PLCP) protocol data unit (PPDU): A Clause 20 PPDU with the TXVECTOR FORMAT parameter equal to HT_MF or HT_GF.

high-throughput (HT) station (STA) 2G4: An HT STA that is also a STA 2G4.

high-throughput (HT) station (STA) 5G: An HT STA that is also a STA 5G.

individually addressed bufferable unit (BU): An individually addressed MSDU, individually addressed A-MSDU (HT STAs only) or individually addressed bufferable MMPDU.

interworking service: A service that supports use of an IEEE 802.11 network with non-IEEE 802.11 networks. Functions of the interworking service assist non-access-point (non-AP) stations (STAs) in discovering and selecting IEEE 802.11 networks, in using appropriate quality-of-service (QoS) settings for transmissions, in accessing emergency services, and in connecting to subscription service providers (SSPs).

key counter: A 256-bit (32-octet) counter that is used in the pseudorandom function (PRF) to generate initialization vectors (IVs). There is a single key counter per station (STA) that is global to that STA.

key data encapsulation (KDE): Format for data other than elements in the EAPOL-Key Data field.

key management service: A service to distribute and manage cryptographic keys within a robust security network (RSN).

mesh awake window: A period of time during which the mesh station (STA) operates in awake state after its Beacon or Probe Response frame transmission that contained the Mesh Awake Window element.

mesh coordination function (MCF): A coordination function that combines aspects of the contention-based and scheduled access methods. The MCF includes the functionality provided by both enhanced distributed channel access (EDCA) and MCF controlled channel access (MCCA).

mesh coordination function (MCF) controlled channel access (MCCA): A coordination function for the mesh basic service set (MBSS).

mesh coordination function (MCF) controlled channel access opportunity (MCCAOP): A period of time scheduled for frame transmissions between mesh stations (STAs) using MCF controlled channel access (MCCA).

Mesh Data frame: An individually addressed Data frame with both the From DS and To DS bits set to 1 and that is transmitted from a mesh station (STA) to a peer mesh STA, or a group addressed Data frame that has From DS set to 1 and To DS set to 0 that is transmitted by a mesh STA.

Michael: The message integrity code (MIC) for the Temporal Key Integrity Protocol (TKIP).

minimum downlink transmission time (DTT) to uplink transmission time (UTT) [DTT2UTT] spacing: The minimum time within a power save multi-poll (PSMP) sequence between the end of a station's (STA's) PSMP-DTT and the start of its PSMP-UTT.

modulation and coding scheme (MCS): A specification of the high-throughput (HT) physical layer (PHY) parameters that consists of modulation order (e.g., BPSK, QPSK, 16-QAM, 64-QAM) and forward error correction (FEC) coding rate (e.g., 1/2, 2/3, 3/4, 5/6).

modulation and coding scheme 32 (MCS 32) format: A physical layer convergence procedure (PLCP) protocol data unit (PPDU) format of the high-throughput (HT) physical layer (PHY) in which signals in two halves of the occupied channel width contain the same information. This HT PPDU format supports the lowest rate.

modulation and coding scheme (MCS) feedback (MFB) requester: A station (STA) that transmits a physical layer convergence procedure (PLCP) protocol data unit (PPDU) containing an HT Control field in which the modulation and coding scheme (MCS) request (MRQ) subfield is equal to 1.

modulation and coding scheme (MCS) feedback (MFB) responder: A station (STA) that transmits a physical layer convergence procedure (PLCP) protocol data unit (PPDU) containing an HT Control field with the MFB field containing an MCS index or the value 127 in response to a PPDU containing an HT Control field in which the modulation and coding scheme (MCS) request (MRQ) subfield is equal to 1.

multiple basic service set identifier (BSSID) set: A collection of cooperating APs, such that all the APs use a common operating class, channel, and antenna connector.

non-40-MHz-capable (non-FC) high-throughput (HT) station (STA): A STA that is not an FC HT STA.

nonaggregate medium access control (MAC) protocol data unit (non-A-MPDU) frame: A frame that is transmitted in a physical layer convergence procedure (PLCP) protocol data unit (PPDU) with the TXVECTOR AGGREGATION parameter either absent or equal to NOT_AGGREGATED.

nonbufferable medium access control (MAC) management protocol data unit (MMPDU): An MMPDU that is not a bufferable MMPDU.

nonextended rate PHY (NonERP): A PHY conforming to Clause 16 or Clause 17, but not to Clause 19

non-high-throughput (non-HT) duplicate: A transmission format of the physical layer (PHY) that duplicates a 20 MHz non-HT transmission in two adjacent 20 MHz channels and allows a station (STA) in a non-HT basic service set (BSS) on either channel to receive the transmission.

non-high-throughput (non-HT) duplicate frame: A frame transmitted in a non-HT duplicate physical layer convergence procedure (PLCP) protocol data unit (PPDU).

non-high-throughput (non-HT) duplicate physical layer convergence procedure (PLCP) protocol data unit (PPDU): A PPDU transmitted by a Clause 20 physical layer (PHY) with the TXVECTOR FORMAT parameter equal to NON_HT and the CH_BANDWIDTH parameter equal to NON_HT_CBW40.

non-high-throughput (non-HT) physical layer convergence procedure (PLCP) protocol data unit (PPDU): A Clause 20 physical layer (PHY) PPDU with the TXVECTOR FORMAT parameter equal to NON_HT.

Non-High-Throughput (non-HT) SIGNAL field (L-SIG) transmit opportunity (TXOP) protection: A protection mechanism in which protection is established by the non-HT SIG Length and Rate fields indicating a duration that is longer than the duration of the PPDU itself.

non-phased-coexistence-operation-capable (non-PCO-capable) 20/40 station (STA): A high-throughput (HT) STA that included a value of 1 in the Supported Channel Width Set subfield (indicating its capability to operate on a 40 MHz channel) of its most recent transmission of a frame containing an HT Capabilities element and that sets the PCO field in the HT Extended Capabilities field to 0.

non-space-time-block-coding (non-STBC) frame: A frame that is transmitted in a physical layer convergence procedure (PLCP) protocol data unit (PPDU) that has the TXVECTOR STBC parameter equal to 0, or a frame that is received in a PPDU that has the RXVECTOR STBC parameter equal to 0.

null data packet (NDP) announcement: A physical layer convergence procedure (PLCP) protocol data unit (PPDU) that contains one or more +HTC frames (i.e., frames with an HT (high-throughput) Control field) that have the NDP Announcement subfield equal to 1.

operating class: An E.1 index into a set of values for radio operation in a regulatory domain.

pairwise master key (PMK) R0 key holder (R0KH): The component of robust security network association (RSNA) key management of the Authenticator that is authorized to derive and hold the PMK-R0, derive the PMK-R1s, and distribute the PMK-R1s to the R1KHs.

pairwise master key (PMK) R1 key holder (R1KH): The component of robust security network association (RSNA) key management of the Authenticator that receives a PMK-R1 from the R0KH, holds the PMK-R1, and derives the PTKs.

pairwise master key (PMK) S0 key holder (S0KH): The component of robust security network association (RSNA) key management of the Supplicant that derives and holds the PMK-R0, derives the PMK-R1s, and provides the PMK-R1s to the S1KH.

pairwise master key (PMK) S1 key holder (S1KH): The component of robust security network association (RSNA) key management in the Supplicant that receives a PMK-R1 from the S0KH, holds the PMK-R1, and derives the PTKs.

pairwise master key security association (PMKSA): The context resulting from a successful IEEE 802.1X authentication exchange between the peer and Authentication Server (AS) or from a preshared key (PSK).

pairwise transient key security association (PTKSA): The context resulting from a successful 4-Way Handshake exchange between the peer and Authenticator.

payload protected (PP) aggregate medium access control (MAC) service data unit (A-MSDU): An A-MSDU that is protected with CTR with CBC-MAC Protocol (CCMP) but does not include the A-MSDU Present field (bit 7 of the QoS Control field) in the construction of the additional authentication data (AAD).

per-frame sequence counter: For Temporal Key Integrity Protocol (TKIP), the counter that is used as the nonce in the derivation of the per-frame encryption key. For Counter mode with Cipher-block chaining Message authentication code Protocol (CCMP), the per-frame initialization vector (IV).

phased coexistence operation (PCO): A basic service set (BSS) mode with alternating 20 MHz and 40 MHz phases controlled by an access point (AP).

phased coexistence operation (PCO) active access point (AP): A high-throughput (HT) AP that is operating PCO.

phased coexistence operation (PCO) active basic service set (BSS): A BSS in which a PCO active access point (AP) has the PCO Active field in the HT Operation element equal to 1.

phased coexistence operation (PCO) active non-access-point (non-AP) station (STA): A high-throughput (HT) non-AP STA that is associated with a PCO basic service set (BSS) and following PCO.

phased coexistence operation (PCO) active station (STA): A STA that is either a PCO active access point (AP) or a PCO active non-AP STA.

phased-coexistence-operation-capable (PCO-capable) access point (AP): A high-throughput (HT) AP that sets the PCO field in the HT Extended Capabilities field to 1.

phased-coexistence-operation-capable (PCO-capable) non-access-point (non-AP) station (STA): A high-throughput (HT) non-AP STA that sets the PCO field in the HT Extended Capabilities field to 1.

phased-coexistence-operation-capable (PCO-capable) station (STA): A STA that is either a PCO capable access point (AP) or a PCO capable non-AP STA.

phased coexistence operation (PCO) inactive basic service set (BSS): A 20/40 MHz BSS in which an access point (AP) has the PCO Active field in the HT Operation element equal to 0.

power save multi-poll (PSMP): A mechanism that provides a time schedule that is used by an access point (AP) and its stations (STAs) to access the wireless medium. The mechanism is controlled using the PSMP frame.

power save multi-poll (PSMP) burst: A series of one or more PSMP sequences, separated by short interframe space (SIFS).

power save multi-poll downlink transmission time (PSMP-DTT): A period of time described by a PSMP frame during which the access point (AP) transmits.

power save multi-poll (PSMP) sequence: A sequence of frames where the first frame is a PSMP frame that is followed by transmissions in zero or more power save multi-poll downlink transmission times (PSMP-DTTs) and then by transmissions in zero or more power save multi-poll uplink transmission times (PSMP-UTTs). The schedule of the PSMP-DTTs and PSMP-UTTs is defined in the PSMP frame.

power save multi-poll (PSMP) session: The periodic generation of a PSMP burst aligned to its service period (SP).

power save multi-poll uplink transmission time (PSMP-UTT): A period of time described by a PSMP frame during which a non-access-point (non-AP) station (STA) can transmit.

power save multi-poll uplink transmission time (PSMP-UTT) spacing (PUS): The period of time between the end of one PSMP-UTT and the start of the following PSMP-UTT within the same PSMP sequence.

pre-robust security network association (pre-RSNA): The type of association used by a pair of stations (STAs) if the procedure for establishing authentication or association between them did not include the 4-Way Handshake.

pre-robust security network association (pre-RSNA) equipment: A device that is not able to create robust security network associations (RSNAs).

Protected Dual of Public Action frame: An Action frame with the category value specified in 8.4.1.11 Table 8-38. For each Protected Dual of Public Action frame, there is a dual Action frame in a category that is specified with “No” in the “Robust” column of Table 8-38.

quality-of-service (QoS) frame: A frame containing the QoS control field.

reverse direction (RD) initiator: A station (STA) that is a transmit opportunity (TXOP) holder that transmits a medium access control (MAC) protocol data unit (MPDU) in which the RDG/More PPDU subfield is equal to 1.

reverse direction (RD) responder: A station (STA) that is not the RD initiator and whose medium access control (MAC) address matches the value of the Address 1 field of a received MAC protocol data unit (MPDU) in which the RDG/More PPDU subfield is equal to 1.

Robust Action frame: An Action frame with a category value specified in 8.4.1.11 Table 8-38 with “Yes” in the “Robust” column.

robust security network (RSN): A security network that allows only the creation of robust security network associations (RSNAs). An RSN can be identified by the indication in the RSN element (RSNE) of Beacon frames that the group cipher suite specified is not wired equivalent privacy (WEP).

robust security network association (RSNA): The type of association used by a pair of stations (STAs) if the procedure to establish authentication or association between them includes the 4-Way Handshake. Note that existence of an RSNA between two STAs does not of itself provide robust security. Robust security is provided when all STAs in the network use RSNAs.

robust-security-network-association- (RSNA-) capable equipment: A device that contains a station (STA) that is able to create RSNAs. Such a device might use pre-RSNAs because of configuration. Notice that RSNA-capable does not imply full compliance with the RSNA Protocol Implementation Conformance Statement (PICS). A legacy device that has been upgraded to support Temporal Key Integrity Protocol (TKIP) might be RSNA-capable, but is not compliant with the PICS if it does not also support Counter mode with Cipher-block chaining Message authentication code Protocol (CCMP).

robust-security-network-association- (RSNA-) enabled equipment: A station (STA) when it is RSNA-capable and dot11RSNAActivated is true.

robust security network association (RSNA) key management: Key management that includes the 4-Way Handshake, the Group Key Handshake, and the PeerKey Handshake. If fast basic service set (BSS) transition (FT) is enabled, the FT 4-Way Handshake and FT authentication sequence are also included.

secondary channel: A 20 MHz channel associated with a primary channel used by high-throughput (HT) stations (STAs) for the purpose of creating a 40 MHz channel.

security network: A basic service set (BSS) where the station (STA) starting the BSS provides information about the security capabilities and configuration of the BSS by including the robust security network element (RSNE) in Beacon frames.

Self-protected Action frame: An Action frame that is not eligible for protection by the Robust Management frame service. The protection on each Self-protected Action frame is provided by the protocol that uses the frame.

signaling and payload protected (SPP) aggregate medium access control (MAC) service data unit (A-MSDU): An A-MSDU that is protected with CTR with CBC-MAC Protocol (CCMP) and that includes the A-MSDU Present field (bit 7 of the QoS Control field) in the construction of the additional authentication data (AAD).

sounding physical layer convergence procedure (PLCP) protocol data unit (PPDU): A PPDU that is intended by the transmitting station (STA) to enable the receiving STA to estimate the channel between the transmitting STA and the receiving STA. The Not Sounding field in the High-Throughput SIGNAL field (HT-SIG) is equal to 0 in sounding PPDUs.

space-time block coding (STBC) beacon: A beacon that is transmitted using the basic STBC modulation and coding scheme (MCS) to enable discovery of the basic service set (BSS) by high-throughput (HT) stations (STAs) that support the HT STBC feature in order to extend the range of the BSS.

space-time block coding (STBC) delivery traffic indication map (DTIM): An STBC beacon transmission that is a DTIM beacon.

space-time block coding (STBC) frame: A frame that is transmitted in a physical layer convergence procedure (PLCP) protocol data unit (PPDU) that has a nonzero value of the TXVECTOR STBC parameter, or a frame that is received in a PPDU that has a nonzero value of the RXVECTOR STBC parameter.

staggered preamble: A physical layer convergence procedure (PLCP) preamble in a sounding PLCP protocol data unit (PPDU) that is not a null data packet (NDP) and that includes one or more Data Long Training fields (DLTFs) and one or more Extension Long Training fields (ELTFs).

staggered sounding: The use of a sounding physical layer convergence procedure (PLCP) protocol data unit (PPDU) that is not a null data packet (NDP) and that includes one or more Data Long Training fields (DLTFs) and one or more Extension Long Training fields (ELTFs).

station (STA) 2G4: A STA that is operating on a channel that belongs to any operating class that has a value of 25 or 40 for the entry in the “Channel spacing” column and that has a value of 2.407 or 2.414 for the entry in the “Channel starting frequency” column of any of the tables found in E.1.

station (STA) 5G: A STA that is operating on a channel that belongs to any operating class that has a value of 20 or 40 for the entry in the “Channel spacing” column and that has a value of 5 for the entry in the “Channel starting frequency” column of any of the tables found in E.1.

station-to-station link (STSL): A direct link established between two stations (STAs) while associated to a common access point (AP). This term refers to a generic mechanism that allows direct station-to-station communication while remaining in the infrastructure mode. Establishment of this type of link includes an initialization step. The STSL is terminated by specific teardown procedures under the conditions prescribed in this standard. The only example of this procedure currently specified is direct link established by the direct-link setup (DLS).

station-to-station link (STSL) master key (SMK): A random value generated by an access point (AP) during an SMK Handshake. It is used for deriving an STSL transient key (STK).

station-to-station link (STSL) master key (SMK) Handshake: A key management protocol between two parties that creates a new SMK.

station-to-station link (STSL) master key security association (SMKSA): The context resulting from a successful STSL master key (SMK) Handshake.

station-to-station link (STSL) transient key (STK): A value that is derived from the STSL master key (SMK), initiator MAC address (MAC_I), peer MAC address (MAC_P), initiator nonce (INonce), and peer nonce (PNonce), using the pseudorandom function (PRF). The value is split into as many as five keys, i.e., temporal encryption key, two temporal message integrity code (MIC) keys, EAPOL-Key encryption key (KEK), and EAPOL-Key confirmation key (KCK).

station-to-station link (STSL) transient key security association (STKSA): The context resulting from a successful 4-Way STSL transient key (STK) exchange.

subscription service provider (SSP) roaming: The act when a station (STA) uses an SSP’s IEEE 802.11 infrastructure, with which the terminal has no direct agreement, based on a subscription and formal agreement with the STA’s own SSP.

time priority management frame: a management frame that is transmitted using specific frame type channel access rules.

transition security network (TSN): A security network that allows the creation of pre-robust security network associations (pre-RSNAs) as well as RSNAs. A TSN is identified by the indication in the robust security network element (RSNE) of Beacon frames that the group cipher suite in use is wired equivalent privacy (WEP).

transmit power: The effective isotropic radiated power (EIRP) when referring to the operation of an orthogonal frequency division multiplexing (OFDM) physical layer (PHY) in a country where so regulated.

trigger-enabled access category (AC): A quality-of-service (QoS) station (STA) AC where frames of subtype QoS Data and QoS Null from the STA that map to the AC trigger an unscheduled service period (SP) if one is not in progress.

wired equivalent privacy (WEP): A deprecated cryptographic data confidentiality algorithm specified by this standard.

wireless-network-management-sleep (WNM-sleep) mode: An extended power save mode for non-access-point (non-AP) stations (STAs) whereby a non-AP STA need not listen for every delivery traffic indication map (DTIM) Beacon frame and does not perform group temporal key/integrity group temporal key (GTK/IGTK) updates.

3.3 Abbreviations and acronyms

3GPP	3rd Generation Partnership Project
802.x LAN	IEEE 802-based local area networks such as IEEE 802.3 and IEEE 802.11
AA	Authenticator address
AAA	authentication, authorization, and accounting
AAD	additional authentication data
AC	access category
ACI	access category index
ACK	acknowledgment
ACM	admission control mandatory
ACU	admission control unit
ADDBA	add Block Acknowledgment
ADDTS	add traffic stream
AES	advanced encryption standard
AES-128-CMAC	advanced encryption standard (with 128-bit key) cipher-based message authentication code
AFC	Automatic Frequency Control
AGC	Automatic Gain Control
AID	association identifier
AIFS	arbitration interframe space
AIFSN	arbitration interframe space number
AKM	authentication and key management
AKMP	Authentication and Key Management Protocol
AMPE	authenticated mesh peering exchange
A-MPDU	aggregate MAC protocol data unit
A-MSDU	aggregate MAC service data unit
ANonce	Authenticator nonce
ANPI	average noise power indicator
ANQP	Access Network Query Protocol
AP	access point

APSD	automatic power save delivery
ARP	Address Resolution Protocol
AS	Authentication Server
ASEL	antenna selection
ASN.1	Abstract Syntax Notation One
ASRA	additional step required for access
ATIM	announcement traffic indication message
BA	Block Acknowledgment
BAR	Block Acknowledgment request
BCC	binary convolutional code
BIP	Broadcast/Multicast Integrity Protocol
BPSK	binary phase shift keying
BSA	basic service area
BSS	basic service set
BSSID	basic service set identifier
BT	bit time
BU	bufferable unit
CAP	controlled access phase
CBC	cipher-block chaining
CBP	contention-based protocol
CBC-MAC	cipher-block chaining message authentication code
CCA	clear channel assessment
CCK	complementary code keying
CCM	CTR with CBC-MAC
CCMP	CTR with CBC-MAC Protocol
CF	contention-free
CFP	contention-free period
C-MPDU	coded MPDU
CP	contention period
C-PSDU	coded PSDU
CRC	cyclic redundancy code
CS	carrier sense
CSD	cyclic shift diversity
CSI	channel state information
CSMA/CA	carrier sense multiple access with collision avoidance
CTR	counter mode
CTS	clear to send
CTS1	clear to send 1
CTS2	clear to send 2
CW	contention window
DA	destination address
DBPSK	differential binary phase shift keying
DCF	distributed coordination function
DCLA	dc level adjustment
DELBA	delete Block Acknowledgment
DELTS	delete traffic stream

DFS	dynamic frequency selection
DFT	discrete Fourier transform
DIFS	distributed (coordination function) interframe space
DLL	data link layer
DLS	direct-link setup
DLTF	Data Long Training field
DMS	directed multicast service
DMSID	directed multicast service identifier
DN	destination network
DO	DFS owner
Dp	desensitization
DQPSK	differential quadrature phase shift keying
DR	data rate
DS	distribution system
DSCP	differentiated services code point
DSE	dynamic station enablement
DSM	distribution system medium
DSS	distribution system service
DSSDU	distribution system service data unit
DSSS	direct sequence spread spectrum
DSSS-OFDM	Direct sequence spread spectrum orthogonal frequency division multiplexing
DST	daylight saving time
DTIM	delivery traffic indication map
EAP	Extensible Authentication Protocol (IETF RFC 3748-2004 [B38])
EAPOL	Extensible Authentication Protocol over LANs (IEEE Std 802.1X-2004)
EAS	emergency alert system
EBR	expedited bandwidth request
ECS	extended channel switching
ED	energy detection
EDCA	enhanced distributed channel access
EDCAF	enhanced distributed channel access function
EDT	eastern daylight time
EHCC	extended hyperbolic congruence code
EIFS	extended interframe space
EIRP	equivalent isotropically radiated power
ELTF	Extension Long Training field
EOSP	end of service period
ERP	extended rate PHY
ERP-CCK	extended rate PHY using CCK modulation
ERP-DSSS	extended rate PHY using DSSS modulation
ERP-DSSS/CCK	extended rate PHY using DSSS or CCK modulation
ERP-OFDM	extended rate PHY using OFDM modulation
ERP-PBCC	extended rate PHY using extended rate PBCC modulation
ESA	extended service area
ESR	emergency services reachable
ESS	extended service set

EST	eastern standard time
EVM	error vector magnitude
FC	frame control
FCS	frame check sequence
FEC	forward error correction
FER	frame error ratio
FFT	Fast Fourier Transform
FH	frequency hopping
FHSS	frequency-hopping spread spectrum
FIFO	first in first out
FMS	flexible multicast service
FMSID	flexible multicast stream identifier
FOV	field of view
FSM	finite state machine
FT	fast BSS transition
FTAA	fast BSS transition authentication algorithm
FTE	fast BSS transition element
FTO	fast BSS transition originator
GANN	gate announcement
GAS	generic advertisement service
GFSK	Gaussian frequency shift key or keying
GI	guard interval
GMK	group master key
GNonce	group nonce
GPRS	general packet radio service
GPS	Global Positioning System
GTK	group temporal key
GTKSA	group temporal key security association
HC	hybrid coordinator
HCC	hyperbolic congruence code
HCCA	HCF controlled channel access
HCF	hybrid coordination function
HEC	header error check
HEMM	HCCA, EDCA mixed mode
HESSID	homogenous extended service set identifier
HIPERLAN	high-performance radio local area network
HPA	high power amplifier
HR/DSSS	High Rate direct sequence spread spectrum using the long preamble and header
HR/DSSS/PBCC	High Rate direct sequence spread spectrum using the optional packet binary convolutional coding mode and the long preamble and header
HR/DSSS/PBCC/short	High Rate direct sequence spread spectrum using the optional packet binary convolutional coding mode and the optional short preamble and header
HR/DSSS/short	High Rate direct sequence spread spectrum using the optional short preamble and header mode
HT	high throughput
HTC	high throughput control
HT-GF-STF	High-Throughput Greenfield Short Training field

HT-SIG	High-Throughput SIGNAL field
HT-STF	High-Throughput Short Training field
HWMP	hybrid wireless mesh protocol
HWMP SN	hybrid wireless mesh protocol sequence number
IBSS	independent basic service set
ICMP	Internet Control Message Protocol
ICV	integrity check value
IDFT	inverse discrete Fourier transform
IFFT	inverse Fast Fourier Transform
IFS	interframe space
IGTK	integrity group temporal key
IGTKSA	integrity group temporal key security association
IMp	intermodulation protection
INonce	initiator nonce
IPI	idle power indicator
IPN	IGTK packet number
I/Q	in phase and quadrature
IR	infrared
IrDA	infrared data association
ISM	industrial, scientific, and medical
IUT	implementation under test
IV	initialization vector
KCK	EAPOL-Key confirmation key
KDE	key data encapsulation
KDF	key derivation function
KEK	EAPOL-Key encryption key
LAN	local area network
LCI	location configuration information
LDPC	low-density parity check
LED	light-emitting diode
LFSR	linear feedback shift register
LLC	logical link control
L-LTF	Non-HT Long Training field
LME	layer management entity
LNA	low noise amplifier
LRC	long retry count
LSB	least significant bit
L-SIG	Non-HT SIGNAL field
L-STF	Non-HT Short Training field
LTF	Long Training field
MAC	medium access control
MAC_I	initiator mac address
MAC_P	peer mac address
MAF	MCCA access fraction
MBCA	mesh beacon collision avoidance
MBSS	mesh basic service set

MCCA	MCF controlled channel access
MCCAOP	MCF controlled channel access opportunity
MCF	mesh coordination function
MCS	modulation and coding scheme
MDE	Mobility Domain element
MDID	mobility domain identifier
MFB	MCS feedback
MFPC	management frame protection capable
MFPR	management frame protection required
MGTK	mesh group temporal key
MIB	management information base
MIC	message integrity code
MIH	media-independent handover
MIMO	multiple input, multiple output
MLME	MAC sublayer management entity
MLPP	multi-level precedence and preemption
MME	Management MIC element
MMPDU	MAC management protocol data unit
MPDU	MAC protocol data unit
MPM	mesh peering management
MPSP	mesh peer service period
MRQ	MCS request
MSB	most significant bit
MSDU	MAC service data unit
MSGCF	MAC state generic convergence function
MSK	master session key
MTK	mesh temporal key
MUI	message unique identifier
N/A	not applicable
NAI	network access identifier
NAS	network access server
NAV	network allocation vector
NDP	null data packet
NonERP	nonextended rate PHY
NTP	Network Time Protocol (IETF RFC 1305-1992 [B25])
OBSS	overlapping basic service set
OCB	outside the context of a BSS
OFDM	orthogonal frequency division multiplexing
OI	organization identifier
OSI	Open Systems Interconnection (ISO/IEC 7498-1:1994)
OUI	organizationally unique identifier
PAE	port access entity (IEEE Std 802.1X-2004)
PBAC	protected block ack agreement capable
PBCC	packet binary convolutional code
PC	point coordinator
PCF	point coordination function

PCO	phased coexistence operation
PDU	protocol data unit
PER	packet error ratio
PERR	path error
PHB	per-hop behavior
PHY	physical layer
PHYCS	PHY carrier sense
PHYED	PHY energy detection
PICS	protocol implementation conformance statement
PIFS	point (coordination function) interframe space
PLCP	physical layer convergence procedure
PLME	physical layer management entity
PLW	PSDU length word
PMD	physical medium dependent
PMK	pairwise master key
PMK-R0	pairwise master key, first level
PMK-R1	pairwise master key, second level
PMKID	pairwise master key identifier
PMKSA	pairwise master key security association
PN	packet number
PN	pseudonoise (code sequence)
PNonce	peer nonce
PP A-MSDU	payload protected aggregate MAC service data unit
PPDU	PLCP protocol data unit
PPM	pulse position modulation
PREP	path reply
PREQ	path request
PRF	pseudorandom function
PRNG	pseudorandom number generator
PS	power save (mode)
PSAP	public safety answering point
PSDU	PLCP service data unit
PSF	PLCP Signaling field
PSK	preshared key
PSMP	power save multi-poll
PSMP-DTT	power save multi-poll downlink transmission time
PSMP-UTT	power save multi-poll uplink transmission time
PTI	peer traffic indication
PTK	pairwise transient key
PTKSA	pairwise transient key security association
PXU	proxy update
PXUC	proxy update confirmation
QAM	quadrature amplitude modulation
QBPSK	quadrature binary phase shift keying
QLRC	QoS long retry counter
QoS	quality of service

QPSK	quadrature phase shift keying
QSRC	QoS short retry counter
R0KH	PMK-R0 key holder in the Authenticator
R0KH-ID	PMK-R0 key holder identifier in the Authenticator
R1KH	PMK-R1 key holder in the Authenticator
R1KH-ID	PMK-R1 key holder identifier in the Authenticator
RA	receiver address or receiving station address
RADIUS	remote authentication dial-in user service (IETF RFC 2865-2000 [B31])
RANN	root announcement
RAV	resource allocation vector
RCPI	received channel power indicator
RD	reverse direction
RDE	RIC Data element
RDG	reverse direction grant
RF	radio frequency
RFC	request for comments
RIC	resource information container
RIFS	reduced interframe space
RLAN	radio local area network
RPI	receive power indicator
RRB	remote request broker
RSC	receive sequence counter
RSN	robust security network
RSNA	robust security network association
RSNE	Robust Security Network element
RSNI	received signal to noise indicator
RSPI	receiver service period initiated
RSSI	receive signal strength indicator
RTS	request to send
RX	receive or receiver
RXASSI	receive antenna selection sounding indication
RXASSR	receive antenna selection sounding request
S0KH	PMK-R0 key holder in the Supplicant
S0KH-ID	PMK-R0 key holder identifier in the Supplicant
S1KH	PMK-R1 key holder in the Supplicant
S1KH-ID	PMK-R1 key holder identifier in the Supplicant
SA	source address
SAE	simultaneous authentication of equals
SAP	service access point
S-APSD	scheduled automatic power save delivery
SA Query	Security Association Query
SDL	specification and description language
SDU	service data unit
SFD	start frame delimiter
SKCK	STSL key confirmation key
SKEK	STSL key encryption key

SI	service interval
SIFS	short interframe space
SLRC	station long retry count
SM	spatial multiplexing
SME	station management entity
SMK	STSL master key
SMKSA	STSL master key security association
SMT	station management
SNAP	Sub-Network Access Protocol
SNonce	Supplicant nonce
SNR	signal-to-noise ratio
SP	service period
SPA	Supplicant address
SPP A-MSDU	signaling and payload protected aggregate MAC service data unit
SQ	signal quality (PN code correlation strength)
SRC	short retry count
SS	station service
SSID	service set identifier
SSP	subscription service provider
SSPN	subscription service provider network
SSRC	station short retry count
STA	station
STBC	space-time block coding
STK	STSL transient key
STKSA	STSL transient key security association
STSL	station-to-station link
STT	selective translation table
SYNC	synchronization
TA	transmitter address or transmitting station address
TAI	Temps Atomique International (International Atomic Time)
TBTT	target beacon transmission time
TC	traffic category
TCLAS	traffic classification
TDLS	tunneled direct-link setup
TDLS peer PSM	tunneled direct-link setup peer power save mode
TFS	traffic filtering service
TID	traffic identifier
TIE	Timeout Interval element
TIM	traffic indication map
TK	temporal key
TKIP	Temporal Key Integrity Protocol
TMPTT	target measurement pilot transmission time
TOA	time of arrival
TOD	time of departure
TPC	transmit power control
TPK	TDLS Peer Key

TPKSA	TDLS Peer Key Security Association
TPU	TDLS Peer U-APSD
TS	traffic stream
TSC	TKIP sequence counter
TSF	timing synchronization function
TSID	traffic stream identifier
TSN	transition security network
TSPEC	traffic specification
TTAK	TKIP-mixed transmit address and key
TTL	time to live
TTTT	target TIM transmission time
TU	time unit
TX	transmit or transmitter
TXASSI	transmit antenna selection sounding indication
TXASSR	transmit antenna selection sounding request
TXE	transmit enable
TXOP	transmission opportunity
U-APSD	unscheduled automatic power save delivery
UCT	unconditional transition
UESA	unauthenticated emergency service accessible
ULS	Universal Licensing System
U-NII	unlicensed national information infrastructure
UP	user priority
URI	uniform resource identifier
URL	universal resource locator
URN	Uniform Resource Name
UTC	Coordinated Universal Time
VLAN	virtual local area network
VoIP	voice over Internet Protocol (IP)
WLAN	wireless local area network
WDS	wireless distribution system
WEP	wired equivalent privacy
WM	wireless medium
WNM	wireless network management

4. General description

4.1 General description of the architecture

This clause presents the concepts and terminology used within this standard. Specific terms are defined in Clause 3. Illustrations convey key IEEE 802.11 concepts and the interrelationships of the architectural components. IEEE Std 802.11 uses an architecture to describe functional components of an IEEE 802.11 LAN. The architectural descriptions are not intended to represent any specific physical implementation of IEEE Std 802.11.

4.2 How WLAN systems are different

4.2.1 Introduction

Wireless networks have fundamental characteristics that make them significantly different from traditional wired LANs.

4.2.2 Wireless station (STA)

In the design of wired LANs it is implicitly assumed that an address is equivalent to a physical location. In wireless networks, this is not always the case. In IEEE Std 802.11, the addressable unit is a station (STA). The term implies no more than the origin or/and destination of a message. Physical and operational characteristics are defined by modifiers that are placed in front of the term STA. For example, in the case of location and mobility, the addressable units are the fixed STA, the portable STA, and the mobile STA. The STA is a message destination, but not (in general) a fixed location.

A STA might take on multiple distinct characteristics, each of which shape its function. For example, a single addressable unit might simultaneously be a portable STA, a quality-of-service (QoS) STA, a dependent STA, and a hidden STA.

4.2.3 Media impact on design and performance

The PHYs used in IEEE Std 802.11 are fundamentally different from wired media. Thus IEEE 802.11 PHYs:

a) Use a medium that has neither absolute nor readily observable boundaries outside of which STAs with conformant PHY transceivers are known to be unable to receive network frames
b) Are unprotected from other signals that are sharing the medium
c) Communicate over a medium significantly less reliable than wired PHYs
d) Have dynamic topologies
e) Lack full connectivity, and therefore the assumption normally made that every STA can hear every other STA is invalid (i.e., STAs might be “hidden” from each other)
f) Have time-varying and asymmetric propagation properties
g) Might experience interference from logically disjoint IEEE 802.11 networks operating in overlapping areas

Because of limitations on wireless PHY ranges, WLANs intended to cover reasonable geographic distances may be built from basic coverage building blocks. When providing QoS services, the MAC endeavors to provide QoS “service guarantees” within the limitations of the medium properties identified above. In other words, particularly in unlicensed spectrum, true guarantees are often not possible. However, gradations of service are always possible; and in sufficiently controlled environments, QoS guarantees are possible.

4.2.4 The impact of handling mobile STAs

One of the requirements of IEEE Std 802.11 is to handle *mobile* as well as *portable* STAs. A *portable* STA is one that is moved from location to location, but that is only used while at a fixed location. *Mobile* STAs actually access the LAN while in motion.

For technical reasons, it is not sufficient to handle only portable STAs. Propagation effects blur the distinction between portable and mobile STAs; stationary STAs often appear to be mobile due to propagation effects.

Another aspect of mobile STAs is that they may often be battery powered. Hence power management is an important consideration. For example, it cannot be presumed that a STA's receiver is always powered on.

4.2.5 Interaction with other IEEE 802® layers

IEEE Std 802.11 is required to appear to higher layers [logical link control (LLC)] as a wired IEEE 802 LAN. This requires that the IEEE 802.11 network handle STA mobility within the MAC sublayer. To meet reliability assumptions (that LLC makes about lower layers), it is necessary for IEEE Std 802.11 to incorporate functionality that is untraditional for MAC sublayers.

In a robust security network association (RSNA), IEEE Std 802.11 provides functions to protect data frames, IEEE Std 802.1X-2004 provides authentication and a Controlled Port, and IEEE Std 802.11 and IEEE Std 802.1X-2004 collaborate to provide key management. All STAs in an RSNA have a corresponding IEEE 802.1X entity that handles these services. This standard defines how an RSNA utilizes IEEE Std 802.1X-2004 to access these services.

When used to support applications with QoS requirements, each IEEE 802.11 LAN provides a link within an end-to-end QoS environment that may be established between, and managed by, higher layer entities. To handle QoS traffic in a manner comparable to other IEEE 802 LANs, the IEEE 802.11 QoS facility requires the IEEE 802.11 MAC sublayers to incorporate functionality that is not traditional for MAC sublayers. In addition, it may be necessary for certain higher layer management entities to be "WLAN aware" at least to the extent of understanding that the available bandwidth and other QoS characteristics of a WLAN are subject to frequent, and sometimes substantial, dynamic changes due to causes other than traffic load and are outside the direct control of network management entities.

4.2.6 Interaction with non-IEEE-802 protocols

An RSNA utilizes non-IEEE-802 protocols for its authentication and key management (AKM) services. Some of these protocols are defined by other standards organizations, such as the Internet Engineering Task Force (IETF).

4.3 Components of the IEEE 802.11 architecture

4.3.1 General

The IEEE 802.11 architecture consists of several components that interact to provide a WLAN that supports STA mobility transparently to upper layers.

The basic service set (BSS) is the basic building block of an IEEE 802.11 LAN. Figure 4-1 shows two BSSs, each of which has two STAs that are members of the BSS.

It is useful to think of the ovals used to depict a BSS as the coverage area within which the member STAs of the BSS may remain in communication. (The concept of area, while not precise, is often good enough.) This area is called the Basic Service Area (BSA). If a STA moves out of its BSA, it can no longer directly communicate with other STAs present in the BSA.

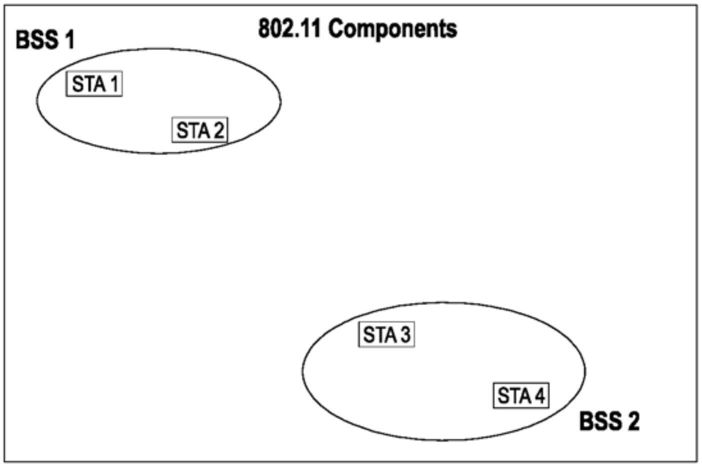


Figure 4-1—BSSs

4.3.2 The independent BSS (IBSS) as an ad hoc network

The IBSS is the most basic type of IEEE 802.11 LAN. A minimum IEEE 802.11 LAN may consist of only two STAs. Since the BSSs shown in Figure 4-1 are simple and lack other components (contrast this with Figure 4-2), the two can be taken to be representative of two IBSSs.

This mode of operation is possible when IEEE 802.11 STAs are able to communicate directly. Because this type of IEEE 802.11 LAN is often formed without preplanning, for only as long as the LAN is needed, this type of operation is often referred to as an *ad hoc network*.

4.3.3 STA membership in a BSS is dynamic

A STA's membership in a BSS is dynamic (STAs turn on, turn off, come within range, and go out of range). To become a member of an infrastructure BSS or an IBSS, a STA joins the BSS using the synchronization procedure described in 10.1.4.5. To start a new mesh BSS or to become a member of a mesh BSS, a STA starts the transmission of Beacons and performs the synchronization maintenance procedure described in 13.13. To access all the services of an infrastructure BSS, a STA becomes "associated." These associations are dynamic and involve the use of the distribution system service (DSS), which is described in 4.4.3. A mesh STA does not become associated as there is no central entity in a mesh BSS (MBSS). Instead, a mesh STA peers with other mesh STAs.

4.3.4 Distribution system (DS) concepts

4.3.4.1 Overview

PHY limitations determine the direct station-to-station distance that may be supported. For some networks this distance is sufficient; for other networks, increased coverage is required.

Instead of existing independently, an infrastructure BSS may also form a component of an extended form of network that is built with multiple BSSs. The architectural component used to interconnect infrastructure BSSs is the DS.

IEEE Std 802.11 logically separates the WM from the distribution system medium (DSM). Each logical medium is used for different purposes, by a different component of the architecture. The IEEE 802.11 definitions neither preclude, nor demand, that the multiple media be either the same or different.

Recognizing that the multiple media are *logically* different is key to understanding the flexibility of the architecture. The IEEE 802.11 LAN architecture is specified independently of the physical characteristics of any specific implementation.

The DS enables mobile device support by providing the logical services necessary to handle address to destination mapping and seamless integration of multiple BSSs.

An access point (AP) is any entity that has STA functionality and enables access to the DS, via the WM for associated STAs.

Figure 4-2 adds the DS, DSM and AP components to the IEEE 802.11 architecture picture.

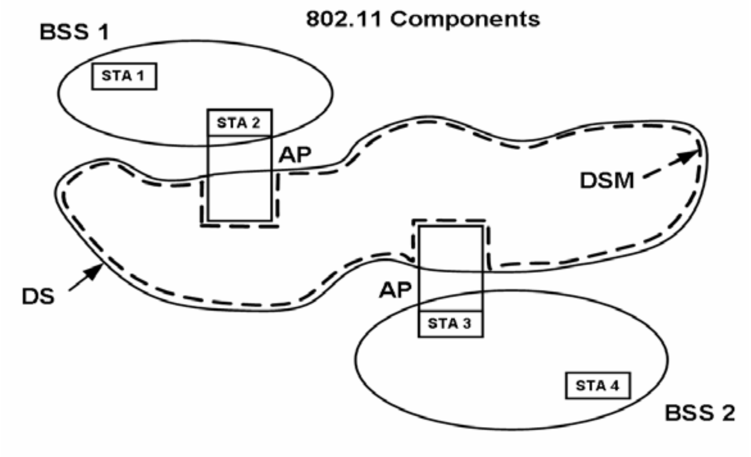


Figure 4-2—DSs and APs

Data move between a BSS and the DS via an AP. Note that all APs are also STAs; thus they are addressable entities. The addresses used by an AP for communication on the WM and on the DSM are not necessarily the same.

Data sent to the AP's STA address by one of the STAs associated with it are always received at the uncontrolled port for processing by the IEEE 802.1X port access entity. In addition, if the controlled port is authorized, these frames conceptually transit the DS.

4.3.4.2 Extended service set (ESS): The large coverage network

The DS and infrastructure BSSs allow IEEE Std 802.11 to create a wireless network of arbitrary size and complexity. IEEE Std 802.11 refers to this type of network as the ESS network. An ESS is the union of the infrastructure BSSs with the same SSID connected by a DS. The ESS does not include the DS.

The key concept is that the ESS network appears the same to an LLC layer as an IBSS network. STAs within an ESS may communicate and mobile STAs may move from one BSS to another (within the same ESS) transparently to LLC.

Owing to its distributed nature, a mesh BSS (MBSS) has no central entity like the AP of an infrastructure BSS. Instead, an MBSS forms a single set of independent mesh STAs. This set is indivisible and cannot be further unified. The ESS concept does not apply to the MBSS. However, it is possible to use a Mesh BSS as all or part of the DS that connects an ESS.

Nothing is assumed by IEEE Std 802.11 about the relative physical locations of the BSSs in Figure 4-3.

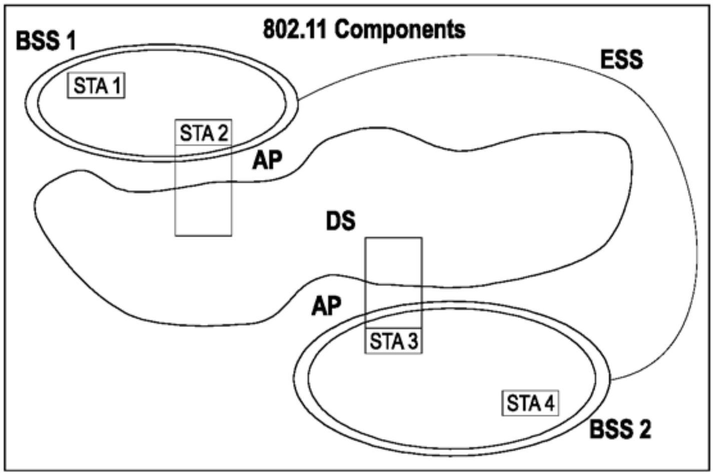


Figure 4-3—ESS

All of the following are possible

a) The BSSs partially overlap. This is commonly used to arrange contiguous coverage within a physical volume.
b) The BSSs could be physically disjoint. Logically there is no limit to the distance between BSSs.
c) The BSSs are physically collocated. This could be done to provide redundancy.
d) One (or more) IBSS or ESS networks are physically present in the same location as one (or more) ESS networks. This could arise for a number of reasons. Some examples are when an IBSS network is operating in a location that also has an ESS network, when physically overlapping IEEE 802.11 networks have been set up by different organizations, and when two or more different access and security policies are needed in the same location.

4.3.4.3 Robust security network association (RSNA)

An RSNA defines a number of security features in addition to wired equivalent privacy (WEP) and IEEE 802.11 authentication. These features include the following:

- Enhanced authentication mechanisms for STAs
- Key management algorithms
- Cryptographic key establishment
- Enhanced data cryptographic encapsulation mechanisms, such as Counter mode with Cipher-block chaining Message authentication code Protocol (CCMP), and, optionally, Temporal Key Integrity Protocol (TKIP).
- Fast basic service set (BSS) transition (FT) mechanism
- Enhanced cryptographic encapsulation mechanisms for robust management frames

An RSNA may rely on components external to the IEEE 802.11 architecture.

The first component is an IEEE 802.1X port access entity (PAE). PAEs are present on all STAs in an RSNA and control the forwarding of data to and from the medium access control (MAC). An AP always implements the Authenticator PAE and Extensible Authentication Protocol (EAP) Authenticator roles, and a non-AP STA always implements the Supplicant PAE and EAP peer roles. In an IBSS each STA implements both the Authenticator PAE and Supplicant PAE roles and both EAP Authenticator and EAP peer roles.

A second component is the Authentication Server (AS). The AS may authenticate the elements of the RSNA itself, i.e., the STAs may provide material that the RSNA elements use to authenticate each other. The AS communicates through the IEEE 802.1X Authenticator with the IEEE 802.1X Supplicant on each STA, enabling the STA to be authenticated to the AS and vice versa. An RSNA depends upon the use of an EAP method that supports mutual authentication of the AS and the STA, such as those that meet the requirements in IETF RFC 4017. In certain applications, the AS may be integrated into the same physical device as the AP, or into a STA in an IBSS.

In some applications, there is no need for a PAE or AS, and a STA and AP, or two STAs in an IBSS, or two mesh STAs in an MBSS, may authenticate each other using a password.

An RSNA using fast BSS transition relies on an external protocol to distribute keys between the pairwise master key (PMK) R0 key holder (R0KH) and PMK-R1 key holder (R1KH) Authenticator components. The requirements for this protocol are described in 12.2.2.

4.3.5 Area concepts

For wireless PHYs, well-defined coverage areas simply do not exist. Propagation characteristics are dynamic and unpredictable. Small changes in position or direction may result in dramatic differences in signal strength. Similar effects occur whether a STA is stationary or mobile (as moving objects may impact station-to-station propagation).

Figure 4-4 shows a signal strength map for a simple square room with a standard metal desk and an open doorway. Figure 4-4 is a static snapshot; the propagation patterns change dynamically as STAs and objects in the environment move. In Figure 4-4 the dark (solid) blocks in the lower left are a metal desk and there is a doorway at the top right of the figure. The figure indicates relative differences in field strength with different intensities and indicates the variability of field strength even in a static environment. The difference between the greatest signal strength and the least signal strength in Figure 4-4 is 50 dB.

While the architecture diagrams show sharp boundaries for BSSs, this is an artifact of the pictorial representation, not a physical reality. Because dynamic three-dimensional field strength pictures are difficult to draw, well-defined shapes are used by IEEE 802.11 architectural diagrams to represent the coverage of a BSS.

Further description difficulties arise when attempting to describe collocated coverage areas. Consider Figure 4-5, in which STA 6 could belong to BSS 2 or BSS 3.

While the concept of sets of STAs is correct, it is often convenient to talk about areas. For many topics the concept of area is sufficient. *Volume* is a more precise term than area, though still not technically correct. For historical reasons and convenience, this standard uses the common term *area*.

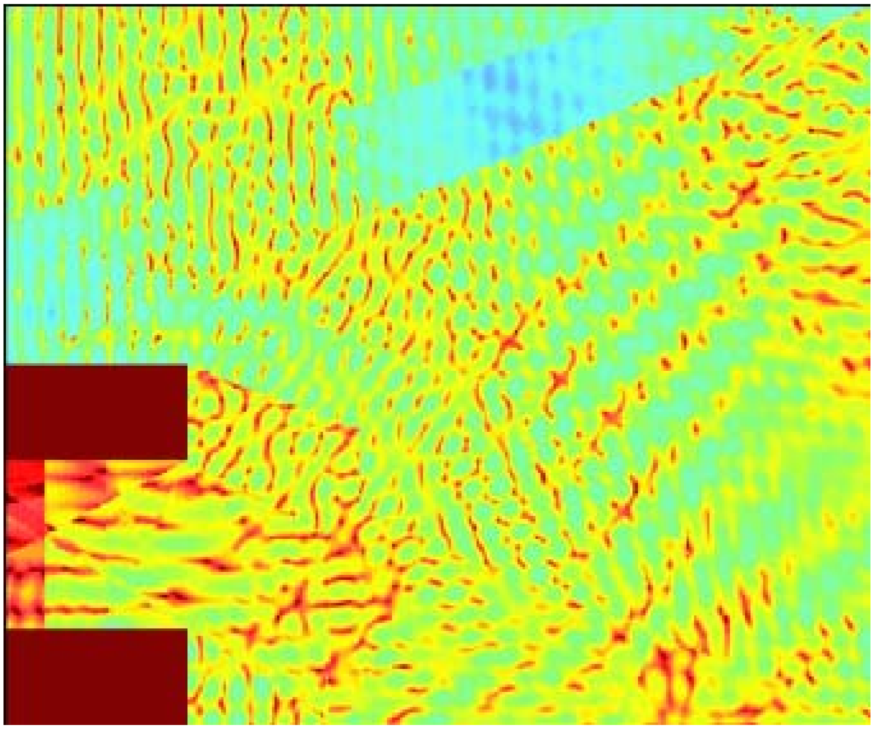


Figure 4-4—A representative signal intensity map

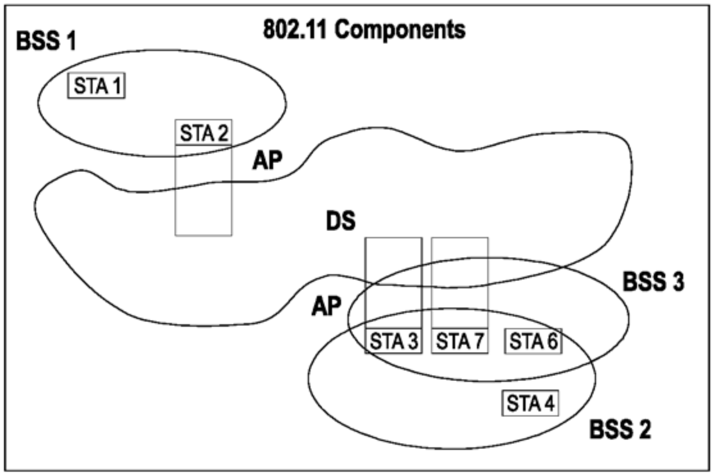


Figure 4-5—Collocated coverage areas

4.3.6 Integration with wired LANs

To integrate the IEEE 802.11 architecture with a traditional wired LAN, a final *logical* architectural component is introduced—a *portal*.

A portal is the logical point at which MSDUs from an integrated non-IEEE-802.11 LAN enter the IEEE 802.11 DS. For example, a portal is shown in Figure 4-6 connecting to a wired IEEE 802 LAN.

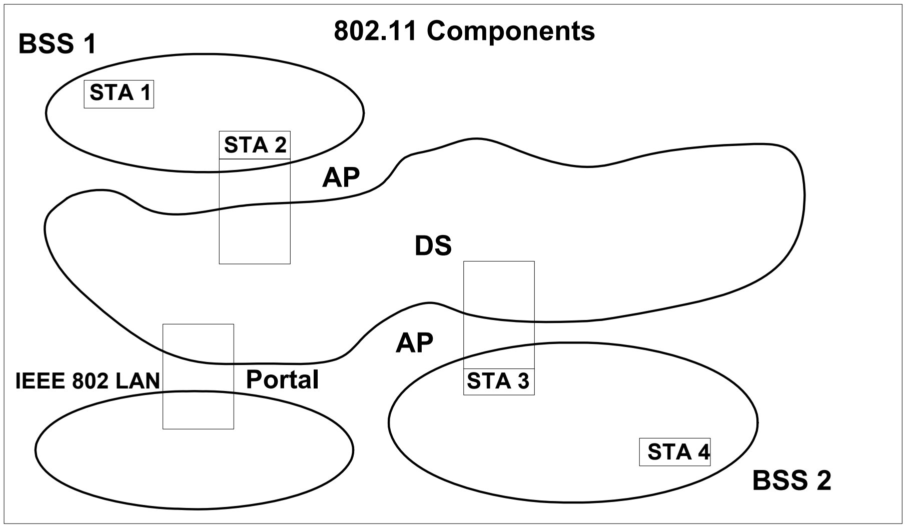


Figure 4-6—Connecting to other IEEE 802 LANs

All data from non-IEEE-802.11 LANs enter the IEEE 802.11 architecture via a portal. The portal is the logical point at which the integration service is provided. The integration service is responsible for any addressing changes that might be required when MSDUs pass between the DS and the integrated LAN. It is possible for one device to offer both the functions of an AP and a portal.

4.3.7 QoS BSS: The QoS network

The IEEE 802.11 QoS facility provides MAC enhancements to support LAN applications with QoS requirements. The QoS enhancements are available to QoS STAs associated with a QoS access point in a QoS BSS. A subset of the QoS enhancements is available for use between STAs that are members of the same QoS IBSS. Similarly, a subset of the QoS enhancements is available for use between neighbor peer mesh STAs. A mesh BSS is one type of QoS BSS and it is described in 4.3.15. Because a nonmesh QoS STA implements a superset of STA functionality, as defined in this standard, the STA might associate with a non-QoS access point in a non-QoS BSS, to provide non-QoS MAC data service when there is no QoS BSS with which to associate. As a mesh STA does not implement the necessary service, the mesh STA does not associate with any access point.

The enhancements that distinguish QoS STAs from non-QoS STAs and QoS APs from non-QoS APs are collectively termed the *QoS facility*. Which of the QoS-specific mechanisms a QoS STA supports might vary among QoS implementations, as well as between QoS STAs and QoS APs, over ranges specified in subsequent clauses. All service primitives, frame formats, coordination function and frame exchange rules, and management interface functions except for the Block Acknowledgment (Block Ack) function, direct-link setup (DLS), and automatic power save delivery (APSD) are part of the core QoS facilities. A QoS STA or QoS AP implements those core QoS facilities necessary for its QoS functions to interoperate with other QoS STAs. Functions such as the Block Ack, DLS, and APSD are separate from the core QoS facilities; and the presence of these functions is indicated by STAs separately from the core QoS facilities.

For infrastructure BSS and IBSS, this standard provides two mechanisms for the support of applications with QoS requirements.

The first mechanism, designated the *enhanced distributed channel access* (EDCA), delivers traffic based on differentiating user priorities (UPs). This differentiation is achieved by varying the following for different UP values:

— Amount of time a STA senses the channel to be idle before backoff or transmission, or

- The length of the contention window to be used for the backoff, or
- The duration a STA may transmit after it acquires the channel.

These transmissions may also be subject to certain channel access restrictions in the form of admission control. Details of this mechanism are provided in 9.19.2.

The second mechanism, designated the *hybrid coordination function* (HCF) *controlled channel access* (HCCA), allows for the reservation of transmission opportunities (TXOPs) with the hybrid coordinator (HC). A STA based on its requirements requests the HC for TXOPs, both for its own transmissions as well as for transmissions from the AP to itself.[16] The request is initiated by the station management entity (SME) of the STA. The HC, which is collocated at the AP, either accepts or rejects the request based on an admission control policy. If the request is accepted, the HC schedules TXOPs for both STAs (both the AP and the non-AP STA). For transmissions from the non-AP STA, the HC polls the STA based on the parameters supplied by the STA at the time of its request. For transmissions to the STA, the AP directly obtains TXOPs from the collocated HC and delivers the queued frames to the STA, again based on the parameters supplied by the STA. Details of the mechanism are provided in 9.19.3 and 10.4. This mechanism may be used for applications such as voice and video, which may need periodic service from the HC. If the application constraints dictate the use of this mechanism, the application initiates this mechanism by using the management service primitives.

Non-QoS STAs may associate in a QoS BSS, if allowed to associate by the AP. All individually addressed frames that are sent to non-QoS STAs by an AP do not use the frame formats associated with the QoS facility.

A QoS STA associated in a non-QoS BSS acts as a non-QoS STA.

4.3.8 Wireless LAN Radio Measurements

4.3.8.1 General

Wireless LAN (WLAN) Radio Measurements enable STAs to understand the radio environment in which they exist. WLAN Radio Measurements enable STAs to observe and gather data on radio link performance and on the radio environment. A STA may choose to make measurements locally, request a measurement from another STA, or may be requested by another STA to make one or more measurements and return the results. Radio Measurement data is made available to STA management and upper protocol layers where it may be used for a range of applications. The measurements enable adjustment of STA operation to better suit the radio environment. The Radio Measurement service includes measurements that extend the capability, reliability, and maintainability of WLANs by providing standard measurements across vendors, and the service provides the resulting measurement data to upper layers in the communications stack.

In addition to featuring standard measurements and delivering measurement information to upper layers, there are applications that require quantifiable radio environment measurements in order to attain the necessary performance levels. These applications include VoIP, video over IP, location based applications, as well as applications requiring mitigation of harsh radio environments (multifamily dwellings, airplanes, factories, municipalities, etc.). Radio Measurements address most of the existing issues in using unlicensed radio spectrum to meet the requirements of these emerging technologies.

To address the mobility requirements of technologies, such as VoIP and video streaming, Radio Measurements, such as Channel Load request/report and the Neighbor request/report, may be used to collect transition information, which can drastically speed up handoffs between BSSs within the same ESS. By accessing and using this information, the STAs (either in APs or in non-AP STAs) can make intelligent

[16] In the case of downlink traffic streams (TSs).

decisions about the most effective way to utilize the available spectrum, power, and bandwidth for their communications.

The request/report measurements are as follows:

- beacon
- frame
- channel load
- noise histogram
- STA statistics
- location configuration information (LCI)
- neighbor report
- link measurement
- transmit stream/category measurement

The request-only mechanism is as follows:

- measurement pause

The report-only mechanism is as follows:

- measurement pilot

These measurement mechanisms provide the capability for a STA to manage and query its radio environment, and to make appropriate assessments about its health and efficiency. It is the first step in making WLAN smart and capable of making appropriate decisions for fast transition, for mesh connectivity, and for managing the radio environment for all wireless devices.

4.3.8.2 Beacon

The Beacon request/report pair enables a STA to request from another STA a list of APs whose beacons it can receive on a specified channel or channels. This measurement may be done by active mode (like active scan), passive mode (like passive scan), or beacon table modes. If the measurement request is accepted and is in passive mode, a duration timer is set. Then the measuring STA monitors the requested channel; measures beacon, probe response, and measurement pilot power levels (received channel power indicator (RCPI)); and logs all beacons, probe responses, and measurement pilots received within the measurement duration. If the measurement request is in active mode, the measuring STA sends a probe request on the requested channel at the beginning of the measurement duration; then monitors the requested channel; measures beacon, probe response, and measurement pilot power levels (RCPI); and logs all beacons, probe responses, and measurement pilots received within the measurement duration. If the request is beacon table mode, then the measuring STA returns a Beacon Report containing the current contents of any stored beacon information for any supported channel with the requested service set identifier (SSID) and basic service set identifier (BSSID) without performing additional measurements.

4.3.8.3 Measurement Pilot

The Measurement Pilot frame provides a subset of the information provided in a Beacon frame, is smaller than a Beacon, and is transmitted more often than a Beacon. The purpose of the Measurement Pilot frame is to assist a STA with scanning.

4.3.8.4 Frame

The frame request/report pair returns a picture of all the channel traffic and a count of all the frames received at the measuring STA. For each unique Transmitter Address, the STA reports the Transmitter Address, number of frames received from this transmitter, average power level (RCPI) for these frames, and BSSID of the transmitter.

4.3.8.5 Channel load

The channel load request/report pair returns the channel utilization measurement as observed by the measuring STA.

4.3.8.6 Noise histogram

The noise histogram request/report pair returns a power histogram measurement of non-IEEE 802.11 noise power by sampling the channel when virtual carrier sense indicates idle and the STA is neither transmitting nor receiving a frame.

4.3.8.7 STA statistics

The STA statistics request/report pair returns groups of values for STA counters and for BSS Average Access Delay. The STA counter group values include transmitted fragment counts, group addressed transmitted frame counts, failed counts, retry counts, multiple retry counts, frame duplicate counts, Request to Send (RTS) success counts, RTS failure counts, Acknowledgement (ACK) failure counts, received fragment counts, group addressed received frame counts, FCS error counts, and transmitted frame counts. BSS Average Access Delay group values include AP average access delay, average access delay for each access category, associated STA count, and channel utilization.

4.3.8.8 Location

The Location request/report pair returns a requested location in terms of latitude, longitude, and altitude. It includes types of altitude such as floors and permits various reporting resolutions. The requested location may be the location of the requestor (e.g., Where am I?) or the location of the reporting STA (e.g., Where are you?)

4.3.8.9 Measurement pause

The measurement pause request is defined, but no report comes back from this request. The measurement pause permits the inclusion of a quantified delay between the execution of individual measurements that are provided in a series within a measurement request frame. The measurement pause used as the last measurement in a frame provides control of the measurement period when measurement request frames are to be repeated.

4.3.8.10 Neighbor report

The neighbor report request is sent to an AP, which returns a neighbor report containing information about known neighbor APs that are candidates for a service set transition. Neighbor reports contain information from the table dot11RMNeighborReportTable in the MIB concerning neighbor APs. This request/report pair enables a STA to gain information about the neighbors of the associated AP to be used as potential roaming candidates.

4.3.8.11 Link measurement

The link measurement request/report exchange provides measurements of the RF characteristics of a STA-to-STA link. This measurement indicates the instantaneous quality of a link.

4.3.8.12 Transmit stream/category measurement

The Transmit Stream/Category measurement is a request/report pair that enables a QoS STA to inquire of a peer QoS STA the condition of an ongoing traffic stream between them. The Transmit Stream/Category Measurement Report provides the transmit-side performance metrics for the measured traffic stream. Trigger conditions included in the Transmit Stream/Category Measurement Request may initiate triggered Transmit Stream/Category Measurement Reports upon detection of the trigger condition.

4.3.9 Operation in licensed frequency bands

4.3.9.1 General

IEEE 802.11 devices can operate on frequencies that are licensed by national regulatory bodies. Although this standard has been generalized so that it is independent of license type, band, and country of operation, only the bands and associated regulations listed in Annex D have been specifically considered.

4.3.9.2 Dynamic STA enablement (DSE) in licensed bands

The DSE operating procedures are used to automate the channel provisioning and regulatory controls needed for unregistered IEEE 802.11 STAs to operate as dependent STAs in licensed spectrum.[17]

4.3.9.3 Contention-Based Protocol (CBP) in nonexclusively licensed bands

The granting of licenses on a nonexclusive, uncoordinated basis in the same area leads to the possibility of overlapping networks. When overlapping networks cause co-channel interference, regulations, such as those governing the 3650 MHz band in the United States, require the use of a CBP "by which a transmitter provides reasonable opportunities for other transmitters to operate."[18] IEEE 802.11 carrier sense multiple access with collision avoidance (CSMA/CA) is suitable for this purpose in most situations.

4.3.9.4 Using DSE STA identification to resolve interference

When CSMA/CA is not able to sufficiently sense the presence of another licensee's STA (i.e., a hidden STA) or if a secondary licensee causes inference to a primary licensee, the licensee is obliged to resolve complaints that result from interference caused by any STA under its control (including dependent STAs). In order to facilitate the interference resolution processes, all STAs operating in nonexclusively licensed spectrum use the DSE STA and location information procedures.

The STA identification and location information procedures are inherently tied because, by default, registered STAs broadcast their actual location as their unique identifier. Dependent STAs broadcast the location of the STA that has enabled them as well as a unique code selected by the licensee. This method puts a victim of the interference in contact with the party responsible for rectifying the problem, and, at the same time, it protects the privacy of the dependent STA's operator.

4.3.9.5 Further coexistence enhancements in nonexclusively licensed bands

While not explicitly required to meet specific rules, a number of optional IEEE 802.11 mechanisms, when used together, are able to meet general requirements for spectrum sharing, incumbent detection, and other cognitive radio functions in licensed bands. The specific mechanisms for each band are detailed in E.2.

[17] In some licensed frequency bands, wireless equipment can be owned and operated by individuals who do not hold a license. In such instances, devices are permitted to operate only if they are either communicating with, or receiving permission to transmit from, a STA that is maintained by a licensed operator. The Japanese 4.9 GHz band and the U.S. 4.94–4.99 GHz public safety band are examples in which IEEE 802.11 STAs operate under such arrangements.

[18] Definition of CBP from FCC 05-56, Report and Order and Memorandum Opinion and Order, clause 58.

4.3.10 High-throughput (HT) STA

The IEEE 802.11 HT STA provides PHY and MAC features that can support a throughput of 100 Mb/s and greater, as measured at the MAC data service access point (SAP). An HT STA supports HT features as identified in Clause 9 and Clause 20. An HT STA operating in the 5 GHz band supports transmission and reception of frames that are compliant with mandatory PHY specifications as defined in Clause 18. An HT STA operating in the 2.4 GHz band supports transmission and reception of frames that are compliant with mandatory PHY specifications as defined in Clause 17 and Clause 19. An HT STA is also a QoS STA. The HT features are available to HT STAs associated with an HT AP in a BSS. A subset of the HT features is available for use between two HT STAs that are members of the same IBSS. Similarly, a subset of the HT features is available for use between two HT STAs that have established mesh peering (see 8.4.2.58 for details).

An HT STA has PHY features consisting of the modulation and coding scheme (MCS) set described in 20.3.5 and physical layer convergence procedure (PLCP) protocol data unit (PPDU) formats described in 20.1.4. Some PHY features that distinguish an HT STA from a non-HT STA are referred to as multiple input, multiple output (MIMO) operation; spatial multiplexing (SM); spatial mapping (including transmit beamforming); space-time block coding (STBC); low-density parity check (LDPC) encoding; and antenna selection (ASEL). The allowed PPDU formats are non-HT format, HT-mixed format, and HT-greenfield format. The PPDUs may be transmitted with 20 MHz or 40 MHz bandwidth.

An HT STA has MAC features that include frame aggregation, some Block Ack features, power save multi-poll (PSMP) operation, reverse direction (RD), and protection mechanisms supporting coexistence with non-HT STAs.

4.3.11 STA transmission of data frames outside the context of a BSS

In addition to defining procedures for STA communication within a BSS, this standard also allows a STA that is not a member of a BSS to transmit data frames. Such data frames are defined as being transmitted outside the context of a BSS. A STA transmits a data frame outside the context of a BSS only if dot11OCBActivated is true.

NOTE—The specific frame subtypes that a STA is allowed to send when it has dot11OCBActivated true are specified in 10.20.

When dot11OCBActivated is true, a data frame can be sent to either an individual or a group destination MAC address. This type of communication is only possible between STAs that are able to communicate directly over the wireless medium. It allows immediate communication, avoiding the latency associated with establishing a BSS. When dot11OCBActivated is true, a STA is not a member of a BSS and it does not utilize the IEEE 802.11 authentication, association, or data confidentiality services. This capability is particularly well-suited for use in rapidly varying communication environments such as those involving mobile STAs in which the interval over which the communication exchanges take place may be of very short-duration (e.g., on the order of tens or hundreds of milliseconds). Since IEEE 802.11 MAC sublayer authentication services are not used when dot11OCBActivated is true, any required authentication services are provided by the station management entity (SME) or by applications outside of the MAC sublayer.

A STA whose MIB does not include the dot11OCBActivated attribute operates as if the attribute is false.

Communication of data frames when dot11OCBActivated is true might take place in a frequency band that is dedicated for its use, and such a band might require licensing depending on the regulatory domain. A STA for which dot11OCBActivated is true initially transmits and receives on a channel known in advance, either through regulatory designation or some other out-of-band communication. A STA's SME determines PHY layer parameters, as well as any changes in the operating channel, e.g., using information obtained via out-of-band communication or over-the-air frame exchange. When dot11OCBActivated is true, a sending STA sets the BSSID field to the wildcard BSSID value (see 8.2.4.3.4).

The Vendor Specific frame (see 8.5.6) provides one means for STAs to exchange management information prior to communicating data frames outside the context of a BSS.

4.3.12 Tunneled direct-link setup

Tunneled direct-link setup (TDLS) is characterized by the use of signaling frames that are encapsulated in data frames so that the signaling frames are transmitted through an AP transparently. Therefore, unlike with DLS, the AP does not need to be direct-link aware, nor does it have to support the same set of capabilities that are used on the direct link, in order for TDLS to be used. To allow a STA to enter a TDLS power save mode, TDLS provides two power save mechanisms: TDLS peer U-APSD and TDLS peer PSM. TDLS allows STAs to use the TDLS Peer Key Handshake to provide data confidentiality and message authentication. STAs that set up a TDLS direct link remain associated with the AP and transmit frames directly to the other TDLS peer STA.

4.3.13 Wireless network management

4.3.13.1 Overview

Wireless network management (WNM) enables STAs to exchange information for the purpose of improving the overall performance of the wireless network. STAs use WNM protocols to exchange operational data so that each STA is aware of the network conditions, allowing STAs to be more cognizant of the topology and state of the network. WNM protocols provide a means for STAs to be aware of the presence of collocated interference, and enable STAs to manage RF parameters based on network conditions.

In addition to providing information on network conditions, WNM also provides a means to exchange location information, provide support for the multiple BSSID capability on the same wireless infrastructure, support efficient delivery of group addressed frames, and enable a WNM-Sleep mode in which a STA can sleep for long periods of time without receiving frames from the AP.

The WNM service includes the following:

- BSS Max idle period management
- BSS transition management
- Channel usage
- Collocated interference reporting
- Diagnostic reporting
- Directed multicast service (DMS)
- Event reporting
- Flexible multicast service (FMS)
- Location services
- Multicast diagnostic reporting
- Multiple BSSID capability
- Proxy ARP
- QoS traffic capability
- SSID list
- Triggered STA statistics
- TIM broadcast
- Timing measurement
- Traffic filtering service
- U-APSD Coexistence

- WNM-Notification
- WNM-Sleep mode

4.3.13.2 BSS Max idle period management

BSS Max idle period management enables an AP to indicate a time period during which the AP does not disassociate a STA due to nonreceipt of frames from the STA. This supports improved STA power saving and AP resource management.

4.3.13.3 BSS transition management

BSS transition management enables an AP to request non-AP STAs to transition to a specific AP, or to indicate to a non-AP STA a set of preferred APs, due to network load balancing or BSS Termination.

4.3.13.4 Channel usage

Channel usage information is provided by the AP to the non-AP STA to recommend channels for noninfrastructure networks or an off-channel TDLS direct link. The non-AP STAs can use the channel usage information as part of channel selection processing for a noninfrastructure network or an off-channel TDLS direct link.

4.3.13.5 Collocated interference reporting

Collocated interference reporting enables the requesting STA to obtain information on interference due to collocated radios at the reporting STA. The requesting STA can use that information to schedule its transmissions to minimize the effects of the interference.

4.3.13.6 Diagnostic reporting

Diagnostic requests enable a STA to request a non-AP STA to report on information that may be helpful in diagnosing and resolving problems with the WLAN network. Diagnostic reports include information on hardware, configuration, and STA capabilities.

4.3.13.7 Directed multicast service (DMS)

The DMS enables a non-AP STA to request the AP to transmit group addressed frames destined to the requesting STA as individually addressed frames.

4.3.13.8 Event reporting

Event requests enable a STA to request a non-AP STA to send particular real-time event messages. The types of events include Transition, RSNA, WNM Log, and Peer-to-Peer Link events. A transition event is transmitted after a non-AP STA successfully completes a BSS Transition. Transition events are used to diagnose transition performance problems. An RSNA event report describes the type of Authentication used for the RSNA. RSNA events are used to diagnose security and authentication performance problems. A WNM Log event report enables a non-AP STA to transmit a set of WNM Log event messages to the requesting STA. WNM Log event reports are used to access the contents of a STA's WNM Log. A Peer-to-Peer Link event report enables a non-AP STA to inform the requesting STA that a Peer-to-Peer link has been established. Peer-to-Peer Link event reports are used to monitor the use of Peer-to-Peer links in the network.

4.3.13.9 FMS

The flexible multicast service enables a non-AP STA to request an alternate delivery traffic indication map (DTIM) delivery interval for one or more sets of group addressed streams that the non-AP STA receives. This enables the non-AP STA to wake up at the alternate DTIM interval rather than every DTIM and enables significant power saving when a non-AP STA receives group addressed traffic. The FMS also enables a STA to establish a data rate and delivery interval for group addressed traffic higher than the minimum data rate available.

Delivery of group addressed data to power saving STAs using a DTIM beacon is described in 10.2.1.4.

4.3.13.10 Location services

Location Configuration Request and Response frames enable STAs to configure a collection of location related parameters for Location Track Notification frames. The AP can indicate that it can provide location data to support applications such as emergency services. Location services also provide the ability for STAs to exchange location information using Radio Measurement Request and Report frames. The protocol supports exchange-by-value and exchange-by-reference mechanisms. The location value can be exchanged in Geospatial (LCI) and Civic formats. The location reference is a URL that defines from where the location value is retrieved.

4.3.13.11 Multicast diagnostic reporting

Multicast diagnostic reports enable a non-AP STA to report statistics for multicast traffic it received from a transmitting STA. This can be used by an AP to measure quality of multicast reception by a non-AP STA.

4.3.13.12 Multiple BSSID capability

The Multiple BSSID capability enables the advertisement of information for BSSIDs using a single Beacon or Probe Response frame instead of multiple Beacon and Probe Response frames, each corresponding to a single BSSID. The Multiple BSSID capability also enables the indication of buffered frames for multiple BSSIDs using a single TIM element in a single beacon.

4.3.13.13 Proxy ARP

The Proxy ARP capability enables an AP to indicate that the non-AP STA does not receive ARP frames. The Proxy ARP capability enables the non-AP STA to remain in power save for longer periods of time.

4.3.13.14 QoS traffic capability

QoS traffic capability procedures enable the QoS STA to indicate that it is capable of transmitting traffic belonging to the corresponding user priority (UP) from applications that require generation of such traffic. The QoS Traffic Capability might be used for example as an input to estimate the blocking probability of a voice application based on the number of voice capable non-AP STAs.

4.3.13.15 SSID list

The SSID List element enables the non-AP STA to request information on a list of SSIDs. This is intended to reduce the number of Probe Request frames sent by the non-AP STA.

4.3.13.16 Triggered STA statistics

The Triggered STA Statistics reporting capability enables generation of a STA statistics report (see 4.3.8.9) when the statistics of interest reach a predefined threshold.

4.3.13.17 TIM broadcast

The TIM broadcast protocol defines a mechanism to enable a STA to receive an indication of buffered individually addressed traffic, independent of the Beacon frame, reducing the wake time of the STA.

4.3.13.18 Timing measurement

Timing Measurement frames allow a recipient STA to accurately measure the offset of its clock relative to a clock in the sending STA. With the regular transfer of Timing Measurement frames from one STA to another, it is possible for the recipient STA to track changes in the offset of its clock with respect to the sending STA over time and thus detect and compensate for any drift between the clocks.

4.3.13.19 Traffic filtering service

Traffic filtering is a service that may be provided by an AP to its associated STAs, where the AP examines MSDUs and management frames destined for a STA. The AP determines if any of those frames match a specific set of traffic filters that may be enabled at the AP per the request of the STA. Individually addressed frames that do not match any of the traffic filters in the set are discarded. Individually addressed frames that do match at least one of the set of the enabled traffic filters are delivered to the STA. The STA may also negotiate to have a notification frame sent prior to the delivery of the frame matching the traffic filter.

4.3.13.20 U-APSD Coexistence

The U-APSD Coexistence capability enables the non-AP STA to indicate a requested transmission duration to the AP for use of U-APSD service periods. Use of the transmission duration enables the AP to transmit frames during the service period and improve the likelihood that the non-AP STA receives the frames when the non-AP STA is experiencing interference. The U-APSD Coexistence capability reduces the likelihood that the AP transmits frames during the service period that are not received successfully.

4.3.13.21 WNM-Notification

WNM-Notification provides a mechanism for a STA to notify another STA of a management event. One event is defined: firmware update notification.

4.3.13.22 WNM-Sleep mode

WNM-Sleep mode is an extended power save mode for non-AP STAs in which a non-AP STA need not listen for every DTIM Beacon frame, and need not perform GTK/IGTK updates. WNM-Sleep mode enables a non-AP STA to signal to an AP that it will be sleeping for a specified length of time. This enables a non-AP STA to reduce power consumption and remain associated while the non-AP STA has no traffic to send to or receive from the AP.

4.3.14 Subscription service provider network (SSPN) interface

An AP can interact with external networks using a SSPN interface for the purpose of authenticating users and provisioning services, as shown in Figure 4-7. The exchange of authentication and provisioning information between the SSPN and the AP passes transparently through the Portal. The protocol used to exchange this information is outside the scope of this standard. The logical SSPN interface provides the means for an AP to consult an SSPN for authenticating and authorizing a specific non-AP STA and to report statistics and status information to the SSPN. Authentication and provisioning information for non-AP STAs received from the SSPN are stored in the AP management information base (MIB) and are used to limit layer-2 services provided to that non-AP STA. Detailed interactions describing the SSPN interface are provided in 10.24.5.

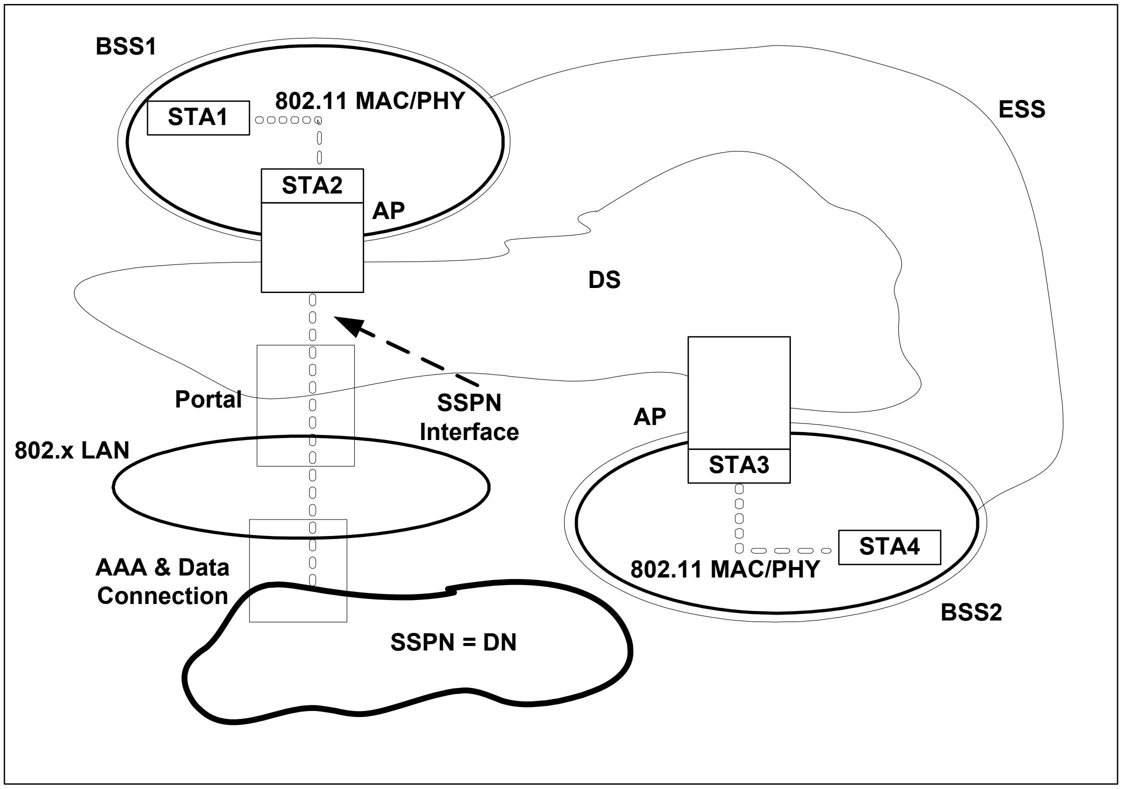


Figure 4-7—SSPN interface service architecture

The SSPN interface provides the non-AP STA access to the services provisioned in the SSPN via the currently associated BSS. SSPN access may involve virtual local area network (VLAN) mapping or tunnel establishment that are transparent to the non-AP STA and outside the scope of this standard. The SSPN interface also allows the non-AP STA to access services in destination networks (DNs) other than the SSPN. An example of a DN other than SSPN is the provision of Internet access via the IEEE 802 LAN, or an intermediary network that connects the IEEE 802.11 infrastructure and the SSPN.

NOTE—The SSPN Interface Service is not supported in an IBSS.

4.3.15 Mesh BSS: IEEE 802.11 wireless mesh network

4.3.15.1 General

The IEEE 802.11 mesh facility provides MAC enhancements to support wireless LAN mesh topologies. The mesh facilities are available to mesh STAs that belong to a mesh BSS (MBSS). For a mesh STA that has not become a member of an MBSS, only the mesh discovery service is available. The enhancements that distinguish mesh STAs from nonmesh STAs are collectively termed the “mesh facility.” The mesh-specific mechanisms vary among implementations.

4.3.15.2 Overview of the mesh BSS

A mesh BSS is an IEEE 802.11 LAN consisting of autonomous STAs. Inside the mesh BSS, all STAs establish wireless links with neighbor STAs to mutually exchange messages. Further, using the multi-hop capability, messages can be transferred between STAs that are not in direct communication with each other over a single instance of the wireless medium. From the data delivery point of view, it appears as if all STAs in a mesh BSS are directly connected at the MAC layer even if the STAs are not within range of each other. The multi-hop capability enhances the range of the STAs and benefits wireless LAN deployments.

STAs in a mesh BSS might be sources, sinks, or propagators of traffic; some mesh STAs might only propagate traffic for other STAs. As described in 4.3.15.4, a mesh BSS might have interfaces to external networks and can be a DSM for infrastructure BSSs.

Within a mesh BSS, STAs utilize the mesh coordination function (MCF) to access the channel. MCF is based on the core QoS facilities specified in 4.3.7, and a mesh BSS is categorized as one type of QoS BSS. MCF is described in 9.20.

4.3.15.3 Mesh STA

A STA that belongs to a mesh BSS is termed a "mesh station" (mesh STA). Mesh STAs are QoS STAs that support mesh services, i.e., they participate in formation and operation of a mesh basic service set (MBSS). A mesh STA implements a subset of the QoS functionality:

- Use of QoS frame format
- EDCA (as a part of MCF)
- Block acknowledgement (optional)
- No acknowledgement (optional)

A mesh BSS does not incorporate the full hybrid coordinator (HC) and BSS QoS functionality. MBSSs do not incorporate the following:

- HCCA
- Traffic specifications (TSPECs)
- Traffic stream (TS) management
- Admission control
- Automatic power save delivery (APSD)
- Direct-link setup (DLS)
- Tunneled direct-link setup (TDLS)

4.3.15.4 IEEE 802.11 components and mesh BSS

Example mesh and infrastructure BSSs are illustrated in Figure 4-8. Only mesh STAs participate in mesh functionalities such as formation of the mesh BSS, path selection, and forwarding. Accordingly, a mesh STA is not a member of an IBSS or an infrastructure BSS. Consequently, mesh STAs do not communicate with nonmesh STAs.

However, instead of existing independently, an MBSS might also access the distribution system (DS). The MBSS interconnects with other BSSs through the DS. Then, mesh STAs can communicate with nonmesh STAs. Therefore, a logical architectural component is introduced in order to integrate the MBSS with the DS—the mesh gate. Data move between an MBSS and the DS via one or more mesh gates. Thus, the mesh gate is the logical point at which MSDUs from an MBSS enter the IEEE 802.11 DS. Once an MBSS contains a mesh gate that connects it to the IEEE 802.11 DS, the MBSS can be integrated with other infrastructure BSSs too, given that their APs connect to the same DS. Several mesh gates are shown in Figure 4-8 connecting different MBSSs to the DS.

When an MBSS accesses the IEEE 802.11 DS through its mesh gate, the MBSS can be integrated with a non-IEEE-802.11 LAN. To integrate the IEEE 802.11 DS to which this MBSS connects, the DS needs to contain a portal. See 4.3.6. Consequently, mesh gate and portal are different entities. The portal integrates the IEEE 802.11 architecture with a non-IEEE-802.11 LAN (e.g., a traditional wired LAN), whereas the mesh gate integrates the MBSS with the IEEE 802.11 DS.

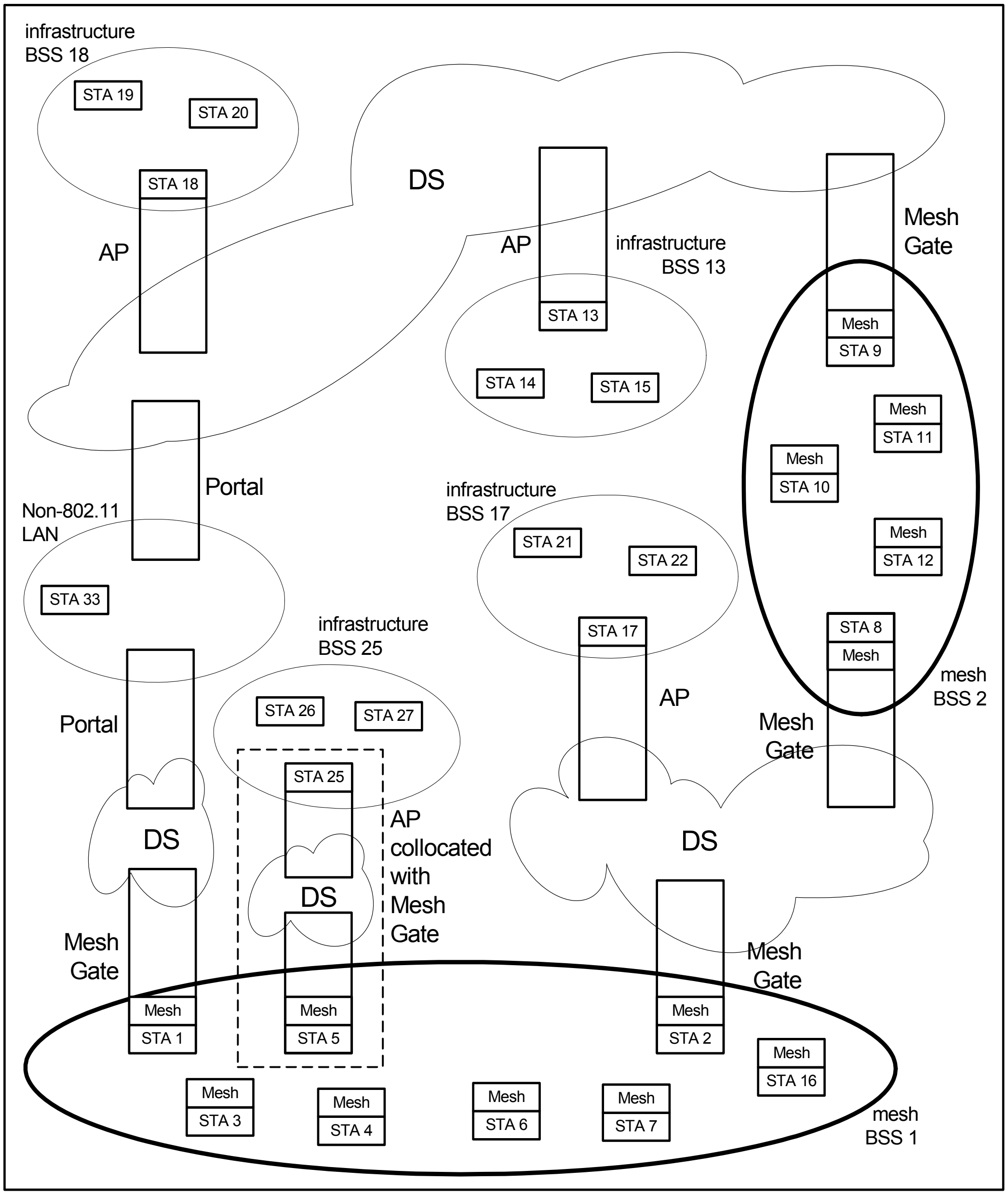


Figure 4-8—Example MBSS containing mesh STAs, mesh gates, APs, and portals

It is possible for one device to offer any combination of the functions of an AP, a portal, and a mesh gate; see 13.11.5. An example device combining the functions of an AP and a mesh gate is shown in Figure 4-9. The implementation of such collocated entities is beyond the scope of this standard. The configuration of a mesh gate that is collocated with an access point allows the utilization of the mesh BSS as a distribution system medium. In this case, two different entities (mesh STA and access point) exist in the collocated device and the mesh BSS is hidden to STAs that associate to the access point. The mesh STA collocation is outlined in 13.11.5, which states that the usage of a distinct MAC address for each collocated STA avoids ambiguities.

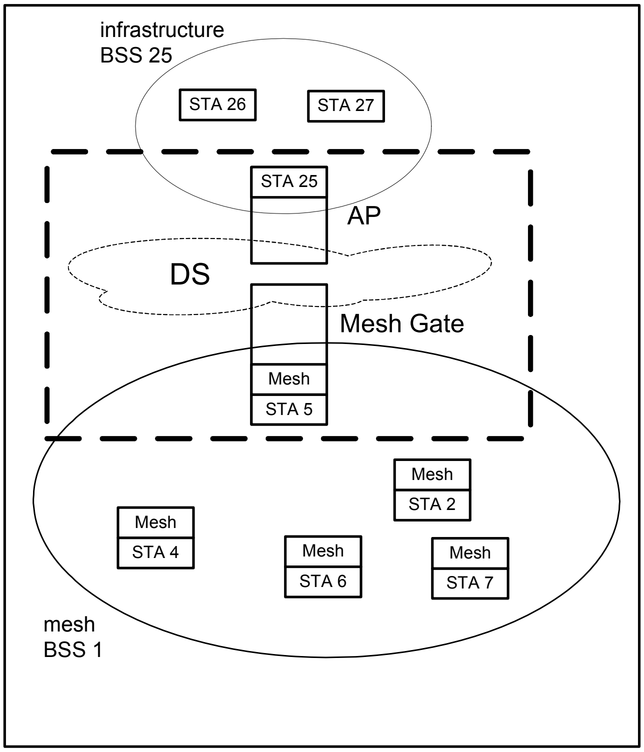


Figure 4-9—Example device consisting of mesh STA and AP STA to connect an MBSS and an infrastructure BSS

4.3.15.5 Introduction to mesh functions

4.3.15.5.1 General

A mesh BSS is formed and operated by the set of services called mesh services. Mesh services are provided by the following major mesh facilities:

- Mesh discovery
- Mesh peering management
- Mesh security
- Mesh beaconing and synchronization
- Mesh coordination function
- Mesh power management
- Mesh channel switching
- Three address, four address, and extended address frame formats
- Mesh path selection and forwarding
- Interworking with external networks
- Intra-mesh congestion control
- Emergency service support in mesh BSS

4.3.15.5.2 Mesh discovery

A mesh STA performs either active scanning or passive scanning to discover an operating mesh BSS. Each mesh STA transmits Beacon frames periodically and responds with Probe Response frames when a Probe Request frame is received, so that neighbor mesh STAs can perform mesh discovery appropriately. The identification of the mesh BSS is given by the Mesh ID element contained in the Beacon and the Probe Response frames. The details for the mesh discovery facility are described in 13.2.

4.3.15.5.3 Mesh peering management (MPM)

Within a mesh BSS, direct communication between neighbor mesh STAs is allowed only when they are peer mesh STAs. After mesh discovery, two neighbor mesh STAs agree to establish a mesh peering to each other, and, after successfully establishing the mesh peering, they become peer mesh STAs. A mesh STA can establish a mesh peering with multiple neighbor mesh STAs. The mesh peering management (MPM) facilitates the mesh peering establishment and closure of the mesh peerings. The details of MPM are described in 13.3.

4.3.15.5.4 Mesh security

In an MBSS, mesh link security protocols are used to authenticate a pair of mesh STAs and to establish session keys between them. Mesh authentication protocols establish a shared, common pairwise master key (PMK), and authenticate a peer mesh STA. The authenticated mesh peering exchange protocol relies on the existence of the PMK between the two mesh STAs to establish an authenticated peering and derive session keys. The details of mesh security are described in 11.3, 13.3.3, 13.5, and 13.6.

4.3.15.5.5 Mesh beaconing and synchronization

In order to assist mesh discovery, mesh power management, and synchronization in a mesh BSS, all mesh STAs periodically transmit Beacon frames. Synchronization in a mesh BSS is maintained by the MBSS's active synchronization method. The default synchronization method is the neighbor offset synchronization method. Mesh beacon collision avoidance (MBCA) mitigates collisions of Beacon frames among hidden nodes. The details of mesh beaconing and synchronization are described in 13.13.

4.3.15.5.6 Mesh coordination function (MCF)

A mesh STA uses the mesh coordination function (MCF) for channel access. MCF consists of EDCA (contention-based channel access defined in 9.20.2) and MCCA (controlled channel access defined in 9.20.3). MCCA is a reservation based channel access method and aims to optimize the efficiency of frame exchanges in a mesh BSS.

4.3.15.5.7 Mesh power management

A mesh STA can manage the activity level of its links per mesh peering. A mesh STA sets the activity level of each of its mesh peerings to either active mode, light sleep mode, or deep sleep mode. The mesh STA performs mesh power mode tracking for each of its neighbor peer mesh STAs, and delivers the frames based on the rules defined in 13.14.

4.3.15.5.8 Mesh channel switching

When a mesh STA switches the operating channel, it uses the channel switch protocol defined in 10.9.8 and 10.10. The channel switch protocol enables the propagation of channel switching messages throughout the mesh BSS, prior to the channel switch execution.

4.3.15.5.9 Frame addressing in an MBSS

Three address, four address, and extended address frame formats enable the distribution of messages over multiple instances of the wireless medium within a mesh BSS and integration to the ESS. Frame format details are described in Clause 8 and 9.32.3.

4.3.15.5.10 Mesh path selection and forwarding

Mesh path selection enables path discovery over multiple instances of the wireless medium within a mesh BSS. The overview of the mesh path selection framework is described in 13.8. The hybrid wireless mesh protocol (HWMP) is defined as the default path selection protocol for the mesh BSS. HWMP provides both proactive path selection and reactive path selection. The details of HWMP are described in 13.10. The path selection protocol uses link metrics in the assessment of a mesh path to the destination. The airtime link metric is the default link metric. It is defined in 13.9.

Once the mesh path of a particular pair of the source mesh STA and the destination mesh STA is found through the mesh path selection function, mesh STAs propagate the data by the forwarding function. The details of the forwarding function are described in 9.32.

As a result of the mesh path selection and forwarding, MSDUs are transmitted among all the mesh STAs in a mesh BSS, even if the mesh STAs are not neighbor STAs of each other. Figure 4-10 depicts the MSDU transfer within a mesh BSS.

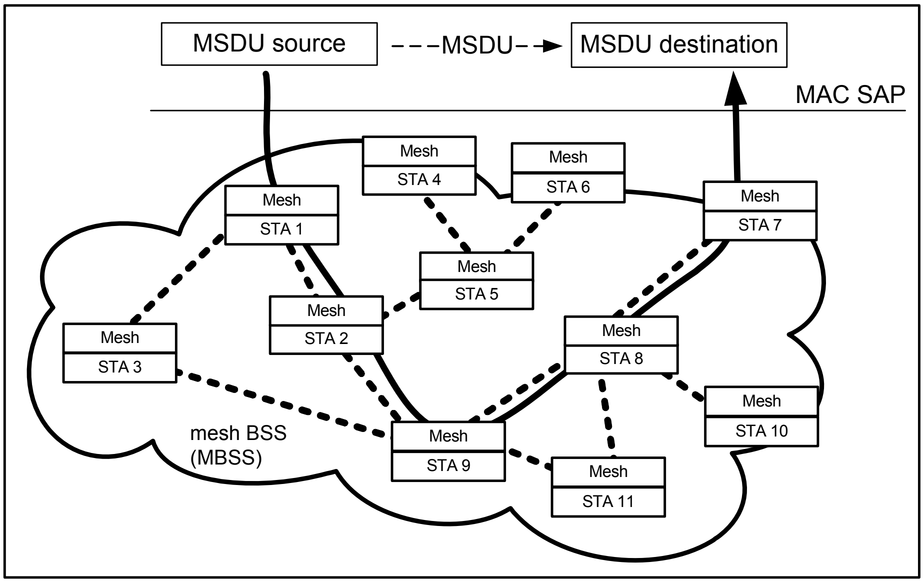


Figure 4-10—MAC data transport over an MBSS

4.3.15.5.11 Interworking with the DS

A mesh BSS might contain one or more mesh gates that connect to one or more distribution systems. A mesh gate can announce its presence in the mesh BSS by sending Gate Announcement frames. Alternatively a mesh gate can announce its presence in the mesh BSS by sending HWMP Path Selection frames with the RANN element or the PREQ element indicating mesh gate announcement, when it is configured as a root mesh STA. Typically a mesh gate announces its presence when it is collocated with a portal or it has access to a portal. Gate announcements allow mesh STAs to select the appropriate mesh gate and build a path towards it. It should be noted that, when multiple mesh gates that have access to the same DS are present in the mesh BSS, proper configuration is necessary.

When a mesh gate has access to IEEE 802 STAs outside the mesh BSS, the mesh gate acts as a proxy for the IEEE 802 STAs outside the MBSS. Such a mesh gate is called a proxy mesh gate. The details of the proxy functionality are described in 13.11.3.

The details of the mesh BSS interworking are described in 13.11.

4.3.15.5.12 Intra-mesh congestion control

Intra-mesh congestion control is used to provide flow control over the multi-hop communication. Intra-mesh congestion control is useful to mitigate wasteful wireless medium utilization caused by buffer overflow at mesh STAs. Intra-mesh congestion control consists of three main mechanisms: local congestion monitoring and congestion detection, congestion control signaling, and local rate control. The details of the intra-mesh congestion control are described in 13.12.

4.3.15.5.13 Emergency service support in mesh BSS

Depending on regulations, emergency services support might be mandated over the mesh network. In this case the Beacon and Probe Response frames inform whether a mesh STA supports emergency services, advertising to other mesh STAs that mesh peering for emergency services is possible. If a mesh STA subsequently requires emergency services, an emergency indication is then set within the Mesh Peering Open frame. Mesh STAs that support emergency services, accept peering from other mesh STAs requiring emergency services, transferring frames to an emergency server (such as a PSAP).

4.4 Logical service interfaces

4.4.1 General

A DS may be created from many different technologies including current IEEE 802 wired LANs. IEEE Std 802.11 does not constrain the DS to be either data link or network layer based. Nor does IEEE Std 802.11 constrain a DS to be either centralized or distributed in nature.

IEEE Std 802.11 explicitly does not specify the details of DS implementations. Instead, IEEE Std 802.11 specifies *services*. The services are associated with different components of the architecture. There are two categories of IEEE 802.11 service—the station service (SS) and the distribution system service (DSS). Both categories of service are used by the IEEE 802.11 MAC sublayer.

The complete set of IEEE 802.11 architectural services are as follows:

a) Authentication
b) Association
c) Deauthentication
d) Disassociation
e) Distribution
f) Integration
g) Data confidentiality
h) Reassociation
i) MSDU delivery
j) DFS

k) TPC
l) Higher layer timer synchronization (QoS facility only)
m) QoS traffic scheduling (QoS facility only)
n) Radio measurement
o) DSE

This set of services is divided into two groups: the SS and the DSS. The SS is part of every STA. The DSS is provided by the DS.

4.4.2 SS

The service provided by STAs is known as the SS.

The SS is present in every IEEE 802.11 STA (including APs, as APs include STA functionality). The SS is specified for use by MAC sublayer entities. All conformant STAs provide SS.

The SS is as follows:

a) Authentication (not used when dot11OCBActivated is true)
b) Deauthentication (not used when dot11OCBActivated is true)
c) Data confidentiality (not used when dot11OCBActivated is true)
d) MSDU delivery
e) DFS
f) TPC
g) Higher layer timer synchronization (QoS facility only)
h) QoS traffic scheduling (QoS facility only)
i) Radio measurement
j) DSE

4.4.3 DSS

The service provided by the DS is known as the DSS.

This service is represented in the IEEE 802.11 architecture by arrows within APs and mesh gates, indicating that the service is used to cross media and possibly address space logical boundaries. An AP and a mesh gate are logical entities, and the functions described may be shared by one or more physical entities.

The services that comprise the DSS are as follows:

a) Association (not mesh facility)
b) Disassociation (not mesh facility)
c) Distribution
d) Integration
e) Reassociation (not mesh facility)
f) QoS traffic scheduling (QoS facility only)
g) DSE
h) Interworking with the DS (mesh facility only)

DSSs are specified for use by MAC sublayer entities.

Figure 4-11 combines the components from previous figures with both types of services to show the complete IEEE 802.11 architecture.

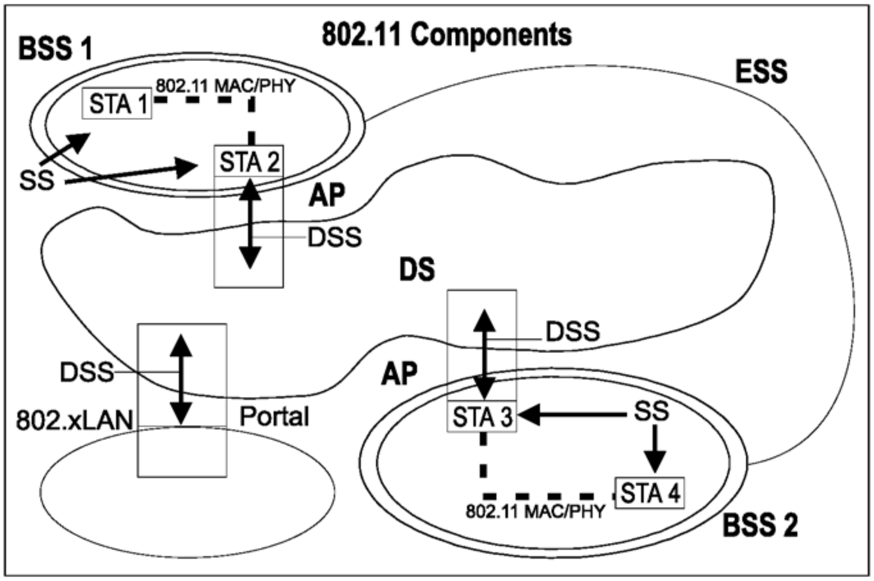


Figure 4-11—Complete IEEE 802.11 architecture

4.5 Overview of the services

4.5.1 General

There are many services specified by IEEE Std 802.11. Six of the services are used to support medium access control (MAC) service data unit (MSDU) delivery between STAs. Three of the services are used to control IEEE 802.11 LAN access and confidentiality. Two of the services are used to provide spectrum management. One of the services provides support for LAN applications with QoS requirements. Another of the services provides support for higher layer timer synchronization. One of the services is used for radio measurement.

This subclause presents the services, an overview of how each service is used, and a description of how each service relates to other services and the IEEE 802.11 architecture. The services are presented in an order designed to help build an understanding of the operation of an IEEE 802.11 ESS network. As a result, the services that comprise the SS and DSS are intermixed in order (rather than being grouped by category).

Each of the services is supported by one or more MAC frame types. Some of the services are supported by MAC management messages and some by MAC data messages. All of the messages gain access to the WM via the IEEE 802.11 MAC sublayer medium access method specified in Clause 9.

The IEEE 802.11 MAC sublayer uses three types of messages—*data*, *management*, and *control* (see Clause 8). The data messages are handled via the MAC data service path.

MAC management messages are used to support the IEEE 802.11 services and are handled via the MAC management service path.

MAC control messages are used to support the delivery of IEEE 802.11 data and management messages.

The examples here assume an ESS network environment. The differences between the ESS and the IBSS network environments are discussed separately in 4.7.

4.5.2 Distribution of messages within a DS

4.5.2.1 Distribution

This is the primary service used by IEEE 802.11 STAs. It is conceptually invoked by every data message to or from an IEEE 802.11 STA operating in an ESS (when the frame is sent via the DS). Distribution is via the DSS.

Refer to the ESS network in Figure 4-11 and consider a data message being sent from STA 1 to STA 4. The message is sent from STA 1 and received by STA 2 (the "input" AP). The AP gives the message to the distribution service of the DS. It is the job of the distribution service to deliver the message within the DS in such a way that it arrives at the appropriate DS destination for the intended recipient. In this example, the message is distributed to STA 3 (the "output" AP) and STA 3 accesses the WM to send the message to STA 4 (the intended destination).

How the message is distributed within the DS is not specified by IEEE Std 802.11. All IEEE Std 802.11 is required to do is to provide the DS with enough information for the DS to be able to determine the "output" point that corresponds to the intended recipient. The necessary information is provided to the DS by the three association related services (association, reassociation, and disassociation).

The previous example was a case in which the AP that invoked the distribution service was different from the AP that received the distributed message. If the message had been intended for a STA that was a member of the same BSS as the sending STA, then the "input" and "output" APs for the message would have been the same.

In either example, the distribution service was logically invoked. Whether the message actually had to traverse the physical DSM or not is a DS implementation matter and is not specified by this standard.

While IEEE Std 802.11 does not specify DS implementations, it does recognize and support the use of the WM as one possible DSM. This is specifically supported by the IEEE 802.11 frame formats. (Refer to Clause 8 for details.) A mesh BSS might form an entire DS or a part of a DS using the WM, as shown in Figure 4-8. Mesh services are used to form a mesh BSS and distribute messages. Clause 13 defines how mesh BSSs are formed and how messages are distributed through a mesh BSS.

4.5.2.2 Integration

If the distribution service determines that the intended recipient of a message is a member of an integrated LAN, the "output" point of the DS would be a portal instead of an AP.

Messages that are distributed to a portal cause the DS to invoke the Integration function (conceptually after the distribution service). The Integration function is responsible for accomplishing whatever is needed to deliver a message from the DSM to the integrated LAN media (including any required media or address space translations). Integration is one of the services in the DSS.

Messages received from an integrated LAN (via a portal) by the DS for an IEEE 802.11 STA invoke the Integration function before the message is distributed by the distribution service.

The details of an Integration function are dependent on a specific DS implementation and are outside the scope of this standard.

4.5.2.3 QoS traffic scheduling

QoS traffic scheduling provides intra-BSS QoS frame transfers under the HCF, using either contention-based or controlled channel access. At each TXOP, a traffic scheduling entity at the STA selects a frame for transmission, from the set of frames at the heads of a plurality of traffic queues, based on requested UP and/or parameter values in the traffic specification (TSPEC) for the requested MSDU. Additional information is available in 9.19.

4.5.3 Services that support the distribution service

4.5.3.1 General

The primary purpose of a MAC sublayer is to transfer MSDUs between MAC sublayer entities. The information required for the distribution service to operate is provided by the association services. Before a data message can be handled by the distribution service, a STA is “associated.”

To understand the concept of association, it is necessary first to understand the concept of mobility.

4.5.3.2 Mobility types

The three transition types of significance to this standard that describe the mobility of STAs within a network are as follows:

a) ***No-transition:*** In this type, two subclasses that are usually indistinguishable are identified:
   1) Static—no motion.
   2) Local movement—movement within the PHY range of the communicating STAs, i.e., movement within a basic service area (BSA).

b) ***BSS-transition:*** This type is defined as a STA movement from one BSS in one ESS to another BSS within the same ESS. A fast BSS transition is a BSS transition that establishes the state necessary for data connectivity before the reassociation rather than after the reassociation.

c) ***ESS-transition:*** This type is defined as STA movement from a BSS in one ESS to a BSS in a different ESS. This case is supported only in the sense that the STA may move. Maintenance of upper-layer connections cannot be guaranteed by IEEE Std 802.11; in fact, disruption of service is likely to occur.

The FT Protocol provides a mechanism for a STA to perform a BSS transition between access points (APs) in a robust security network (RSN) or when quality-of-service (QoS) admission control is enabled in the ESS.

The different association services support the different categories of mobility.

4.5.3.3 Association

To deliver a message within a DS, the distribution service needs to know which AP to access for the given IEEE 802.11 STA. This information is provided to the DS by the concept of association. Association is necessary, but not sufficient, to support BSS-transition mobility. Association is sufficient to support no-transition mobility. Association is one of the services in the DSS.

Before a STA is allowed to send a data message via an AP, it first becomes associated with the AP. The act of becoming associated invokes the association service, which provides the STA to AP mapping to the DS. The DS uses this information to accomplish its message distribution service. How the information provided by the association service is stored and managed within the DS is not specified by this standard.

Within a robust security network (RSN), association is handled differently. In an RSNA, the IEEE 802.1X Port determines when to allow data traffic across an IEEE 802.11 link. A single IEEE 802.1X Port maps to one association, and each association maps to an IEEE 802.1X Port. An IEEE 802.1X Port consists of an IEEE 802.1X Controlled Port and an IEEE 802.1X Uncontrolled Port. The IEEE 802.1X Controlled Port is blocked from passing general data traffic between two STAs until an IEEE 802.1X authentication procedure completes successfully over the IEEE 802.1X Uncontrolled Port. Once the AKM completes successfully, data protection is enabled to prevent unauthorized access, and the IEEE 802.1X Controlled Port unblocks to allow protected data traffic. IEEE 802.1X Supplicants and Authenticators exchange protocol information via the IEEE 802.1X Uncontrolled Port. It is expected that most other protocol exchanges will make use of the IEEE 802.1X Controlled Ports. However, a given protocol might need to bypass the authorization function and make use of the IEEE 802.1X Uncontrolled Port.

NOTE—See IEEE Std 802.1X-2004 for a discussion of Controlled Port and Uncontrolled Port.

At any given instant, a STA is associated with no more than one AP. This allows the DS to determine a unique answer to the question, “Which AP is serving STA X?” Once an association is completed, a STA may make full use of a DS (via the AP) to communicate. Association is always initiated by the mobile STA, not the AP.

An AP may be associated with many STAs at one time.

A STA learns what APs are present and what operational capabilities are available from each of those APs and then invokes the association service to establish an association. For details of how a STA learns about what APs are present, see 10.1.4.

4.5.3.4 Reassociation

Association is sufficient for no-transition message delivery between IEEE 802.11 STAs. Additional functionality is needed to support BSS-transition mobility. The additional required functionality is provided by the reassociation service. Reassociation is one of the services in the DSS.

The reassociation service is invoked to “move” a current association from one AP to another. This keeps the DS informed of the current mapping between AP and STA as the STA moves from BSS to BSS within an ESS. Reassociation also enables changing association attributes of an established association while the STA remains associated with the same AP. Reassociation is always initiated by the mobile STA.

No facilities are provided to move an RSNA during reassociation. Therefore, the old RSNA is deleted, and a new RSNA is constructed.

4.5.3.5 Disassociation

The disassociation service is invoked when an existing association is to be terminated. Disassociation is one of the services in the DSS.

In an ESS, this tells the DS to void existing association information. Attempts to send messages via the DS to a disassociated STA will be unsuccessful.

The disassociation service may be invoked by either party to an association (non-AP STA or AP). Disassociation is a notification, not a request. Disassociation cannot be refused by the receiving STA except when management frame protection is negotiated and the message integrity check fails.

APs may disassociate STAs to enable the AP to be removed from a network for service or for other reasons.

STAs attempt to disassociate when they leave a network. However, the MAC protocol does not depend on STAs invoking the disassociation service. (MAC management is designed to accommodate loss of communication with an associated STA.)

4.5.4 Access control and data confidentiality services

4.5.4.1 General

Two services are required for IEEE Std 802.11 to provide functionality equivalent to that which is inherent to wired LANs. The design of wired LANs assumes the physical attributes of wire. In particular, wired LAN design assumes the physically closed and controlled nature of wired media. The physically open medium nature of an IEEE 802.11 LAN violates those assumptions.

In a WLAN that does not support RSNA, two services, authentication and data confidentiality, are defined. IEEE 802.11 authentication is used instead of the wired media physical connection. WEP encryption was defined to provide the data confidentiality aspects of closed wired media.

An RSNA uses the IEEE 802.1X authentication service along with enhanced data cryptographic encapsulation mechanisms, such as TKIP and CCMP to provide access control. The IEEE 802.11 station management entity (SME) provides key management via an exchange of IEEE 802.1X EAPOL-Key frames. Data confidentiality and data integrity are provided by RSN key management together with the enhanced data cryptographic encapsulation mechanisms.

4.5.4.2 Authentication

IEEE 802.11 authentication operates at the link level between IEEE 802.11 STAs. IEEE Std 802.11 does not provide either end-to-end (message origin to message destination) or user-to-user authentication.

IEEE Std 802.11 attempts to control LAN access via the authentication service. IEEE 802.11 authentication is an SS. This service may be used by all STAs to establish their identity to STAs with which they communicate, in both ESS and IBSS networks. If a mutually acceptable level of authentication has not been established between two STAs, an association is not established.

IEEE Std 802.11 defines four 802.11 authentication methods: Open System authentication, Shared Key authentication, FT authentication, and simultaneous authentication of equals (SAE). Open System authentication admits any STA to the DS. Shared Key authentication relies on WEP to demonstrate knowledge of a WEP encryption key. FT authentication relies on keys derived during the initial mobility domain association to authenticate the stations as defined in Clause 12. SAE authentication uses finite field cryptography to prove knowledge of a shared password. The IEEE 802.11 authentication mechanism also allows definition of new authentication methods.

An RSNA might support SAE authentication. An RSNA also supports authentication based on IEEE Std 802.1X-2004, or preshared keys (PSKs) after Open System authentication. IEEE 802.1X authentication utilizes the EAP to authenticate STAs and the AS with one another. This standard does not specify an EAP method that is mandatory to implement. See 11.5.5 for a description of the IEEE 802.1X authentication and PSK usage within an IEEE 802.11 IBSS.

In an RSNA, IEEE 802.1X Supplicants and Authenticators exchange protocol information via the IEEE 802.1X Uncontrolled Port. The IEEE 802.1X Controlled Port is blocked from passing general data traffic between two STAs until an IEEE 802.1X authentication procedure completes successfully over the IEEE 802.1X Uncontrolled Port.

SAE authentication or Open System 802.11 authentication is used in an RSN for infrastructure BSS. SAE authentication, Open System 802.11 authentication, or no 802.11 authentication is used in an RSN for IBSS. SAE authentication is used in an MBSS. An RSNA disallows the use of Shared Key 802.11 authentication.

A STA may be authenticated with many other STAs at any given instant.

Because the IEEE 802.1X authentication process could be time-consuming (depending on the authentication protocol in use), the authentication service can be invoked independently of the association service.

This type of preauthentication is typically done by a STA while it is already associated with an AP (with which it previously authenticated). IEEE Std 802.11 does not require that STAs preauthenticate with APs. However, authentication is required before an association establishment is complete.

If the authentication is left until reassociation time, this might impact the speed with which a STA reassociates between APs, limiting BSS-transition mobility performance. The use of preauthentication takes the authentication service overhead out of the time-critical reassociation process.

SAE authentication is performed prior to association and a STA can take advantage of the fact that it can be IEEE 802.11 authenticated to many APs simultaneously by completing the SAE protocol with any number of APs while still being associated to another AP. RSNA security can be established after association using the resulting shared key.

4.5.4.3 Deauthentication

The deauthentication service is invoked when an existing Open System, Shared Key, or SAE authentication is to be terminated. Deauthentication is an SS.

When the deauthentication service is terminating SAE authentication any PTKSA, GTKSA, mesh TKSA, or mesh GTKSA related to this SAE authentication is destroyed. If PMK caching is not enabled, deauthentication also destroys any PMKSA created as a result of this successful SAE authentication.

In an ESS, because authentication is a prerequisite for association, the act of deauthentication causes the STA to be disassociated. The deauthentication service may be invoked by either authenticated party (non-AP STA or AP). Deauthentication is not a request; it is a notification. The association at the transmitting STA is terminated when the STA sends a deauthentication notice to an associated STA. Deauthentication, and if associated, disassociation cannot be refused by the receiving STA except when management frame protection is negotiated and the message integrity check fails.

In an RSN ESS, Open System 802.11 authentication is required. In an RSN ESS, deauthentication results in termination of any association for the deauthenticated STA. It also results in the IEEE 802.1X Controlled Port for that STA being disabled and deletes the pairwise transient key security association (PTKSA). The deauthentication notification is provided to IEEE Std 802.1X-2004 via the MAC layer.

In an RSNA, deauthentication also destroys any related pairwise transient key security association (PTKSA), group temporal key security association (GTKSA), station-to-station link (STSL) master key security association (SMKSA), STSL transient key security association (STKSA), and integrity group temporal key security association (IGTKSA) that exist in the STA and closes the associated IEEE 802.1X Controlled Port. If pairwise master key (PMK) caching is not enabled, deauthentication also destroys the pairwise master key security association (PMKSA) from which the deleted PTKSA was derived.

In an RSN IBSS, Open System authentication is optional, but a STA is required to recognize Deauthentication frames. Deauthentication results in the IEEE 802.1X Controlled Port for that STA being disabled and deletes the PTKSA.

4.5.4.4 Data confidentiality

In a wired LAN, only those STAs physically connected to the wire can send or receive LAN traffic. With a wireless shared medium, there is no physical connection, and all STAs and certain other RF devices in or near the LAN might be able to send, receive, and/or interfere with the LAN traffic. An IEEE 802.11-compliant STA can receive like-PHY IEEE 802.11 traffic that is within range and can transmit to any other IEEE 802.11 STA within range. Thus, the connection of a single wireless link (without data confidentiality) to an existing wired LAN may seriously degrade the security level of the wired LAN.

To bring the security of the WLAN up to the level implicit in wired LAN design, IEEE Std 802.11 provides the ability to protect the contents of messages. This functionality is provided by the data confidentiality service. Data confidentiality is an SS.

IEEE Std 802.11 provides several cryptographic algorithms to protect data traffic, including: WEP, TKIP, and CCMP. WEP and TKIP are based on the ARC4[19] algorithm, and CCMP is based on the advanced encryption standard (AES). A means is provided for STAs to select the algorithm(s) to be used for a given association.

IEEE Std 802.11 provides one security protocol, CCMP, for protection of individually addressed robust management frames. This standard does not provide data confidentiality for group addressed robust management frames.

IEEE Std 802.11 provides one security protocol, CCMP, for protection of individually addressed and group addressed data frames between mesh STAs.

The default data confidentiality state for all IEEE 802.11 STAs is “in the clear,” i.e., without protection. If the data confidentiality service is not invoked, all frames are sent unprotected. If this policy is unacceptable to the sender, it does not send data frames; and if the policy is unacceptable to the receiver, it discards any received data frames. Unprotected data frames and unprotected robust management frames received at a STA configured for mandatory data confidentiality, as well as protected data frames and protected robust management frames using a key not available at the receiving STA, are discarded without an indication to LLC (or without indication to distribution services in the case of “To DS” frames received at an AP). These frames are acknowledged on the WM [if received without frame check sequence (FCS) error] to avoid wasting WM bandwidth on retries of frames that are being discarded.

4.5.4.5 Key management

The enhanced data confidentiality, data authentication, and replay protection mechanisms require fresh cryptographic keys and corresponding security associations. The procedures defined in this standard provide fresh keys by means of protocols called the 4-Way Handshake, FT 4-Way Handshake, FT Protocol, FT Resource Request Protocol, and Group Key Handshake.

4.5.4.6 Data origin authenticity

The data origin authenticity mechanism defines a means by which a STA that receives a data or protected Robust Management frame can determine which STA transmitted the MAC protocol data unit (MPDU). This feature is required in an RSNA to prevent one STA from masquerading as a different STA.

[19] Details of the ARC4 algorithm are available from RSA Security, Inc. Contact RSA Security, 174 Middlesex Turnpike, Bedford, MA 01730 (http://www.rsasecurity.com/), for algorithm details and the uniform ARC4 license terms that RSA offers to anyone wishing to use ARC4 for the purpose of implementing the IEEE 802.11 WEP option. If necessary, contact the IEEE Standards Department Intellectual Property Rights Administrator for details on how to communicate with RSA.

Data origin authenticity is only applicable to individually addressed data frames, and individually addressed robust management frames. The protocols do not guarantee data origin authenticity for group addressed frames, as this cannot be accomplished using symmetric keys and public key methods are too computationally expensive.

4.5.4.7 Replay detection

The replay detection mechanism defines a means by which a STA that receives a data or protected Robust Management frame from another STA can detect whether the received frame is an unauthorized retransmission. This replay protection mechanism is provided for data frames for STAs that use enhanced data cryptographic encapsulation mechanisms. The replay protection mechanism is also provided for robust management frames for STAs that use CCMP and Broadcast/Multicast Integrity Protocol (BIP).

4.5.4.8 Fast BSS transition

The FT mechanism defines a means for a STA to set up security and QoS parameters prior to reassociation to a new AP. This mechanism allows time-consuming operations to be removed from the time-critical reassociation process.

4.5.4.9 Robust management frame protection

Robust management frames are a set of management frames that can be protected by the management frame protection service. The robust management frames are Disassociation, Deauthentication, and robust Action frames. Action frames specified with “No” in the “Robust” column of Table 8-38 are not robust management frames and are not protected.

Management frame protection protocols in an infrastructure BSS or IBSS apply to robust management frames after RSNA PTK establishment for protection of individually addressed frames is completed and after delivery of the IGTK to protect group addressed frames. Robust management frame protection is implemented by CCMP, BIP, and the SA Query procedure.

Management frame protection protocols in an MBSS apply to individually addressed frames after establishment of the RSNA MTK, and to group addressed frames indicated as "Group Addressed Privacy" in Table 8-38. Robust management frame protection is implemented by CCMP.

4.5.5 Spectrum management services

4.5.5.1 General

Two services are required to satisfy requirements in some regulatory domains (see Annex D and Annex E) for operation in the 5 GHz band. These services are called transmit power control (TPC) and dynamic frequency selection (DFS).

4.5.5.2 TPC

Radio regulations may require radio local area networks (RLANs) operating in the 5 GHz band to use transmitter power control, involving specification of a regulatory maximum transmit power and a mitigation requirement for each allowed channel, to reduce interference with satellite services. The TPC service is used to satisfy this regulatory requirement.

The TPC service provides for the following:

- Association of STAs with an AP in a BSS based on the STAs' power capability.
- Specification of regulatory and local maximum transmit power levels for the current channel.
- Selection of a transmit power for each transmission in a channel within constraints imposed by regulatory requirements.
- Adaptation of transmit power based on a range of information, including path loss and link margin estimates.

4.5.5.3 DFS

Radio regulations might require RLANs operating in the 5 GHz band to implement a mechanism to avoid co-channel operation with radar systems and to provide uniform utilization of available channels. The DFS service is used to satisfy these regulatory requirements.

The DFS service provides for the following:

- Association of STAs with an AP in a BSS based on the STAs' supported channels.
- Quieting the current channel so it can be tested for the presence of radar with less interference from other STAs.
- Testing channels for radar before using a channel and while operating in a channel.
- Discontinuing operations after detecting radar in the current channel to avoid interference with radar.
- Detecting radar in the current and other channels based on regulatory requirements.
- Requesting and reporting of measurements in the current and other channels.
- Selecting and advertising a new channel to assist the migration of a BSS after radar is detected.

4.5.6 Traffic differentiation and QoS support

IEEE Std 802.11 uses a shared medium and provides differentiated control of access to the medium to handle data transfers with QoS requirements. The QoS facility (per MSDU traffic category and TSPEC negotiation) allows an IEEE 802.11 LAN to become part of a larger network providing end-to-end QoS delivery or to function as an independent network providing transport on a per-link basis with specified QoS commitments. The specifications regarding the integration and operability of the QoS facility in IEEE 802.11 specification with any other end-to-end QoS delivery mechanism like Resource Reservation Protocol (RSVP) are beyond the scope of this standard.

4.5.7 Support for higher layer timer synchronization

Some applications, e.g., the transport and rendering of audio or video streams, require synchronized timers shared among different STAs. Greater accuracy (in terms of jitter bounds) or finer timer granularity than that provided by a BSS timing synchronization function (TSF) may be an additional requirement. In support of such applications, this standard defines a MAC service that enables layers above the MAC to accurately synchronize application-dependent timers shared among STAs. The service is usable by more than one application at a time.

Although the timer synchronization methods and accuracy requirements are application-dependent and are beyond the scope of this standard, they rely on an indication from each MAC that is provided essentially simultaneously, via group addressed transmissions, to the STAs. The MAC accomplishes this by indicating the occurrence of the end of the last symbol of particular data frames; the data frames of interest are identified by their MAC header Address 1 field when it contains a group address previously registered with the MAC. The last symbol is observed[20] on the air by STAs within a BSS while the delay between the observation and the delivery of the indication is known within a MAC by design (and communicated to the application by implementation-dependent means). The common reference point in time provided by the end of last symbol indication is the essential building block upon which a variety of application-dependent timer synchronization methods may be based.

4.5.8 Radio Measurement service

The Radio Measurement service provides the following:

- The ability to request and report radio measurements in supported channels.
- The ability to perform radio measurements in supported channels.
- An interface for upper layer applications to retrieve radio measurements using MLME primitives and/or MIB access.
- Information about neighbor APs.

4.5.9 Interworking with external networks

The interworking service allows non-AP STAs to access services provided by an external network according to the subscription or other characteristics of that external network. An IEEE 802.11 non-AP STA may have a subscription relationship with an external network, e.g., with an SSPN.

An overview of the interworking functions addressed in this standard is provided below:

- Network discovery and selection
    - Discovery of suitable networks through the advertisement of access network type, roaming consortium and venue information, via management frames
    - Selection of a suitable IEEE 802.11 infrastructure using advertisement services (e.g., Access Network Query Protocol (ANQP) or an IEEE 802.21 Information Server) in the BSS or in an external network reachable via the BSS.
    - Selection of an SSPN or external network with its corresponding IEEE 802.11 infrastructure
- Emergency services
    - Emergency Call and Network Alert support at the link level
- QoS Map distribution
- SSPN interface service between the AP and the SSPN

The generic advertisement service (GAS), described in 4.11, provides both support for a STA’s network selection and a conduit for communication by a non-AP STA with other information resources in a network before joining the wireless LAN.

The interworking service supports emergency services by providing methods for users to access emergency services via the IEEE 802.11 infrastructure, advertising that emergency services are supported (see 10.24.6) and identifying that a traffic stream is used for emergency services.

[20] The synchronization indication (observed time) within the BSS varies slightly due to propagation.

The interworking service provides QoS mapping for SSPNs and other external networks. Since each SSPN or other external network may have its own layer-3 end-to-end packet marking practice (e.g., differentiated services code point (DSCP) usage conventions), a means to remap the layer-3 service levels to a common over-the-air service level is necessary. The QoS Map service provides STAs a mapping of network-layer QoS packet marking to over-the-air QoS frame marking (i.e., user priority).

The SSPN Interface service supports service provisioning and transfer of user permissions from the SSPN to the AP. The method and protocol by which these permissions are transferred from the SSPN are outside the scope of this standard.

4.6 Multiple logical address spaces

Just as the IEEE 802.11 architecture allows for the possibility that the WM, DSM, and an integrated wired LAN may all be different physical media, it also allows for the possibility that each of these components may be operating within different address spaces. IEEE Std 802.11 only uses and specifies the use of the WM address space.

Each IEEE 802.11 PHY operates in a single medium—the WM. The IEEE 802.11 MAC operates in a single address space. MAC addresses are used on the WM in the IEEE 802.11 architecture. Therefore, it is unnecessary for the standard to explicitly specify that its addresses are “WM addresses.” This is assumed throughout this standard.

IEEE Std 802.11 has chosen to use the IEEE 802 48-bit address space (see 8.2.4.3.2). Thus IEEE 802.11 addresses are compatible with the address space used by the IEEE 802 LAN family.

The IEEE 802.11 choice of address space implies that for many instantiations of the IEEE 802.11 architecture, the wired LAN MAC address space and the IEEE 802.11 MAC address space may be the same. In those situations where a DS that uses MAC level IEEE 802 addressing is appropriate, all three of the logical address spaces used within a system could be identical. While this is a common case, it is not the only combination allowed by the architecture. The IEEE 802.11 architecture allows for all three logical address spaces to be distinct.

A multiple address space example is one in which the DS implementation uses network layer addressing. In this case, the WM address space and the DS address space would be different.

Note that IEEE 802.11 STAs within a single ESS share the same address space, fulfilling the transparency requirement from the definition of the DS. The DSS uses this same address space, even in the case where the DSM uses a different address space.

The ability of the architecture to handle multiple logical media and address spaces is key to the ability of IEEE Std 802.11 to be independent of the DS implementation and to interface cleanly with network layer mobility approaches. The implementation of the DS is unspecified and is beyond the scope of this standard.

4.7 Differences between ESS and IBSS LANs

In 4.3.2 the concept of the IBSS LAN was introduced. In an IBSS network, a STA communicates directly with one or more other STAs.

Consider the full IEEE 802.11 architecture as shown in Figure 4-12.

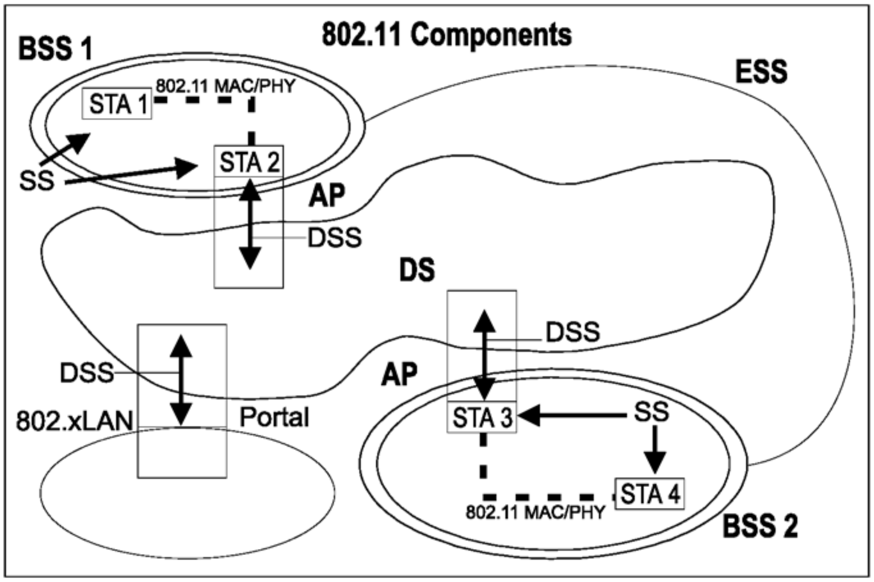


Figure 4-12—IEEE 802.11 architecture (again)

An IBSS consists of STAs that are directly connected. Thus there is (by definition) only one BSS. Further, because there is no physical DS, there is no portal, integrated wired LAN, or DSS. The logical picture reduces to Figure 4-13.

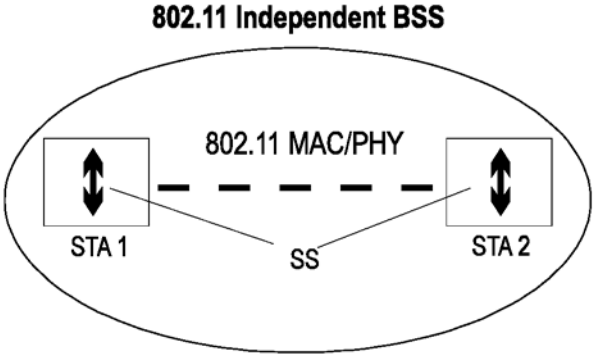


Figure 4-13—Logical architecture of an IBSS

Only the minimum two STAs are shown in Figure 4-13. An IBSS may have an arbitrary number of members. In an IBSS only Class 1 and Class 2 frames are allowed because there is no DS in an IBSS.

The services that apply to an IBSS are the SSs. A QoS IBSS supports operation under the HCF using TXOPs gained through the EDCA mechanism. The parameters that control differentiation of the delivery of MSDUs with different priority using EDCA are fixed. A QoS IBSS has no HC and does not support polled TXOP operation and setting up of TSPEC.

In an IBSS each STA enforces its own security policy. In an ESS, an AP can enforce a uniform security policy across all STAs.

4.8 Differences between ESS and MBSS LANs

In 4.3.15, the concept of the MBSS LAN was introduced. It was noted that using the multi-hop capability it appears as if all mesh STAs are directly connected at the MAC layer even if the STAs are not within range of each other. This is different from an IBSS network, where STAs cannot communicate if they are not within range of each other.

Unlike the IBSS, an MBSS might have access to the DS. An MBSS connects through one or more mesh gates to the DS. Since in an MBSS it appears as if all mesh STAs are directly connected at the MAC layer, the MBSS can be used as a DSM. APs, portals, and mesh gates might use the MBSS as a DSM to provide the DSS. Thus, different infrastructure BSSs can unite over the MBSS to form an ESS for example.

An AP identifies the infrastructure BSS that it forms. This is different from the MBSS where no such central entity exists. Whereas infrastructure BSSs need the ESS and thus the DS to unite, the MBSS network appears the same to an LLC layer without the need for access to a DS. However, if an MBSS has one or more mesh gates providing access to the DS, the MBSS might exist in disjointed areas and yet form a single network.

4.9 Reference model

4.9.1 General

This standard presents the architectural view, emphasizing the separation of the system into two major parts: the MAC of the data link layer (DLL) and the PHY. These layers are intended to correspond closely to the lowest layers of the ISO/IEC basic reference model of Open Systems Interconnection (OSI) (ISO/IEC 7498-1: 1994). The layers and sublayers described in this standard are shown in Figure 4-14.

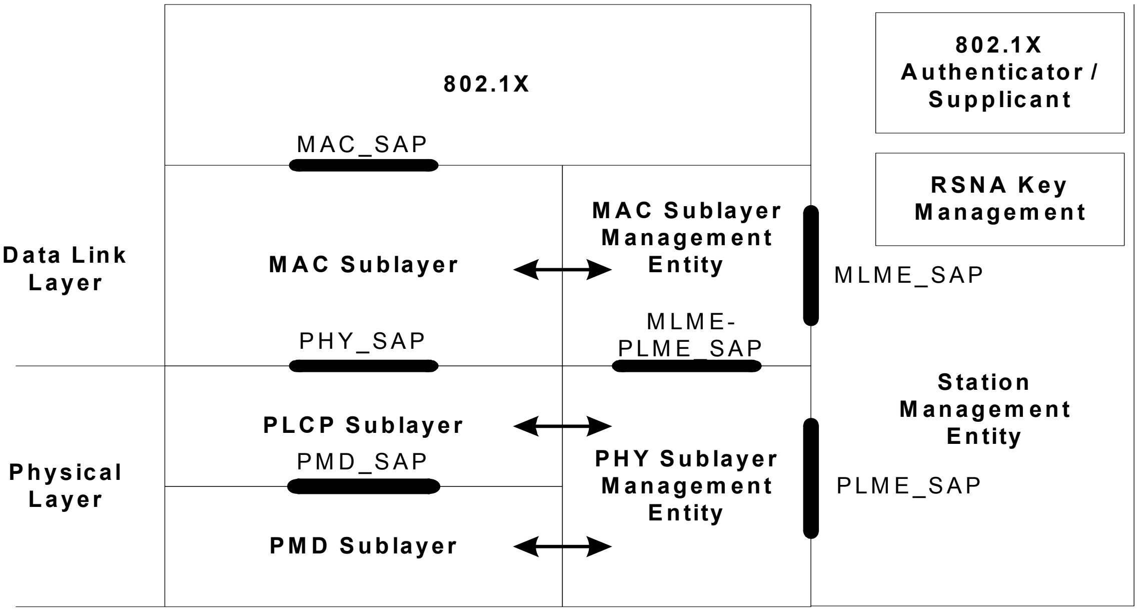


Figure 4-14—Portion of the ISO/IEC basic reference model covered in this standard

There is an interface between the IEEE 802.1X Supplicant/Authenticator and the SME not shown in Figure 4-14. This interface is described in IEEE Std 802.1X-2004.

4.9.2 Interworking reference model

The MAC state generic convergence function (MSGCF) provides services to higher layer protocols based on MAC state machines and interactions between the layers.

Interworking functions may require correlating information from multiple management entities. It is the function of the MSGCF to correlate information for higher layer entities. The MSGCF observes the interactions between the MLME and SME, and between the PLME and SME. After correlation of lower-layer MLME and PLME events, the MSGCF may synthesize indications to higher layer entities.

Figure 4-15 shows an entity, the MSGCF, defined in 6.4, that has access to all management information through exposure to the MAC and PHY Sublayer Management Entities, and provides management information to higher level entities, such as Mobility Managers, supporting heterogeneous medium mobility.

An example of how the MSGCF interfaces to these higher layer entities, is provided by the media-independent handover (MIH) interface, as defined in IEEE 802.21-2008.

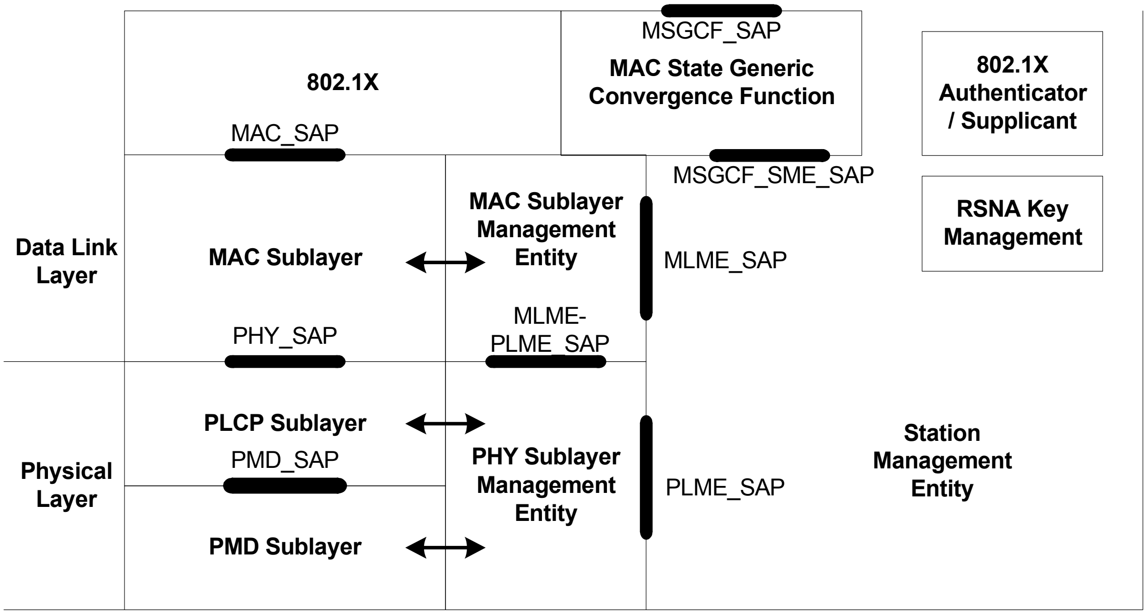


Figure 4-15—Interworking reference model

The MSGCF is designed to provide the status of the connection of a non-AP STA to a set of BSSs comprising a single ESS. Figure 4-16 illustrates the concept of an ESS Link. This reflects the state of a connection to an ESS independent of any particular access point. In Figure 4-16, STA3 is associated with either AP1 or AP2. The state of the ESS Link is up when STA3 is associated with any of the APs comprising an ESS.

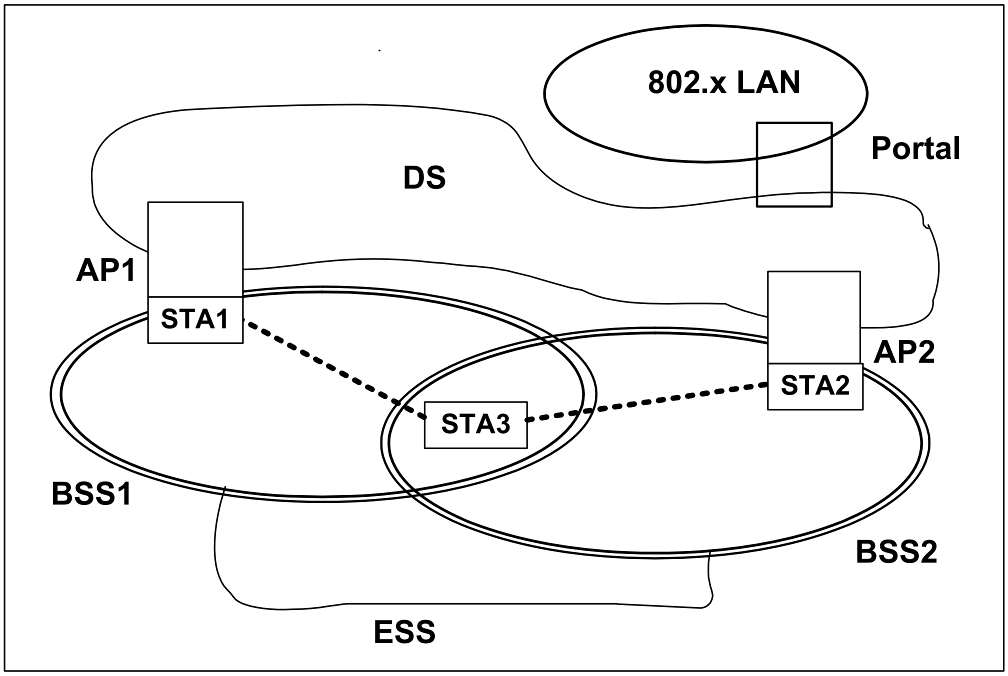


Figure 4-16—ESS link illustration

4.10 IEEE Std 802.11 and IEEE Std 802.1X-2004

4.10.1 General

An RSNA relies on IEEE Std 802.1X-2004 to provide authentication services and uses the IEEE 802.11 key management scheme defined in 11.6. The IEEE 802.1X access control mechanisms apply to the association between a STA and an AP and to the relationship between the IBSS STA and STA peer. The AP's SME performs the Authenticator and, optionally, the Supplicant and AS roles. In an ESS, a non-AP STA's SME performs the Supplicant role. In an IBSS the SME takes on both the Supplicant and Authenticator roles and may take on the AS role.

4.10.2 IEEE 802.11 usage of IEEE Std 802.1X-2004

IEEE Std 802.11 depends upon IEEE Std 802.1X-2004 to control the flow of MAC service data units (MSDUs) between the DS and STAs by use of the IEEE 802.1X Controlled/Uncontrolled Port model. IEEE 802.1X authentication frames are transmitted in IEEE 802.11 data frames and passed via the IEEE 802.1X Uncontrolled Port. The IEEE 802.1X Controlled Port is blocked from passing general data traffic between two STAs until an IEEE 802.1X authentication procedure completes successfully over the IEEE 802.1X Uncontrolled Port. It is the responsibility of both the Supplicant and the Authenticator to implement port blocking. Each association between a pair of STAs creates a unique pair of IEEE 802.1X Ports, and authentication takes place relative to those ports alone.

IEEE Std 802.11 depends upon IEEE Std 802.1X-2004 and the 4-Way Handshake, FT 4-Way Handshake, FT Protocol, FT Resource Request Protocol, and Group Key Handshake, described in Clause 11 and Clause 12, to establish and change cryptographic keys. Keys are established after authentication has completed. Keys may change for a variety of reasons, including expiration of an IEEE 802.1X authentication timer, key compromise, danger of compromise, or policy.

4.10.3 Infrastructure functional model overview

4.10.3.1 General

This subclause summarizes the system setup and operation of an RSN, in three cases: when a password or PSK is used during IEEE 802.11 authentication, when an IEEE 802.1X AS is used after Open System authentication, and when a PSK is used after Open System authentication. For an ESS, the AP includes an Authenticator, and each associated STA includes a Supplicant.

4.10.3.2 AKM operations with AS

The following AKM operations are carried out when an IEEE 802.1X AS is used:

a) Prior to any use of IEEE Std 802.1X-2004, IEEE Std 802.11 assumes that the Authenticator and AS have established a secure channel. The security of the channel between the Authenticator and the AS is outside the scope of this standard.

   Authentication credentials are distributed to the Supplicant and AS prior to association.

b) A STA discovers the AP’s security policy through passively monitoring Beacon frames or through active probing (shown in Figure 4-17). If IEEE 802.1X authentication is used, the EAP authentication process starts when the Authenticator sends the EAP-Request (shown in Figure 4-18) or the Supplicant sends the EAPOL-Start message. EAP authentication frames pass between the Supplicant and AS via the Authenticator and Supplicant’s Uncontrolled Ports. This is shown in Figure 4-18.

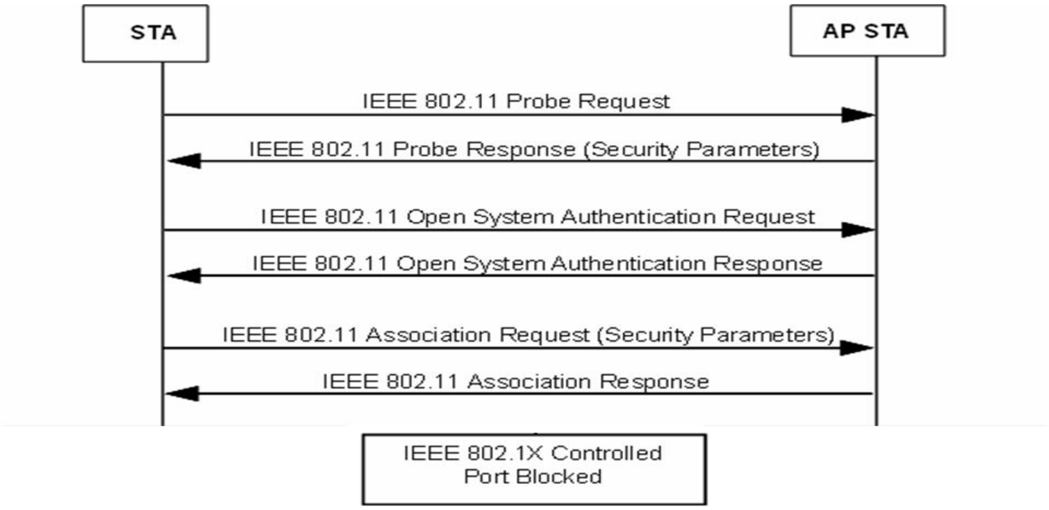

STA
AP STA
IEEE 802.11 Probe Request
IEEE 802.11 Probe Response (Security Parameters)
IEEE 802.11 Open System Authentication Request
IEEE 802.11 Open System Authentication Response
IEEE 802.11 Association Request (Security Parameters)
IEEE 802.11 Association Response
IEEE 802.1X Controlled Port Blocked

Figure 4-17—Establishing the IEEE 802.11 association

c) The Supplicant and AS authenticate each other and generate a PMK. The PMK is sent from the AS to the Authenticator over the secure channel. See Figure 4-18.

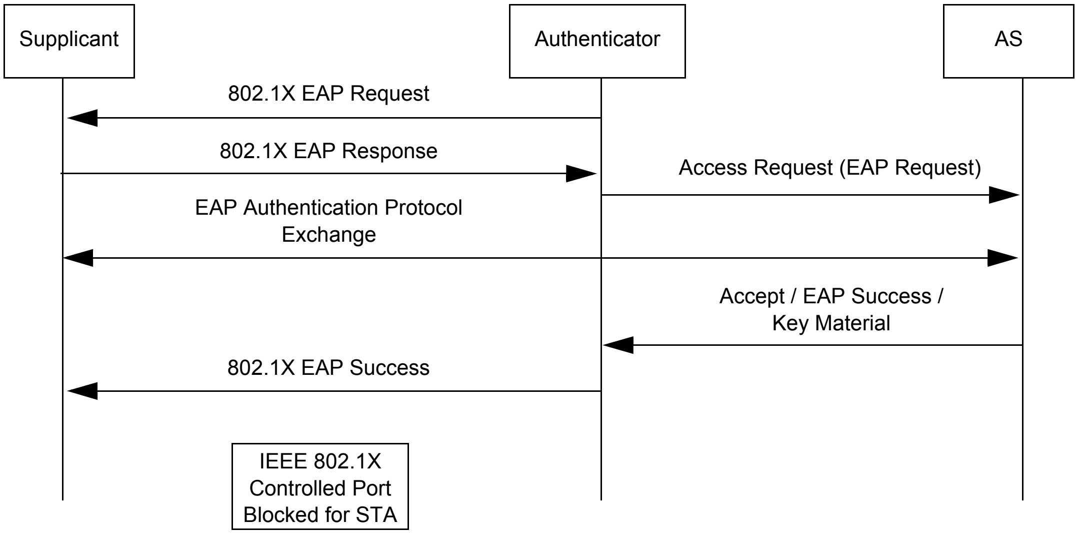

Supplicant
Authenticator
AS
802.1X EAP Request
802.1X EAP Response
Access Request (EAP Request)
EAP Authentication Protocol Exchange
Accept / EAP Success / Key Material
802.1X EAP Success
IEEE 802.1X Controlled Port Blocked for STA

Figure 4-18—IEEE 802.1X EAP authentication

A 4-Way Handshake or FT 4-Way Handshake utilizing EAPOL-Key frames is initiated by the Authenticator to do the following:

- Confirm that a live peer holds the PMK.
- Confirm that the PMK is current.
- In the case of fast BSS transition, derive PMK-R0s and PMK-R1s.
- Derive a fresh pairwise transient key (PTK) from the PMK or, in the case of fast BSS transition, from the PMK-R1.
- Install the pairwise encryption and integrity keys into IEEE Std 802.11.
- Transport the group temporal key (GTK) and GTK sequence number from Authenticator to Supplicant and install the GTK and GTK sequence number in the STA and, if not already installed, in the AP.
- If management frame protection is negotiated, transport the IGTK and the IGTK packet number (IPN) from the Authenticator to the Supplicant and install these values in the STA and, if not already installed, in the AP.
- Verify that the RSN capabilities negotiated are valid as defined in 8.4.2.27.4.
- Confirm the cipher suite selection.

Installing the PTK, and where applicable the GTK keys, causes the MAC to encrypt and decrypt all subsequent MSDUs irrespective of their path through the controlled or uncontrolled ports.

Upon successful completion of the 4-Way Handshake, the Authenticator and Supplicant have authenticated each other; and the IEEE 802.1X Controlled Ports are unblocked to permit general data traffic. See Figure 4-19.

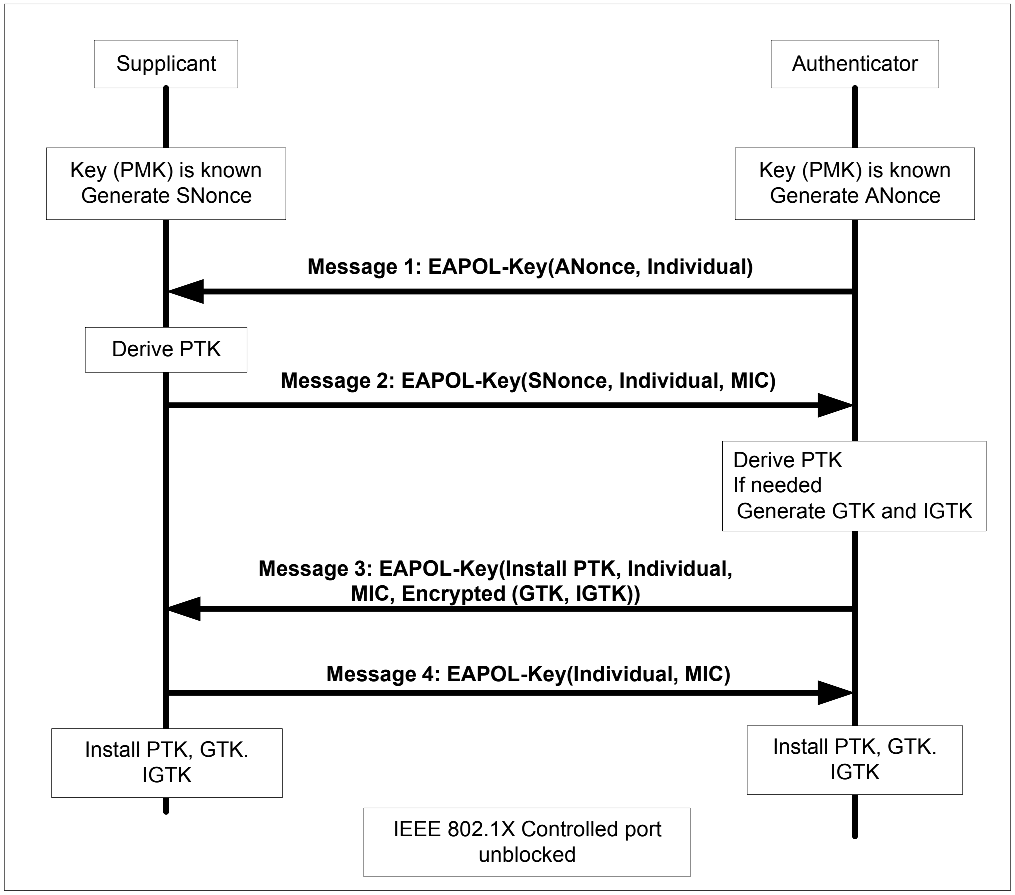

Supplicant
Authenticator
Key (PMK) is known
Generate SNonce
Key (PMK) is known
Generate ANonce
Message 1: EAPOL-Key(ANonce, Individual)
Derive PTK
Message 2: EAPOL-Key(SNonce, Individual, MIC)
Derive PTK
If needed
Generate GTK and IGTK
Message 3: EAPOL-Key(Install PTK, Individual, MIC, Encrypted (GTK, IGTK))
Message 4: EAPOL-Key(Individual, MIC)
Install PTK, GTK. IGTK
Install PTK, GTK. IGTK
IEEE 802.1X Controlled port unblocked

Figure 4-19—Establishing pairwise and group keys

If the Authenticator later changes the GTK, it sends the new GTK and GTK sequence number to the Supplicant using the Group Key Handshake to allow the Supplicant to continue to receive group addressed messages and, optionally, to transmit and receive individually addressed frames. EAPOL-Key frames are used to carry out this exchange. See Figure 4-20.

When management frame protection is negotiated, the Authenticator also uses the Group Key Handshake with all associated STAs to change the IGTK. The Authenticator encrypts the GTK and IGTK values in the EAPOL-Key frame as described in 11.6.

4.10.3.3 AKM Operations with a Password or PSK

The following AKM operations are carried out when authentication is accomplished using a Password or PSK:

— A STA discovers the AP's security policy through passively monitoring the Beacon frames or through active probing. After discovery the STA performs SAE authentication using IEEE 802.11 Authentication frames with the AP (see Figure 4-21).

— Upon the successful conclusion of SAE, both the STA and AP generate a PMK. The STA then associates to an AP and negotiate security policy. The AKM confirmed in the Association Request and Response is the AKM of SAE or Fast BSS Transition.

— The PMK generated by SAE is used in a 4-Way Handshake using EAPOL-Key frames, just as with IEEE 802.1X authentication when an AS is present. See Figure 4-19.

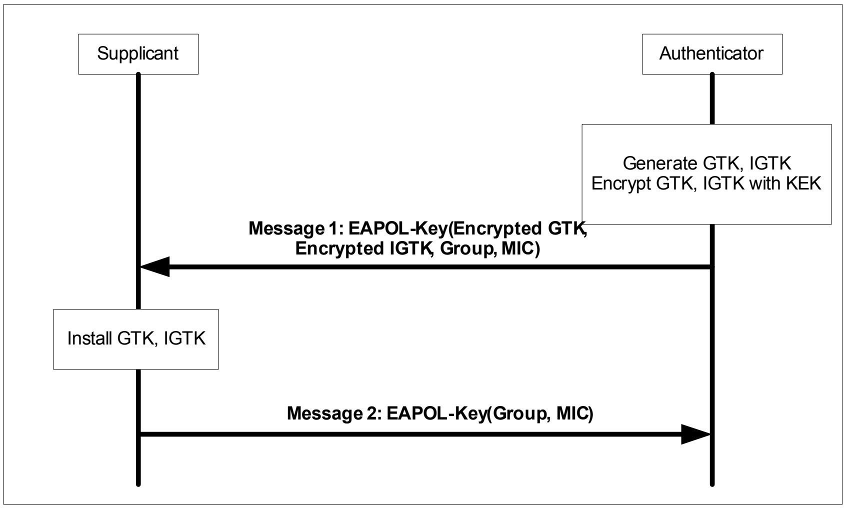

Supplicant
Authenticator
Generate GTK, IGTK
Encrypt GTK, IGTK with KEK
Message 1: EAPOL-Key(Encrypted GTK, Encrypted IGTK, Group, MIC)
Install GTK, IGTK
Message 2: EAPOL-Key(Group, MIC)

Figure 4-20—Delivery of subsequent group keys

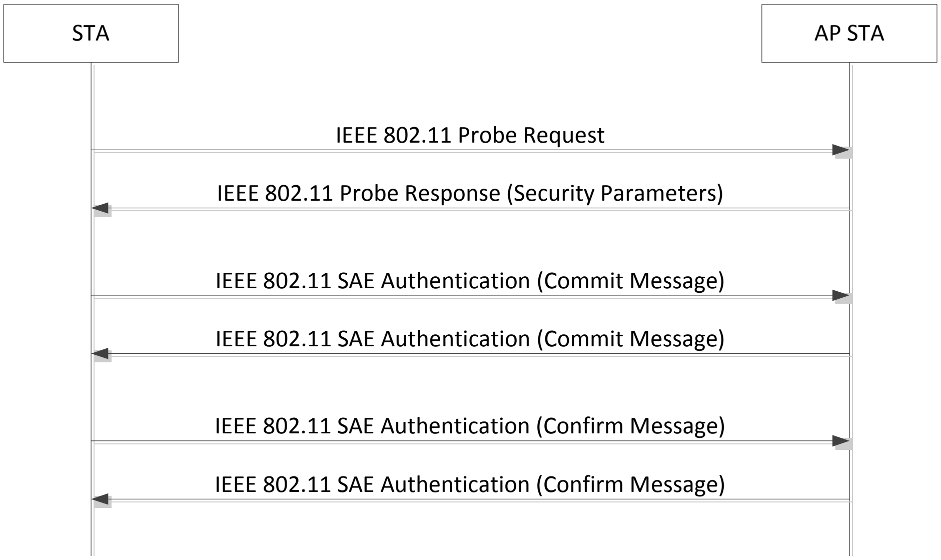

STA
AP STA
IEEE 802.11 Probe Request
IEEE 802.11 Probe Response (Security Parameters)
IEEE 802.11 SAE Authentication (Commit Message)
IEEE 802.11 SAE Authentication (Commit Message)
IEEE 802.11 SAE Authentication (Confirm Message)
IEEE 802.11 SAE Authentication (Confirm Message)

Figure 4-21—Example using SAE Authentication

- The GTK and GTK sequence number are sent from the Authenticator to the Supplicant just as in the AS case. See Figure 4-19 and Figure 4-20.

4.10.3.4 Alternate operations with PSK

The following AKM operations represent an alternate operation of using a PSK. This operation has security vulnerabilities when used with a low-entropy key and is recommended to be used only after taking that into account. When this operation is carried out, the PMK is a PSK:

- A STA discovers the AP’s security policy through passively monitoring Beacon frames or through active probing (shown in Figure 4-17). A STA associates with an AP and negotiates a security policy. The PMK is the PSK.

- The 4-Way Handshake using EAPOL-Key frames is used, just as with IEEE 802.1X authentication, when an AS is present. See Figure 4-19.
- The GTK and GTK sequence number are sent from the Authenticator to the Supplicant just as in the AS case. See Figure 4-19 and Figure 4-20.
- If management frame protection is negotiated, the IGTK and IGTK packet number are sent from the Authenticator to the Supplicant just as in the AS case. See Figure 4-19 and Figure 4-20.

4.10.3.5 Disassociation

Disassociation initiated by either STA in an RSNA causes the deletion of the PTKSA at both ends and the deletion of the GTKSA in a non-AP STA. The controlled and uncontrolled ports created for this association are also deleted.

4.10.4 IBSS functional model description

4.10.4.1 General

This subclause summarizes the system setup and operation of an RSNA in an IBSS. An IBSS RSNA is specified in 11.5.10.

4.10.4.2 Key usage

In an IBSS the individually addressed data frames between two STAs are protected with a pairwise key. The key is part of the PTK, which is derived during a 4-Way Handshake. In an IBSS the 4-Way Handshake may follow IEEE 802.11 authentication of one STA to another. Such authentication may be used by the peer to cause deletion of the PTKSA and Block the Controlled Port thus resetting any previous handshake.

In an IBSS group addressed data frames are protected by a key, e.g., named B1, that is generated by the STA transmitting the group addressed frame. To allow other STAs to decrypt group addressed frames, B1 is sent to all the other STAs in the IBSS. B1 is sent in an EAPOL-Key frame, encrypted under the EAPOL-Key encryption key (KEK) portion of the PTK, and protected from modification by the EAPOL-Key confirmation key (KCK) portion of the PTK.

In an IBSS the SME responds to Deauthentication frames from a STA by deleting the PTKSA associated with that STA.

4.10.4.3 Sample IBSS 4-Way Handshakes

In this example (see Figure 4-22), there are three STAs: S1, S2, S3. The group addressed frames sent by S1 are protected by B1; similarly B2 for S2, and B3 for S3.

For STAs S2 and S3 to decrypt group addressed frames from S1, B1 is sent to S2 and S3. This is done using the 4-Way Handshake initially and using the Group Key Handshake for GTK updates.

The 4-Way Handshake from S1 to S2 allows S1 to send group addressed frames to S2, but does not allow S2 to send group addressed frames to S1 because S2 has a different transmit GTK. Therefore, S2 needs to initiate a 4-Way Handshake to S1 to allow S1 to decrypt S2's group addressed frames. Similarly, S2 also needs to initiate a 4-Way Handshake to S3 to enable S3 to receive group addressed messages from S2.

In a similar manner S3 needs to complete the 4-Way Handshake with S1 and S2 to deliver B3 to S1 and S2.

In this example, there are six 4-Way Handshakes. In general, N Supplicants require $N(N–1)$ 4-Way Handshakes.

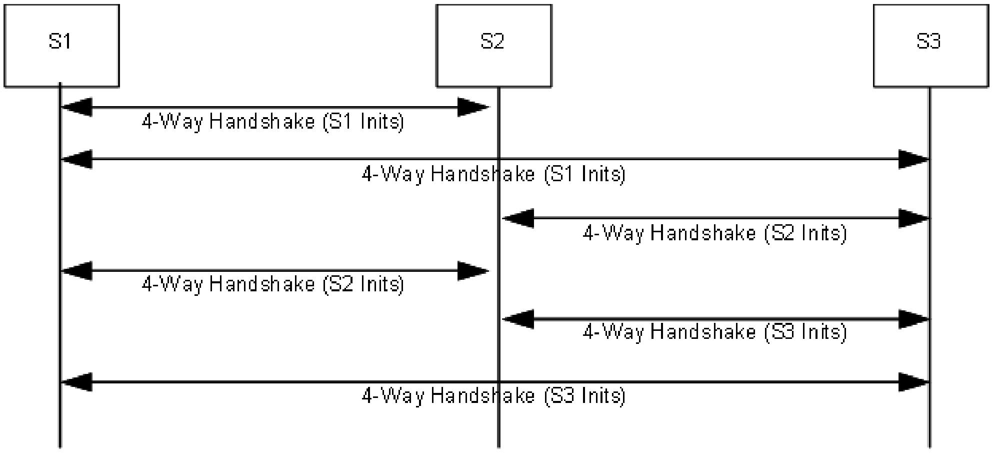

S1
S2
S3
4-Way Handshake (S1 Inits)
4-Way Handshake (S1 Inits)
4-Way Handshake (S2 Inits)
4-Way Handshake (S2 Inits)
4-Way Handshake (S3 Inits)
4-Way Handshake (S3 Inits)

Figure 4-22—Sample 4-Way Handshakes in an IBSS

NOTE—In principle the KCK and KEK from a single 4-Way Handshake can be used for the Group Key Handshake in both directions, but using two 4-Way Handshakes means the Authenticator key state machine does not need to be different between IBSS and ESS.

The Group Key Handshake can be used to send the GTKs to the correct STAs. The 4-Way Handshake is used to derive the pairwise key and to send the initial GTK. Because in an IBSS there are two 4-Way Handshakes between any two Supplicants and Authenticators, the pairwise key used between any two STAs is from the 4-Way Handshake initiated by the STA Authenticator with the higher MAC address (see 11.6.1 for the notion of address comparison). The KCK and KEK used for a Group Key Handshake are the KCK and KEK derived by the 4-Way Handshake initiated by the same Authenticator that is initiating the Group Key Handshake.

In an IBSS a secure link exists between two STAs when both 4-Way Handshakes have completed successfully. The Supplicant and Authenticator 4-Way Handshake state machines interact so the IEEE 802.1X variable portValid is not set to 1 until both 4-Way Handshakes complete.

If a fourth STA comes within range and its SME decides to initiate a security association with the three peers, its Authenticator initiates 4-Way Handshakes with each of the other three Supplicants. Similarly, the original three STA Authenticators in the IBSS need to initiate 4-Way Handshakes to the fourth STA Supplicant. A STA learns that a peer STA is RSNA-enabled and the peer’s security policy (e.g., whether the Authentication and Key Management Protocol (AKMP) is SAE, PSK, or IEEE 802.1X authentication) from the Beacon or Probe Response frame. The initiation might start for a number of reasons:

a) The fourth STA receives a Beacon or Probe Response frame from a MAC address with which it has not completed a 4-Way Handshake.

b) An SME receives a MLME-PROTECTEDFRAMEDROPPED.indication primitive from a MAC address with which it has not completed a 4-Way Handshake. This could be a group addressed data frame transmitted by any of the STAs. In order to set up a security association to the peer STA, an SME that does not know the security policy of the peer can send a Probe Request frame to the peer STA to find its security policy before setting up a security association to the peer STA.

c) An SME receives Message 1 of the 4-Way Handshake sent to a STA because the initiator received a broadcast data frame, Beacon frame, or Probe Response frame from that STA. In order to set up a security association to the peer STA, a STA that received a 4-Way Handshake but does not know the security policy of the peer can send a Probe Request frame to the peer STA to find its security policy before setting up a security association to the peer STA.

4.10.4.4 IBSS IEEE 802.1X example

When IEEE 802.1X authentication is used, each STA needs to include an IEEE 802.1X Authenticator and AS. A STA learns that a peer STA is RSNA-enabled and the peer's security policy (e.g., whether the AKMP is PSK or IEEE 802.1X authentication) from the Beacon or Probe Response frame.

Each Supplicant sends an EAPOL-Start message to every other STA to which it intends to authenticate, and each STA's Authenticator responds with the identity of the credential it intends to use.

The EAPOL-Start and EAP-Request/Identity messages are initiated when a protected data frame (indicated via a MLME-PROTECTEDFRAMEDROPPED.indication primitive), an IEEE 802.1X message, Beacon frame, or Probe Response frame is received from a MAC address with which the STA has not completed IEEE 802.1X authentication. In order to set up a security association to the peer STA, an SME that does not know the security policy of the peer can send a Probe Request frame to the peer STA to find its security policy before setting up a security association to the peer STA.

Although Figure 4-23 shows the two IEEE 802.1X exchanges serialized, they may occur interleaved.

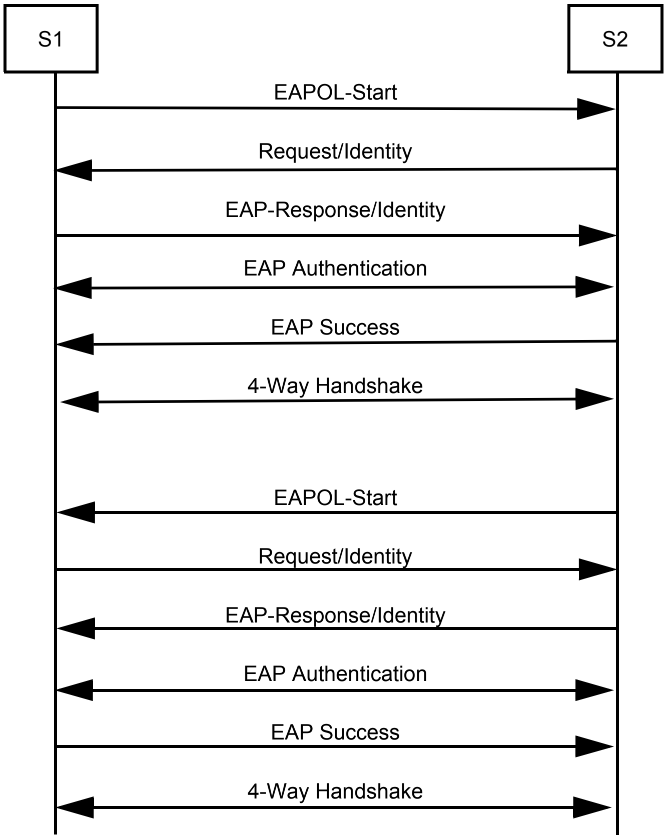


Figure 4-23—Example using IEEE 802.1X authentication

4.10.5 Authenticator-to-AS protocol

The Authenticator-to-AS authentication definition is out of the scope of this standard, but, to provide security assurances, the protocol needs to support the following functions:

a) Mutual authentication between the Authenticator and AS

b) A channel for the Supplicant/AS authentication

c) The ability to pass the generated key from the AS to the Authenticator in a manner that provides authentication of the key source, preserves integrity of the key transfer, and preserves data confidentiality of the key from all other parties

Suitable protocols include, but are not limited to, remote authentication dial-in user service (RADIUS) (IETF RFC 2865-2000 [B31]) and Diameter (IETF RFC 3588-2003 [B36]).

4.10.6 PMKSA caching

The Authenticator and Supplicant may cache PMKSAs, which include the IEEE 802.1X state. A PMKSA can be deleted from the cache for any reason and at any time.

The STA may supply a list of PMK or PSK key identifiers in the (Re)Association Request frame. Each key identifier names a PMKSA; the PMKSA may contain a single PMK. The Authenticator specifies the selected PMK or PSK key identifier in Message 1 of the 4-Way Handshake. The selection of the key identifiers to be included within the (Re)Association Request frame and Message 1 of the 4-Way Handshake is out of the scope of this standard.

4.10.7 Protection of group addressed robust management frames

When management frame protection is negotiated, all group addressed robust management frames are encapsulated using the procedures defined in 10.13. This service provides integrity protection of group addressed robust management frames using BIP.

4.11 Generic advertisement service (GAS)

GAS provides functionality that enables STAs to discover the availability of information related to desired network services, e.g., information about services such as provided in an IBSS, local access services, available subscription service providers (SSPs) and/or SSPNs or other external networks. GAS uses a generic container to advertise network services' information over an IEEE 802.11 network. Public Action frames are used to transport this information.

While the specification of network services information is outside the scope of this standard, in an infrastructure BSS there is a need for STAs to query for information on network services provided by SSPNs or other external networks beyond an AP, before they associate with the wireless LAN. The exchange of information may also be performed after associating to the BSS.

In an IBSS GAS functionality enables a STA to access the availability and information related to desired services provided by other STAs in the IBSS.

There are a number of reasons why providing information to a STA in a preassociated state is beneficial:

— It supports more informed decision making about an IEEE 802.11 infrastructure with which to associate. This is generally more efficient than requiring a non-AP STA to associate with an AP before discovering the information and then deciding whether to stay associated.

— It is possible for the non-AP STA to query multiple networks in parallel.

— The non-AP STA can discover information about APs that are not part of the same administrative group as the AP with which it is associated, supporting the selection of an AP belonging to a different IEEE 802.11 infrastructure that has an appropriate SSP roaming agreement in place.

5. MAC service definition

5.1 Overview of MAC services

5.1.1 Data service

5.1.1.1 General

This service provides peer LLC entities with the ability to exchange MSDUs. To support this service, the local MAC uses the underlying PHY-level services to transport an MSDU to a peer MAC entity, where it is delivered to the peer LLC. Such asynchronous MSDU transport is performed on a connectionless basis. By default, MSDU transport is on a best-effort basis. However, the QoS facility uses a traffic identifier (TID) to specify differentiated services on a per-MSDU basis. The QoS facility also permits more synchronous behavior to be supported on a connection-oriented basis using TSPECs. There are no guarantees that the submitted MSDU will be delivered successfully. Group addressed transport is part of the data service provided by the MAC. Due to the characteristics of the WM, group addressed MSDUs may experience a lower QoS, compared to that of individually addressed MSDUs. All STAs support the data service, but only QoS STAs in a QoS BSS differentiate their MSDU delivery according to the designated traffic category or traffic stream (TS) of individual MSDUs.

Because operation of certain functions of the MAC may cause reordering of some MSDUs, as discussed in more detail below, in non-QoS STAs, there are two service classes within the data service. By selecting the desired service class, each LLC entity initiating the transfer of MSDUs is able to control whether MAC entities are or are not allowed to reorder those MSDUs.

There are two service classes available in a QoS STA: QoSAck and QoSNoAck. The service classes are used to signal if the MSDU is to be transmitted with or without using the MAC-level acknowledgment.

In QoS STAs either associated in a BSS or having membership in an IBSS, the MAC uses a set of rules that tends to cause higher UP MSDUs in a BSS to be sent before lower UP MSDUs in the BSS. The MAC sublayer entities determine the UPs for MSDUs based on the TID values provided with those MSDUs. If a TSPEC has been provided for a TS, via the MAC sublayer management entity, the MAC attempts to deliver MSDUs belonging to that TS in accordance with the QoS parameter values contained in the TSPEC. In a BSS with some STAs supporting the QoS facility and others not supporting the QoS facility, in delivering an MSDU to a non-QoS STA, the QoS STA uses the access category (AC) corresponding to the UP of the MSDU.

5.1.1.2 Determination of UP

The QoS facility supports eight priority values, referred to as *UPs*. The values a UP may take are the integer values from 0 to 7 and are identical to the IEEE 802.1D priority tags. An MSDU with a particular UP is said to belong to a traffic category (TC) with that UP. The UP is provided with each MSDU at the medium access control service access point (MAC_SAP) either directly, in the UP parameter, or indirectly, in a TSPEC designated by the UP parameter.

5.1.1.3 Determination of UP of received frames at the AP sent by other STAs in the BSS

The received individually addressed frames at the AP may be as follows:

a) Non-QoS subtypes, in which case the AP shall assign to them a priority of Contention, if they are received during the contention period (CP), or ContentionFree, if they are received during the contention-free period (CFP).
b) QoS subtypes, in which case the AP shall infer the UP value from the TID in the QoS Control field directly for TID values between 0 and 7. For TID values between 8 and 15 the AP shall extract the

UP value in the UP subfield of the TS Info field in the associated TSPEC or from the UP field in the associated TCLAS (traffic classification) element, as applicable.

QoS APs deliver the UP with the received MSDUs to the DS.

5.1.1.4 Interpretation of priority parameter in MAC service primitives

The value of the priority parameter in the MAC service primitives (see 5.2) may be a noninteger value of either Contention or ContentionFree or may be any integer value in the range 0 to 15.

When the priority parameter has an integer value, it is used in the TID subfields that appear in certain frames that are used to deliver and to control the delivery of QoS data across the WM.

Priority parameter and TID subfield values 0 to 7 are interpreted as UPs for the MSDUs. Outgoing MSDUs with UP values 0 to 7 are handled by MAC entities at STAs in accordance with the UP.

Priority parameter and TID subfield values 8 to 15 specify TIDs that are also TS identifiers (TSIDs) and select the TSPEC for the TS designated by the TID. Outgoing MSDUs with priority parameter values 8 to 15 are handled by MAC entities at STAs in accordance with the UP value determined from the UP subfield as well as other parameter values in the selected TSPEC. When an MSDU arrives with a priority value between 8 and 15 and for which there is no TSPEC defined, then the MSDU shall be sent with priority parameter set to 0.

The noninteger values of the priority parameter are allowed at all non-QoS STAs. The use of priority value of ContentionFree is deprecated at QoS STAs. The integer values of the priority parameter (i.e., TID) are supported only at QoS STAs that are either associated in an infrastructure QoS BSS or members of a QoS IBSS. A range of 0 to 15 is supported by QoS STAs associated in a QoS BSS; whereas a range of 0 to 7 is supported by QoS STAs that are members of a QoS IBSS. If a QoS STA is associated in a non-QoS BSS, the STA is functioning as a non-QoS STA, so the priority value is always Contention or ContentionFree.

At QoS STAs associated in a QoS BSS, MSDUs with a priority of Contention are considered equivalent to MSDUs with TID 0, and those with a priority of ContentionFree are delivered using the contention-free delivery if a point coordinator (PC) is present in the AP. If a PC is not present, MSDUs with a priority of ContentionFree shall be delivered using an UP of 0. At STAs associated in a non-QoS BSS, all MSDUs with an integer priority are considered equivalent to MSDUs with a priority of Contention.

If a STA is associated in a QoS BSS, the MSDUs it receives in QoS data frames are reported with the TID value contained in the MAC header of that frame. The MSDUs such a STA receives in non-QoS data frames are reported to LLC with a priority of Contention, if they are received during the CP, or ContentionFree, if they are received during the CFP.

5.1.1.5 Interpretation of service class parameter in MAC service primitives in a STA

In QoS STAs, the value of the service class parameter in the MAC service primitive (see 5.2) may be a noninteger value of QoSAck or QoSNoAck.

When an MSDU is received from the MAC_SAP and the recipient STA is a QoS STA with the service class set to

- QoSAck, the MSDU is transmitted using a QoS data frame with the Ack Policy subfield in the QoS Control field set to either Normal Ack (normal acknowledgement) or Block Ack.
- QoSNoAck, the MSDU is transmitted using a QoS data frame with the Ack Policy subfield in the QoS Control field set to No Ack (no acknowledgement). If the sender STA is contained within an AP and the frame has a group DA, then the MSDU is buffered for transmission and is also sent to the DS.

When an MSDU is received from the MAC_SAP and the recipient STA is not a QoS STA, the MSDU is transmitted using a non-QoS data frame.

When a QoS data frame is received from another STA, the service class parameter in the MA-UNITDATA.indication primitive is set to

- QoSAck, if the frame is a QoS data frame with the Ack Policy subfield in the QoS Control field equal to either Normal Ack or Block Ack.
- QoSNoAck, if the frame is a QoS data frame with the Ack Policy subfield in the QoS Control field equal to No Ack. This service class is also used where the DA parameter is a group address.

When a non-QoS data frame is received from a STA, the service class parameter in the MA-UNITDATA.indication primitive is set to

- QoSAck, if the frame is an individually addressed frame and is acknowledged by the STA.
- QoSNoAck, if the frame is a group addressed frame and is not acknowledged by the STA.

Note that the group addressed frames sent by a non-QoS STA are not acknowledged regardless of the value of the service class parameter in the MA-UNITDATA.indication primitive.

5.1.2 Security services

Security services in IEEE Std 802.11 are provided by the authentication service and the CCMP and BIP mechanisms. The scope of the security services provided is limited to station-to-station data and robust management frame transmissions. When CCMP is used, the data confidentiality service is provided for data frames and individually addressed robust management frames. For the purposes of this standard, CCMP is viewed as a logical service located within the MAC sublayer as shown in the reference model, Figure 4-14 (in 4.9). Actual implementations of CCMP are transparent to the LLC and other layers above the MAC sublayer.

The security services provided by CCMP in IEEE Std 802.11 are as follows:

a) Data Confidentiality;
b) Authentication; and
c) Access control in conjunction with layer management.

BIP provides message integrity and access control for group addressed robust management frames.

During the authentication exchange, both parties exchange authentication information as described in Clause 11 and Clause 12.

The MAC sublayer security services provided by CCMP and BIP rely on information from nonlayer-2 management or system entities. Management entities communicate information to CCMP and BIP through a set of MAC sublayer management entity (MLME) interfaces and MIB attributes; in particular, the decision tree for CCMP and BIP defined in 11.8 is driven by MIB attributes.

The use of WEP for confidentiality, authentication, or access control is deprecated. The WEP algorithm is unsuitable for the purposes of this standard.

The use of TKIP is deprecated. The TKIP algorithm is unsuitable for the purposes of this standard.

A STA that has associated with management frame protection enabled shall not use pairwise cipher suite selectors WEP-40, WEP-104, TKIP, or “Use Group cipher suite.”

A mesh STA with dot11MeshSecurityActivated equal to true shall not use the pairwise cipher suite selectors WEP-40, WEP-104, or TKIP.

5.1.3 MSDU ordering

The services provided by the MAC sublayer permit, and may in certain cases require, the reordering of MSDUs.

In a non-QoS STA, the MAC does not intentionally reorder MSDUs except as may be necessary to improve the likelihood of successful delivery based on the current operational ("power management," FMS, DMS) mode of the designated recipient STA(s). The sole effect of this reordering (if any), for the set of MSDUs received at the MAC service interface of any single STA, is a change in the delivery order of group addressed MSDUs, relative to individually addressed MSDUs, originating from a single source STA address. If a higher layer protocol using the data service cannot tolerate this possible reordering, the optional StrictlyOrdered service class should be used. MSDUs transferred between any pair of STAs using the StrictlyOrdered service class are not subject to the relative reordering that is possible when the ReorderableGroupAddressed service class is used. However, the desire to receive MSDUs sent using the StrictlyOrdered service class at a STA precludes simultaneous use of the MAC power management facilities at that STA.

In QoS STAs operating in a BSS, there are two service classes, designated as QoSAck and QoSNoAck (see 5.1.1.5 for more information). The MSDUs are reordered, not only to improve the likelihood of successful delivery based on the current operational mode of the designated recipient STA(s), but also to honor the priority parameters, specified in the MA-UNITDATA.request primitive, of the individual MSDUs. The effects of this reordering (if any), for the set of MSDUs received at the MAC service interface of any single STA, are:

a) A change in the delivery order of group addressed MSDUs, relative to individually addressed MSDUs,
b) The reordering of MSDUs with different TID values, originating from a single source STA address, and
c) The reordering of group addressed MSDUs with the same TID but different service classes.

There shall be no reordering of individually addressed MSDUs with the same TID value and addressed to the same destination.

STAs operating in a non-QoS BSS shall follow the reordering rules as defined for a non-QoS STA.

In order for the MAC to operate properly, the DS needs to meet the requirements of ISO/IEC 15802-1:1995.

Operational restrictions that provide the appropriate ordering of MSDUs are specified in 9.8.

5.1.4 MSDU format

This standard is part of the IEEE 802 family of LAN standards, and as such all MSDUs are LLC PDUs as defined in ISO/IEC 8802-2: 1998. In order to achieve interoperability, implementers are recommended to apply the procedures described in ISO/IEC Technical Report 11802-5:1997(E) (previously known as IEEE Std 802.1H-1997 [B21]), along with a selective translation table (STT) that handles a few specific network protocols, with specific attention to the operations required when passing MSDUs to or from LANs or operating system components that use the Ethernet frame format. Note that such translations may be required in a STA.

5.1.5 MAC data service architecture

The MAC data plane architecture (i.e., processes that involve transport of all or part of an MSDU) is shown in Figure 5-1. During transmission, an MSDU goes through some or all of the following processes: MSDU rate limiting, aggregate MSDU (A-MSDU) aggregation, frame delivery deferral during power save mode, sequence number assignment, fragmentation, encryption, integrity protection, frame formatting, and aggregate MAC protocol data unit (A-MPDU) aggregation. IEEE Std 802.1X-2004 may block the MSDU at the Controlled Port. At some point, the data frames that contain all or part of the MSDU are queued per AC/TS.

During reception, a received data frame goes through processes of possible A-MPDU deaggregation, MPDU header and cyclic redundancy code (CRC) validation, duplicate removal, possible reordering if the Block Ack mechanism is used, decryption, defragmentation, integrity checking, and replay detection. After replay detection (or defragmentation if security is not used), possible A-MSDU deaggregation, and possible MSDU rate limiting, one or more MSDUs are, delivered to the MAC_SAP or to the DS. The IEEE 802.1X Controlled/Uncontrolled Ports discard any received MSDU if the Controlled Port is not enabled and if the MSDU does not represent an IEEE 802.1X frame. Frame order enforcement provided by the enhanced data cryptographic encapsulation mechanisms occurs after decryption, but prior to MSDU defragmentation; therefore, defragmentation fails if MPDUs arrive out of order.

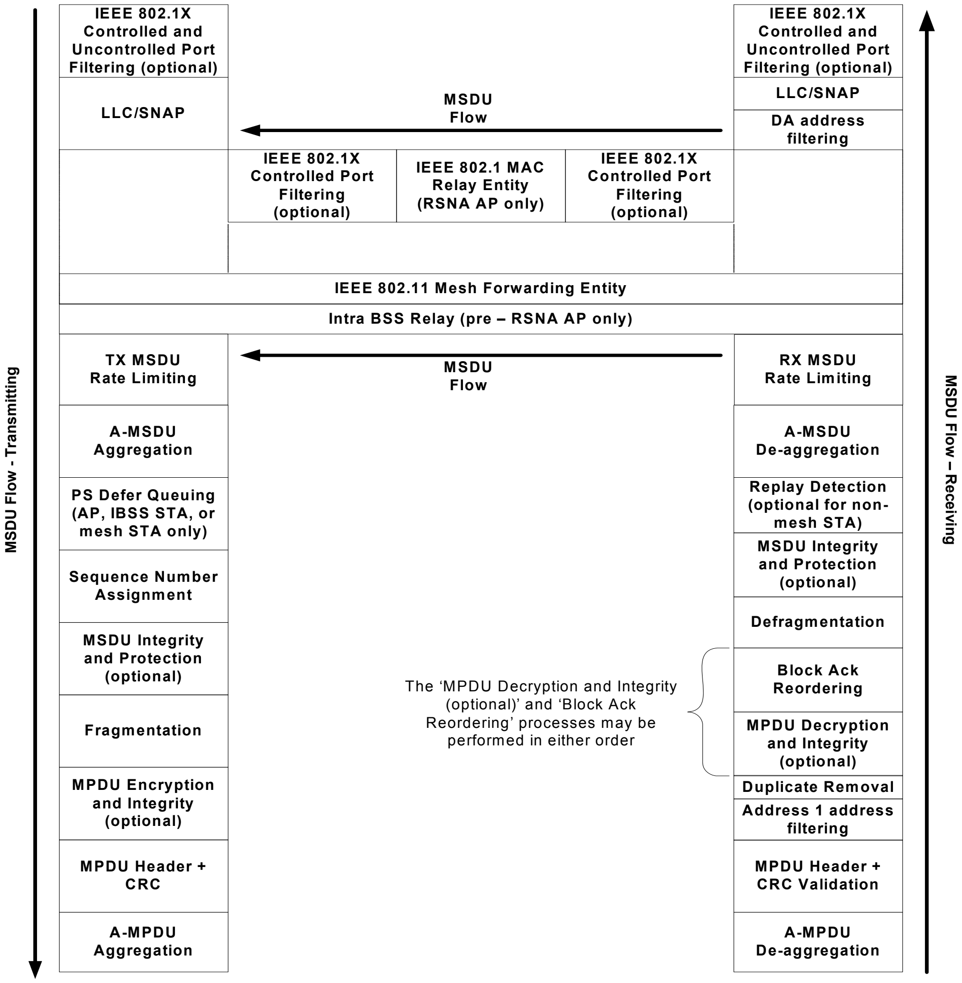

MSDU Flow - Transmitting
IEEE 802.1X Controlled and Uncontrolled Port Filtering (optional)
LLC/SNAP
MSDU Flow
IEEE 802.1X Controlled and Uncontrolled Port Filtering (optional)
LLC/SNAP
DA address filtering
IEEE 802.1X Controlled Port Filtering (optional)
IEEE 802.1 MAC Relay Entity (RSNA AP only)
IEEE 802.1X Controlled Port Filtering (optional)
IEEE 802.11 Mesh Forwarding Entity
Intra BSS Relay (pre – RSNA AP only)
TX MSDU Rate Limiting
MSDU Flow
RX MSDU Rate Limiting
A-MSDU Aggregation
A-MSDU De-aggregation
PS Defer Queuing (AP, IBSS STA, or mesh STA only)
Replay Detection (optional for non-mesh STA)
MSDU Integrity and Protection (optional)
Sequence Number Assignment
Defragmentation
MSDU Integrity and Protection (optional)
Block Ack Reordering
Fragmentation
The 'MPDU Decryption and Integrity (optional)' and 'Block Ack Reordering' processes may be performed in either order
MPDU Decryption and Integrity (optional)
Duplicate Removal
MPDU Encryption and Integrity (optional)
Address 1 address filtering
MPDU Header + CRC
MPDU Header + CRC Validation
A-MPDU Aggregation
A-MPDU De-aggregation
MSDU Flow – Receiving

Figure 5-1—MAC data plane architecture

5.2 MAC data service specification

5.2.1 General

The IEEE 802.11 MAC supports the following service primitives as defined in ISO/IEC 8802-2: 1998:

- MA-UNITDATA.request
- MA-UNITDATA.indication
- MA-UNITDATA-STATUS.indication

The LLC definitions of the primitives and specific parameter value restrictions imposed by IEEE Std 802.11 are given in 5.2.2 to 5.2.4.

5.2.2 MA-UNITDATA.request

5.2.2.1 Function

This primitive requests a transfer of an MSDU from a local LLC sublayer entity to a single peer LLC sublayer entity, or multiple peer LLC sublayer entities in the case of group addresses.

5.2.2.2 Semantics of the service primitive

The parameters of the primitive are as follows:

```
MA-UNITDATA.request(
                        source address,
                        destination address,
                        routing information,
                        data,
                        priority,
                        service class
                        )
```

The source address (SA) parameter specifies an individual MAC sublayer address of the sublayer entity from which the MSDU is being transferred.

The destination address (DA) parameter specifies either an individual or a group MAC sublayer entity address.

The routing information parameter specifies the route desired for the data transfer (a null value indicates source routing is not to be used). For IEEE Std 802.11, the routing information parameter shall be null.

The data parameter specifies the MSDU to be transmitted by the MAC sublayer entity. For IEEE Std 802.11, the length of the MSDU shall be less than or equal to 2304 octets.

The priority parameter specifies the priority desired for the data unit transfer. The allowed values of priority are described in 5.1.1.4.

The service class parameter specifies the service class desired for the data unit transfer. The allowed values of service class are described in 5.1.1.5 and 5.1.3.

5.2.2.3 When generated

This primitive is generated by the LLC sublayer entity when an MSDU is to be transferred to a peer LLC sublayer entity or entities.

5.2.2.4 Effect of receipt

On receipt of this primitive, the MAC sublayer entity determines whether it is able to fulfill the request according to the requested parameters. A request that cannot be fulfilled according to the requested parameters is discarded, and this action is indicated to the LLC sublayer entity using an MA-UNITDATA-STATUS.indication primitive that describes why the MAC was unable to fulfill the request. If the request can be fulfilled according to the requested parameters, the MAC sublayer entity appends all MAC specified fields (including DA, SA, FCS, and all fields that are unique to IEEE Std 802.11), passes the properly formatted frame to the lower layers for transfer to a peer MAC sublayer entity or entities (see 5.1.4), and indicates this action to the LLC sublayer entity using an MA-UNITDATA-STATUS.indication primitive with transmission status set to Successful.

At an AP for which dot11SSPNInterfaceActivated is true, upon receipt of an MA-UNITDATA.request primitive having an individually addressed destination address and a priority of Contention or ContentionFree, the AP's MAC sublayer shall perform rate limiting to enforce the resource utilization limit in dot11NonAPStationMaxAuthBestEffortRate in the dot11InterworkingEntry identified by the destination MAC address of the frame to be transmitted. The specific mechanism to perform rate limiting is outside the scope of this specification.

— If the rate limiting mechanism does not discard the frame, then dot11NonAPStationBestEffortMSDUCount shall be incremented by 1 and dot11NonAPStationBestEffortOctetCount shall be incremented by the number of octets in the MSDU.
— If the rate limiting mechanism discards the frame, then dot11NonAPStationDroppedBestEffortMSDUCount shall be incremented by 1 and dot11NonAPStationDroppedBestEffortOctetCount shall be incremented by the number of octets in the MSDU.

At an AP for which dot11SSPNInterfaceActivated is true, upon receipt of an MA-UNITDATA.request primitive having an individually addressed destination address for which the priority is an integer in the range of 0 to 7, inclusive, then the AP's MAC sublayer shall derive the access category from the priority using the mapping in Table 9-1. The AP's MAC sublayer shall retrieve the MIB variables listed below from the dot11InterworkingEntry identified by the destination MAC address of the frame to be transmitted and perform the following operations:

— If the access category is AC_VO, then the AP's MAC sublayer shall perform rate limiting to enforce the resource utilization limit in dot11NonAPStationMaxAuthVoiceRate; the specific mechanism to perform rate limiting is outside the scope of this specification. If the rate limiting mechanism does not discard the frame, then dot11NonAPStationVoiceMSDUCount shall be incremented by 1 and dot11NonAPStationVoiceOctetCount shall be incremented by the number of octets in the MSDU. If the rate limiting mechanism discards the frame, then dot11NonAPStationDroppedVoiceMSDUCount shall be incremented by 1 and dot11NonAPStationDroppedVoiceOctetCount shall be incremented by the number of octets in the MSDU.
— If the access category is AC_VI, then the AP's MAC sublayer shall perform rate limiting to enforce the resource utilization limit in dot11NonAPStationMaxAuthVideoRate; the specific mechanism to perform rate limiting is outside the scope of this specification. If the rate-limiting mechanism does not discard the frame, then dot11NonAPStationVideoMSDUCount shall be incremented by 1 and dot11NonAPStationVideoOctetCount shall be incremented by the number of octets in the MSDU. If the rate limiting mechanism discards the frame, then dot11NonAPStationDroppedVideoMSDUCount shall be incremented by 1 and dot11NonAPStationDroppedVideoOctetCount shall be incremented by the number of octets in the MSDU.
— If the access category is AC_BE, then the AP's MAC sublayer shall perform rate limiting to enforce the resource utilization limit in dot11NonAPStationMaxAuthBestEffortRate; the specific mechanism to perform rate limiting is outside the scope of this specification. If the rate-limiting mechanism does not discard the frame, then dot11NonAPStationBestEffortMSDUCount shall be incremented by 1 and dot11NonAPStationBestEffortOctetCount shall be incremented by the number of octets in the MSDU. If the rate limiting mechanism discards the frame, then

dot11NonAPStationDroppedBestEffortMSDUCount shall be incremented by 1 and dot11NonAPStationDroppedBestEffortOctetCount shall be incremented by the number of octets in the MSDU.

- If the access category is AC_BK, then the AP's MAC sublayer shall perform rate limiting to enforce the resource utilization limit in dot11NonAPStationMaxAuthBackgroundRate; the specific mechanism to perform rate limiting is outside the scope of this specification. If the rate-limiting mechanism does not discard the frame, then dot11NonAPStationBackgroundMSDUCount shall be incremented by 1 and dot11NonAPStationBackgroundOctetCount shall be incremented by the number of octets in the MSDU. If the rate limiting mechanism discards the frame, then dot11NonAPStationDroppedBackgroundMSDUCount shall be incremented by 1 and dot11NonAPStationDroppedBackgroundOctetCount shall be incremented by the number of octets in the MSDU.

At an AP for which dot11SSPNInterfaceActivated is true, upon receipt of an MA-UNITDATA.request primitive having an individually addressed destination address whose priority is an integer in the range of 8 to 15, inclusive, then the AP's MAC sublayer shall perform rate limiting to enforce the resource utilization limit in dot11NonAPStationAuthMaxHCCAHEMMRate; the specific mechanism to perform rate limiting is outside the scope of this specification.

- If the rate-limiting mechanism does not discard the frame, then dot11NonAPStationHCCAHEMM-MSDUCount shall be incremented by 1, and dot11NonAPStationHCCAHEMMOctetCount shall be incremented by the number of octets in the MSDU.
- If the rate limiting mechanism discards the frame, then dot11NonAPStationDroppedHCCAHEMM-MSDUCount shall be incremented by 1 and dot11NonAPStationDroppedHCCAHEMMOctetCount shall be incremented by the number of octets in the MSDU.

5.2.3 MA-UNITDATA.indication

5.2.3.1 Function

This primitive defines the transfer of an MSDU from the MAC sublayer entity to the LLC sublayer entity, or entities in the case of group addresses. In the absence of error, the contents of the data parameter are logically complete and unchanged relative to the data parameter in the associated MA-UNITDATA.request primitive.

5.2.3.2 Semantics of the service primitive

The parameters of the primitive are as follows:

```
MA-UNITDATA.indication(
                        source address,
                        destination address,
                        routing information,
                        data,
                        reception status,
                        priority,
                        service class
                        )
```

The SA parameter is an individual address as specified by the SA field of the incoming frame.

The DA parameter is either an individual or a group address as specified by the DA field of the incoming frame.

The routing information parameter specifies the route that was used for the data transfer. The MAC sublayer entity shall set this field to null.

The data parameter specifies the MSDU as received by the local MAC entity.

The reception status parameter indicates the success or failure of the received frame for those frames that IEEE Std 802.11 reports via a MA-UNITDATA.indication primitive. This MAC always reports "success" because all failures of reception are discarded without generating MA-UNITDATA.indication primitive.

The priority parameter specifies the receive processing priority that was used for the data unit transfer. The allowed values of priority are described in 5.1.1.4.

The service class parameter specifies the receive service class that was used for the data unit transfer. The allowed values of service class are described in 5.1.1.5 and 5.1.3.

5.2.3.3 When generated

The MA-UNITDATA.indication primitive is passed from the MAC sublayer entity to the LLC sublayer entity or entities to indicate the arrival of a frame at the local MAC sublayer entity. Frames are reported only if they are validly formatted at the MAC sublayer, received without error, received with valid (or null) security and integrity information, and their destination address designates the local MAC sublayer entity.

5.2.3.4 Effect of receipt

The effect of receipt of this primitive by the LLC sublayer is dependent on the content of the MSDU.

At an AP for which dot11SSPNInterfaceActivated is true, upon receipt of a frame of type data having a group DA, the AP's MAC sublayer shall discard the frame if dot11NonAPStationAuthSourceMulticast is false in the dot11InterworkingEntry identified by the source MAC address of the received frame. If dot11NonAPStationAuthSourceMulticast is true, the AP's MAC sublayer shall perform rate limiting to enforce the resource utilization limit in dot11NonAPStationAuthMaxSourceMulticastRate in the dot11InterworkingEntry identified by the source MAC address of the received frame. The specific mechanism to perform rate limiting is outside the scope of this specification.

— If the rate limiting mechanism does not discard the frame, then dot11NonAPStationMulticast-MSDUCount shall be incremented by 1 and dot11NonAPStationMulticastOctetCount shall be incremented by the number of octets in the MSDU.
— If the rate limiting mechanism discards the frame, then dot11NonAPStationDroppedMulticast-MSDUCount. shall be incremented by 1 and dot11NonAPStationDroppedMulticastOctetCount. shall be incremented by the number of octets in the MSDU.

At an AP for which dot11SSPNInterfaceActivated is true, upon receipt of an individually addressed frame of type data and a priority of Contention or ContentionFree, then the AP's MAC sublayer shall perform rate limiting to enforce the resource utilization limit in dot11NonAPStationMaxAuthBestEffortRate in the dot11InterworkingEntry identified by the source MAC address of the received frame. The specific mechanism to perform rate limiting is outside the scope of this specification.

— If the rate limiting mechanism does not discard the frame, then dot11NonAPStationBestEffort-MSDUCount shall be incremented by 1 and dot11NonAPStationBestEffortOctetCount shall be incremented by the number of octets in the MSDU.
— If the rate limiting mechanism discards the frame, then dot11NonAPStationDroppedBestEffort-MSDUCount shall be incremented by 1 and dot11NonAPStationDroppedBestEffortOctetCount shall be incremented by the number of octets in the MSDU.

At an AP for which dot11SSPNInterfaceActivated is true, upon receipt of an individually addressed frame of type data, for which the priority is an integer in the range of 0 to 7, inclusive, then the AP's MAC sublayer shall derive the access category from the priority using the mapping in Table 9-1. The AP's MAC sublayer shall retrieve the MIB variables from the dot11InterworkingEntry identified by the source MAC address of the received frame and perform the following operations:

- If the access category is AC_VO, then the AP's MAC sublayer shall perform rate limiting to enforce the resource utilization limit in dot11NonAPStationMaxAuthVoiceRate; the specific mechanism to perform rate limiting is outside the scope of this specification. If the rate-limiting mechanism does not discard the frame, then dot11NonAPStationVoiceMSDUCount shall be incremented by 1 and dot11NonAPStationVoiceOctetCount shall be incremented by the number of octets in the MSDU. If the rate limiting mechanism discards the frame, then dot11NonAPStationDroppedVoiceMSDUCount shall be incremented by 1 and dot11NonAPStationDroppedVoiceOctetCount shall be incremented by the number of octets in the MSDU.
- If the access category is AC_VI, then the AP's MAC sublayer shall perform rate limiting to enforce the resource utilization limit in dot11NonAPStationMaxAuthVideoRate; the specific mechanism to perform rate limiting is outside the scope of this specification. If the rate-limiting mechanism does not discard the frame, then dot11NonAPStationVideoMSDUCount shall be incremented by 1 and dot11NonAPStationVideoOctetCount shall be incremented by the number of octets in the MSDU. If the rate limiting mechanism discards the frame, then dot11NonAPStationDroppedVideoMSDUCount shall be incremented by 1 and dot11NonAPStationDroppedVideoOctetCount shall be incremented by the number of octets in the MSDU.
- If the access category is AC_BE, then the AP's MAC sublayer shall perform rate limiting to enforce the resource utilization limit in dot11NonAPStationMaxAuthBestEffortRate; the specific mechanism to perform rate limiting is outside the scope of this specification. If the rate-limiting mechanism does not discard the frame, then dot11NonAPStationBestEffortMSDUCount shall be incremented by 1 and dot11NonAPStationBestEffortOctetCount shall be incremented by the number of octets in the MSDU. If the rate limiting mechanism discards the frame, then dot11NonAPStationDroppedBestEffortMSDUCount shall be incremented by 1 and dot11NonAPStationDroppedBestEffortOctetCount shall be incremented by the number of octets in the MSDU.
- If the access category is AC_BK, then the AP's MAC sublayer shall perform rate limiting to enforce the resource utilization limit in dot11NonAPStationMaxAuthBackgroundRate; the specific mechanism to perform rate limiting is outside the scope of this specification. If the rate-limiting mechanism does not discard the frame, then dot11NonAPStationBackgroundMSDUCount shall be incremented by 1 and dot11NonAPStationBackgroundOctetCount shall be incremented by the number of octets in the MSDU. If the rate limiting mechanism discards the frame, then dot11NonAPStationDroppedBackgroundMSDUCount shall be incremented by 1 and dot11NonAPStationDroppedBackgroundOctetCount shall be incremented by the number of octets in the MSDU.

At an AP for which dot11SSPNInterfaceActivated is true, upon receipt of an individually addressed frame of type data for which the priority is an integer in the range of 8 to 15, inclusive, the AP's MAC sublayer shall perform rate limiting to enforce the resource utilization limit in dot11NonAPStationAuthMaxHCCAHEMMRate; the specific mechanism to perform rate limiting is outside the scope of this specification.

- If the rate-limiting mechanism does not discard the frame, then dot11NonAPStationHCCAHEMMMSDUCount shall be incremented by 1, and dot11NonAPStationHCCAHEMMOctetCount shall be incremented by the number of octets in the MSDU.
- If the rate limiting mechanism discards the frame, then dot11NonAPStationDroppedHCCAHEMMMSDUCount shall be incremented by 1 and dot11NonAPStationDroppedHCCAHEMMOctetCount shall be incremented by the number of octets in the MSDU.

5.2.4 MA-UNITDATA-STATUS.indication

5.2.4.1 Function

This primitive has local significance and provides the LLC sublayer with status information for the corresponding preceding MA-UNITDATA.request primitive.

5.2.4.2 Semantics of the service primitive

The parameters of the primitive are as follows:

```
MA-UNITDATA-STATUS.indication(
                                   source address,
                                   destination address,
                                   transmission status,
                                   provided priority,
                                   provided service class
                                   )
```

The SA parameter is an individual MAC sublayer entity address as specified in the associated MA-UNITDATA.request primitive.

The DA parameter is either an individual or group MAC sublayer entity address as specified in the associated MA-UNITDATA.request primitive.

The transmission status parameter is used to pass status information back to the local requesting LLC sublayer entity. IEEE Std 802.11 specifies the following values for transmission status:

a) Successful.

b) Undeliverable (excessive data length).

c) Undeliverable (non-null source routing).

d) Undeliverable: unsupported priority (for priorities other than Contention or ContentionFree at a non-QoS STA; or for priorities other than Contention, ContentionFree, or an integer between and including 0 and 15 at a QoS STA).

e) Undeliverable: unsupported service class (for service classes other than ReorderableGroupAddressed or StrictlyOrdered for non-QoS STAs and service classes other than QoSAck or QoSNoAck for QoS STAs).

f) Unavailable priority (for ContentionFree when no PC or HC is available, or an integer between and including 1 and 15 at a STA that is associated in a non-QoS BSS, or an integer between and including 8 and 15 at a STA that is a member of an IBSS, in which case the MSDU is transmitted with a provided priority of Contention).

g) Undeliverable: unavailable service class (for StrictlyOrdered service when the STA's power management mode is other than "active" for non-QoS STAs; QoS STAs do not return this value as they do not provide the StrictlyOrdered service).

h) Undeliverable (no BSS available).

i) Undeliverable (cannot encrypt with a null key).

j) At a STA where dot11RejectUnadmittedTraffic is true, Undeliverable: un-admitted traffic (for a requested priority between and including 0 and 7 at a STA because there is no admitted TS for this priority and admission control is required for the AC).

k) For an AP in which dot11SSPNInterfaceActivated is true, Undeliverable (violation of limit specified by dot11NonAPStationMaxAuthVoiceRate in the dot11InterworkingTable for the non-AP STA identified by the destination address of the MA-UNITDATA.request primitive).

l) For an AP in which dot11SSPNInterfaceActivated is true, Undeliverable (violation of limit specified by dot11NonAPStationMaxAuthVideoRate in the dot11InterworkingTable for the non-AP STA identified by the destination address of the MA-UNITDATA.request primitive).

m) For an AP in which dot11SSPNInterfaceActivated is true, Undeliverable (violation of limit specified by dot11NonAPStationMaxAuthBestEffortRate in the dot11InterworkingTable for the non-AP STA identified by the destination address of the MA-UNITDATA.request primitive).

n) For an AP in which dot11SSPNInterfaceActivated is true, Undeliverable (violation of limit specified by dot11NonAPStationBackgroundRate in the dot11InterworkingTable for the non-AP STA identified by the destination address of the MA-UNITDATA.request primitive).

o) For an AP in which dot11SSPNInterfaceActivated is true, Undeliverable (violation of limit specified by dot11nonAPStationAuxMaxHCCAHEMMrate in the dot11InterworkingTable for the non-AP STA identified by the destination address of the MA-UNITDATA.request primitive).

If the transmission status parameter is Successful, the provided priority parameter specifies the priority used for the associated data unit transfer (Contention, ContentionFree, or an integer between and including 0 and 15); otherwise the provided priority parameter is not present.

If the transmission status parameter is Successful, the provided service class parameter specifies the class of service for the associated data unit transfer; otherwise the provided service class parameter is not present. In non-QoS STAs, the value of this parameter is ReorderableGroupAddressed or StrictlyOrdered. In QoS STAs, it is QoSAck or QoSNoAck.

5.2.4.3 When generated

The MA-UNITDATA-STATUS.indication primitive is passed from the MAC sublayer entity to the LLC sublayer entity to indicate the status of the service provided for the corresponding MA-UNITDATA.request primitive.

5.2.4.4 Effect of receipt

The effect of receipt of this primitive by the LLC sublayer is dependent upon the type of operation employed by the LLC sublayer entity.

6. Layer management

6.1 Overview of management model

Both the MAC sublayer and PHY conceptually include management entities, called MLME and PLME, respectively. These entities provide the layer management service interfaces through which layer management functions are invoked.

In order to provide correct MAC operation, an SME is present within each STA. The SME is a layer-independent entity that resides in a separate management plane or resides "off to the side." Some of the functions of the SME are specified in this standard. Typically this entity is responsible for such functions as the gathering of layer-dependent status from the various layer management entities (LMEs), and similarly setting the value of layer-specific parameters. The SME would typically perform such functions on behalf of general system management entities and would implement standard management protocols. Figure 4-14 (in 4.9) depicts the relationship among management entities.

The various entities within this model interact in various ways. Certain of these interactions are defined explicitly within this standard, via a SAP across which defined primitives are exchanged. This definition includes the GET and SET operations between MLME, PLME and SME as well as other individually defined service primitives, represented as double arrows within Figure 6-1. Other interactions are not defined explicitly within this standard, such as the interfaces between the MAC and MLME and between the PLME and PLCP and PMD; the specific manner in which these MAC and PHY LMEs are integrated into the overall MAC sublayer and PHY is not specified within this standard.

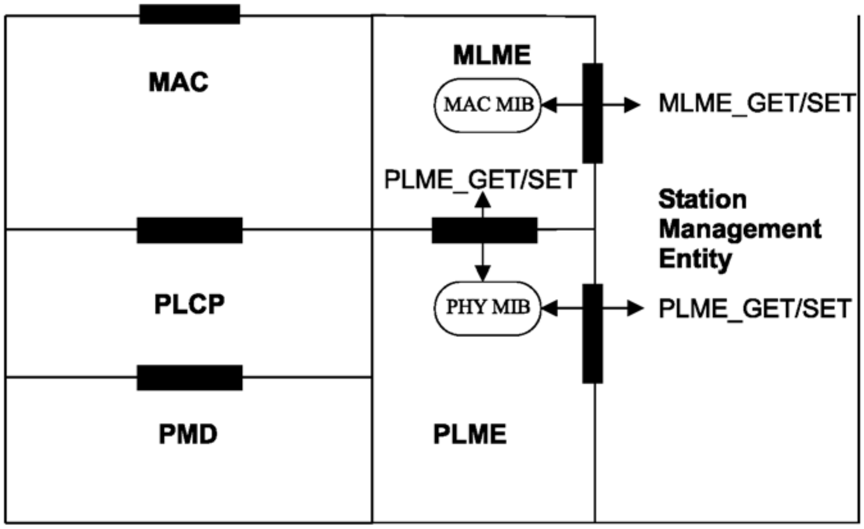


Figure 6-1—GET and SET operations

The management SAPs within this model are the following:

- SME-MLME SAP
- SME-PLME SAP
- MLME-PLME SAP

The latter two SAPs support identical primitives, and in fact might be viewed as a single SAP (called the PLME SAP) that is used either directly by MLME or by SME. In this fashion, the model reflects what is anticipated to be a common implementation approach in which PLME functions are controlled by the MLME (on behalf of SME).

6.2 Generic management primitives

The management information specific to each layer is represented as a MIB for that layer. The MLME and PLME are viewed as "containing" the MIB for that layer. The generic model of MIB-related management primitives exchanged across the management SAPs is to allow the SAP user-entity to either GET the value of a MIB attribute, or to SET the value of a MIB attribute. The invocation of a SET.request primitive might require that the layer entity perform certain defined actions.

Figure 6-1 depicts these generic primitives.

The GET and SET primitives are represented as REQUESTs with associated CONFIRM primitives. These primitives are prefixed by MLME or PLME depending upon whether the MAC sublayer or PHY management SAP is involved. In the following, XX denotes MLME or PLME:

XX-GET.request(MIBattribute)

Requests the value of the given MIBattribute.

XX-GET.confirm(status, MIBattribute, MIBattributevalue)

Returns the appropriate MIB attribute value if status = "success," otherwise returns an error indication in the Status field. Possible error status values include "invalid MIB attribute" and "attempt to get write-only MIB attribute."

XX-SET.request(MIBattribute, MIBattributevalue)

Requests that the indicated MIB attribute be set to the given value. If this MIBattribute implies a specific action, then this requests that the action be performed.

XX-SET.confirm(status, MIBattribute)

If status = "success," this confirms that the indicated MIB attribute was set to the requested value; otherwise it returns an error condition in status field. If this MIBattribute implies a specific action, then this confirms that the action was performed. Possible error status values include "invalid MIB attribute" and "attempt to set read-only MIB attribute."

Additionally, there are certain requests that can be invoked across a given SAP that do not involve the setting or getting of a specific MIB attribute. One of these is supported by each SAP, as follows:

- XX-RESET.request: where XX is MLME or PLME as appropriate

This service is used to initialize the management entities, the MIBs, and the datapath entities. It might include a list of attributes for items to be initialized to nondefault values.

Other SAP-specific primitives are identified in 6.3.

6.3 MLME SAP interface

6.3.1 Introduction

The services provided by the MLME to the SME are specified in this subclause. These services are described in an abstract way (following the model described in ITU-T Recommendation X.210 [B50]) and do not imply any particular implementation or exposed interface. MLME SAP primitives are of the general form ACTION.request followed by ACTION.confirm (for an exchange initiated by the SAP client) and

ACTION.indication followed by ACTION.response (for an exchange initiated by the MLME). The SME uses the services provided by the MLME through the MLME SAP.

6.3.2 Power management

6.3.2.1 Introduction

This mechanism supports the process of establishment and maintenance of the power management mode of a STA.

6.3.2.2 MLME-POWERMGT.request

6.3.2.2.1 Function

This primitive requests a change in the power management mode.

6.3.2.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-POWERMGT.request(
                              PowerManagementMode,
                              WakeUp,
                              ReceiveDTIMs
                              )
```

Name	Type	Valid range	Description
PowerManagementMode	Enumeration	ACTIVE, POWER_SAVE	An enumerated type that describes the desired power management mode of the STA.
WakeUp	Boolean	true, false	When true, the MAC is forced immediately into the Awake state. This parameter has no effect if the current power management mode is ACTIVE.
ReceiveDTIMs	Boolean	true, false	When true, this parameter causes the STA to awaken to receive all DTIM frames. When false, the STA is not required to awaken for every DTIM frame.

6.3.2.2.3 When generated

This primitive is generated by the SME to implement the power-saving strategy of an implementation.

6.3.2.2.4 Effect of receipt

This request sets the STA’s power management parameters. The MLME subsequently issues a MLME-POWERMGT.confirm primitive that reflects the results of the power management change request.

6.3.2.3 MLME-POWERMGT.confirm

6.3.2.3.1 Function

This primitive confirms the change in power management mode.

6.3.2.3.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-POWERMGT.confirm(
    ResultCode
)

Name	Type	Valid range	Description
ResultCode	Enumeration	SUCCESS, NOT_SUPPORTED	Indicates the result of the MLME-POWERMGT.request.

6.3.2.3.3 When generated

This primitive is generated by the MLME as a result of an MLME-POWERMGT.request primitive to establish a new power management mode. It is not generated until the change has completed as defined in 10.2.1.

6.3.2.3.4 Effect of receipt

The SME is notified of the change of power management mode.

6.3.3 Scan

6.3.3.1 Introduction

This mechanism supports the process of determining the characteristics of the available BSSs.

6.3.3.2 MLME-SCAN.request

6.3.3.2.1 Function

This primitive requests a survey of potential BSSs that the STA can later elect to try to join.

6.3.3.2.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-SCAN.request(
    BSSType,
    BSSID,
    SSID,
    ScanType,
    ProbeDelay,
    ChannelList,
    MinChannelTime,
    MaxChannelTime,
    RequestInformation,
    SSID List,
    ChannelUsage,
    AccessNetworkType,
    HESSID,
    MeshID,
    VendorSpecificInfo
)

Name	Type	Valid range	Description
BSSType	Enumeration	INFRASTRUCTURE, INDEPENDENT, MESH, ANY_BSS	Determines whether infrastructure BSS, IBSS, MBSS, or all, are included in the scan.
BSSID	MACAddress	Any valid individual or broadcast MAC address	Identifies a specific or wildcard BSSID.
SSID	Octet string	0–32 octets	Specifies the desired SSID or the wildcard SSID.
ScanType	Enumeration	ACTIVE, PASSIVE	Indicates either active or passive scanning.
ProbeDelay	Integer	N/A	Delay (in microseconds) to be used prior to transmitting a Probe frame during active scanning.
ChannelList	Ordered set of integers	Each channel is selected from the valid channel range for the appropriate PHY and carrier set.	Specifies a list of channels that are examined when scanning for a BSS.
MinChannelTime	Integer	N/A	The minimum time (in TU) to spend on each channel when scanning.
MaxChannelTime	Integer	≥ MinChannelTime	The maximum time (in TU) to spend on each channel when scanning.
RequestInformation	As defined in 8.4.2.13	As defined in 8.4.2.13	This element is optionally present if dot11RadioMeasurementActivated is true and is placed in a Probe Request frame to request that the responding STA include the requested information in the Probe Response frame.
SSID List	A set of SSID Element	As defined in 8.4.2.2.	One or more SSID elements that are optionally present when dot11MgmtOptionSSIDListActivated is true.
ChannelUsage	A set of Channel Usage element	As defined in 8.4.2.88	Specifies request types for the Channel Usage request.
AccessNetworkType	As defined in Table 8-174	0 to 15	Specifies a desired specific access network type or the wildcard access network type. This field is present when dot11InterworkingServiceActivated is true.
HESSID	MAC Address	Any valid individual MAC address or the broadcast MAC address	Specifies the desired specific HESSID network identifier or the wildcard network identifier. This field is present when dot11InterworkingServiceActivated is true.
Mesh ID	Octet string	0–32 octets	Only present if BSSType = MESH or BSSType = ANY_BSS. Specifies the desired Mesh ID or wildcard Mesh ID.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.3.2.3 When generated

This primitive is generated by the SME for a STA to determine if there are other BSSs that it can join.

6.3.3.2.4 Effect of receipt

This request initiates the scan process when the current frame exchange sequence is completed.

6.3.3.3 MLME-SCAN.confirm

6.3.3.3.1 Function

This primitive returns the descriptions of the set of BSSs detected by the scan process.

6.3.3.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-SCAN.confirm(
                              BSSDescriptionSet,
                              BSSDescriptionFromMeasurementPilotSet,
                              ResultCode,
                              VendorSpecificInfo
                              )
```

Name	Type	Valid range	Description
BSSDescriptionSet	Set of BSSDescriptions	N/A	The BSSDescriptionSet is returned to indicate the results of the scan request. It is a set containing zero or more instances of a BSSDescription.
BSSDescriptionFromMeasurementPilotSet	Set of BSS DescriptionFrom MeasurementPilots	N/A	The BSSDescriptionFromMeasurementPilotSet is returned to indicate the results of the scan request derived from measurement pilots. It is a set containing zero or more instances of a BSSDescriptionFrom-MeasurementPilot. Present only if the value of dot11RMMeasurementPilotActivated is nonzero.
ResultCode	Enumeration	SUCCESS, NOT_SUPPORTED	Indicates the result of the MLME-SCAN.confirm primitive.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

Each BSSDescription consists of the elements shown in the following table, in which the term *peer STA* refers to the STA transmitting the Beacon frame or Probe Response frame from which the BSSDescription was determined and the term *local STA* refers to the STA performing the scan, and in which the “IBSS adoption” column indicates whether

a) This parameter is adopted by a STA that is joining an IBSS.

b) This parameter is adopted by a STA that is a member of an IBSS that receives a beacon from a STA that is a member of the same IBSS and that has a timestamp value that is greater than the local TSF value (see 10.1.5).

Name	Type	Valid range	Description	IBSS adoption
BSSID	MACAddress	N/A	The BSSID of the found BSS or the MAC address of the found mesh STA.	Adopt
SSID	Octet string	0–32 octets	The SSID of the found BSS.	Adopt
BSSType	Enumeration	INFRASTRUCTURE, INDEPENDENT, MESH	The type of the found BSS.	Adopt

Name	Type	Valid range	Description	IBSS adoption
Beacon Period	Integer	N/A	The Beacon period (in TU) of the found BSS if the BSSType is not MESH, or of the found mesh STA if the BSSType = MESH.	Adopt
DTIM Period	Integer	As defined in 8.4.2.7	The DTIM period (in beacon periods) of the BSS if the BSSType is not MESH, or of the mesh STA if the BSSType = MESH.	Adopt
Timestamp	Integer	N/A	The timestamp of the received frame (probe response/beacon) from the found BSS.	Adopt
Local Time	Integer	N/A	The value of the local STA's TSF timer at the start of reception of the first octet of the timestamp field of the received frame (probe response or beacon) from the found BSS.	Do not adopt
PHY Parameter Set	As defined in frame format or according to the relevant PHY clause.	As defined in frame format or according to the relevant PHY clause.	The parameter sets relevant to the PHY from the received Beacon or Probe Response frame. If no PHY Parameter Set element is present in the received frame, this parameter contains the channel number on which the frame was received. Valid channel numbers are defined in the relevant PHY clause.	Adopt
CF Parameter Set	CF Parameter Set element	As defined in 8.4.2.6	The parameter set for the CF periods, if found BSS supports CF mode.	Do not adopt
IBSS Parameter Set	IBSS Parameter Set element	As defined in 8.4.2.8	The parameter set for the IBSS, if found BSS is an IBSS.	Adopt
CapabilityInformation	Capability Information field	As defined in 8.4.1.4	The advertised capabilities of the BSS.	Do not adopt
BSSBasicRateSet	Set of integers	1–127 inclusive (for each integer in the set)	The set of data rates that shall be supported by all STAs that desire to join this BSS.	Adopt
OperationalRateSet	Set of integers	1–127 inclusive (for each integer in the set)	The set of data rates that the peer STA desires to use for communication within the BSS. This set is a superset of the rates contained in the BSSBasicRateSet parameter.	Do not adopt

Name	Type	Valid range	Description	IBSS adoption
Country	As defined in the Country element	As defined in the Country element	The information required to identify the regulatory domain in which the peer STA is located and to configure its PHY for operation in that regulatory domain. Present only if TPC functionality is required, as specified in 10.8, or dot11MultiDomainCapabilityActivated is true or dot11RadioMeasurementActivated is true.	Adopt
IBSS DFS Recovery Interval	Integer	1–255	The time interval that is used for DFS recovery. Present only if DFS functionality is required, as specified in 10.9. Present only if BSSType = INDEPENDENT.	Adopt
RSN	RSN element (RSNE)	As defined in 8.4.2.27	A description of the cipher suites and AKM suites supported in the BSS.	Do not adopt
Load	BSS Load element	As defined in 8.4.2.30	The values from the BSS Load element if such an element was present in the probe response or Beacon frame, else null.	Do not adopt
EDCAParameterSet	EDCA Parameter Set element	As defined in 8.4.2.31	The values from the EDCA Parameter Set element if such an element was present in the probe response or Beacon frame, else null.	Adopt
QoSCapability	QoS Capability element	As defined in 8.4.2.37	The values from the QoS Capability element if such an element was present in the probe response or Beacon frame, else null.	Adopt
AP Channel Report	AP Channel Report element	As defined in 8.4.2.38	The values from the AP Channel Report element if such an element was present in the probe response or Beacon frame, else null.	Adopt
BSS Average Access Delay	BSS Average Access Delay element	As defined in 8.4.2.41	The values from the BSS Average Access Delay element if such an element was present in the probe response or Beacon frame, else null.	Adopt
Antenna	Antenna element	As defined in 8.4.2.42	The values from the Antenna element if such an element was present in the probe response or Beacon frame, else null.	Adopt
BSS Available Admission Capacity	BSS Available Admission Capacity element	As defined in 8.4.2.45	The values from the BSS Available Admission Capacity element if such an element was present in the probe response or Beacon frame, else null.	Adopt

Name	Type	Valid range	Description	IBSS adoption
BSS AC Access Delay	BSS AC Access Delay element	As defined in 8.4.2.46	The values from the BSS AC Access Delay element if such an element was present in the probe response or Beacon frame, else null.	Adopt
Measurement Pilot Transmission Information	Measurement Pilot Transmission element	As defined in 8.4.2.44	The values from the Measurement Pilot Transmission element if such an element was present in the probe response or Beacon frame, else null.	Adopt
Multiple BSSID	As defined in Multiple BSSID element	As defined in 8.4.2.48	The values from the Multiple BSSID element if such an element was present in the probe response or Beacon frame, else null.	Do not adopt
RM Enabled Capabilities	RM Enabled Capabilities element	As defined in 8.4.2.47	The values from the RM Enabled Capabilities element if such an element was present in the probe response or Beacon frame, else null.	Adopt
RCPIMeasurement	Integer	As defined in 16.4.8.6, or 18.3.10.7, or 17.4.8.6	The RCPI of the received frame.	Adopt
RSNIMeasurement	Integer	As defined in 8.4.2.43	The RSNI of the received frame.	Adopt
Requested elements	Set of elements	As defined in 8.4.1.35	Elements requested by the Request element of the Probe Request frame.	Adopt
DSERegisteredLocation	DSE Registered Location element	As defined in 8.4.2.54	The information from the DSE Registered Location element, if such a field is present in Probe Response or Beacon, else null. Present only if DSE functionality is required, as specified in 10.12, or dot11LCIDSERequired is true.	Adopt
HT Capabilities	As defined in frame format	As defined in 8.4.2.58	The values from the HT Capabilities element if such an element was present in the Probe Response or Beacon frame, else null. The parameter is optionally present only if dot11HighThroughputOptionImplemented is true.	Do not adopt
HT Operation	As defined in frame format	As defined in 8.4.2.59	The values from the HT Operation element if such an element was present in the Probe Response or Beacon frame, else null. The parameter is optionally present only if dot11HighThroughputOptionImplemented is true.	Adopt

Name	Type	Valid range	Description	IBSS adoption
BSSMembershipSelectorSet	Set of integers	A value from Table 8-55 for each member of the set	The BSS membership selectors that represent the set of features that shall be supported by all STAs to join this BSS.	Adopt
BSSBasicMCSSet	Set of integers	Each member of the set takes a value in the range 0 to 76, representing an MCS index value	The set of MCS values that shall be supported by all HT STAs to join this BSS. The STA that is creating the BSS shall be able to receive and transmit at each of the MCS values listed in the set.	Adopt
HTOperationalMCSSet	Set of integers	Each member of the set takes a value in the range 0 to 76, representing an MCS index value	The set of MCS values that the peer STA desires to use for communication within the BSS. The STA shall be able to receive at each of the data rates listed in the set. This set is a superset of the MCS values contained in the BSSBasicMCSSet parameter.	Do not adopt
Extended Capabilities	As defined in frame format	As defined in 8.4.2.29	Specifies the parameters within the Extended Capabilities element that are supported by the MAC entity.	Do not adopt
20/40 BSS Coexistence	As defined in frame format	As defined in 8.4.2.62	Specifies the parameters within the 20/40 BSS Coexistence element that are indicated by the MAC entity. The parameter is present if dot112040BSSCoexistence-ManagementSupport is true.	Do not adopt
Overlapping BSS Scan Parameters	As defined in frame format	As defined in 8.4.2.61	Specifies the parameters within the Overlapping BSS Scan Parameters element that are indicated by the MAC entity. This parameter is optionally present if dot11FortyMHzOptionImplemented is true and is not present if dot11FortyMHzOptionImplemented is false.	Adopt
FMSDescriptor	FMS Descriptor element	As defined in 8.4.2.77	The values from the FMS Descriptor element if such an element was present in the Probe Response or Beacon frame, else null.	Do not adopt
QoSTrafficCapability	QoS Traffic Capability element	As defined in 8.4.2.80	The values from the QoS Traffic Capability element if such an element was present in the Probe Response or Beacon frame, else null.	Do not adopt
ChannelUsage	A set of Channel Usage element	As defined in 8.4.2.88	Specifies parameters for the Channel Usage.	Do not adopt

Name	Type	Valid range	Description	IBSS adoption
TimeAdvertisement	Time Advertisement element	As defined in 8.4.2.63	The values from the Time Advertisement element if such an element was present in the Probe Response or Beacon frame, else null.	Do not adopt
TimeZone	Time Zone element	As defined in 8.4.2.89	The values from the Time Zone element if such an element was present in the Probe Response or Beacon frame, else null.	Do not adopt
MeshID	Mesh ID element	As defined in 8.4.2.101	The value of MeshID element if such element was present in the probe response or Beacon frame, else null.	Do not adopt
MeshConfiguration	Mesh Configuration element	As defined in 8.4.2.100	The values from the Mesh Configuration element if such an element was present in the probe response or Beacon frame, else null.	Do not adopt
Mesh Awake Window	Mesh Awake Window element	As defined in 8.4.2.106	The values from the Mesh Awake Window element if such an element was present in the Probe response or Beacon frame, else null.	Do not adopt
BeaconTiming	Beacon Timing element	As defined in 8.4.2.107	The values from the Beacon Timing element if such an element was present in the Probe response or Beacon frame, else null.	Do not adopt
MCCAOP Advertisement Overview	MCCAOP Advertisement Overview element	As defined in frame format	The values from the Beacon Timing element if such an element was present in the Probe response or Beacon frame, else null.	Do not adopt
MCCAOP Advertisement	MCCAOP Advertisement	As defined in frame format	The values from the Beacon Timing element if such an element was present in the Probe response or Beacon frame, else null.	Do not adopt

Each BSSDescriptionFromMeasurementPilot consists of the following elements, in which the term *peer STA* refers to the STA transmitting the Measurement Pilot frame from which the BSSDescription was determined and the term *local STA* refers to the STA performing the scan:

Name	Type	Valid range	Description
BSSID	MACAddress	N/A	The BSSID of the found BSS.
BSS Type	Enumeration	INFRASTRUCTURE	The type of the found BSS.
Local Time	Integer	N/A	The value of the local STA's TSF timer at the start of reception of the first octet of the timestamp field of the received frame from the found BSS.
Condensed Capability Information	Condensed Capability Information field	As defined in 8.5.8.3	The advertised condensed capabilities of the BSS.

Name	Type	Valid range	Description
Condensed Country String	Condensed Country String field	As defined in 8.5.8.3	Together with the Operating Class, the information required to identify the regulatory domain in which the peer STA is located and to configure its PHY for operation in that regulatory domain.
Operating Class	Operating Class field	The field is defined in 8.5.8.3, valid values for the field are defined in Annex E.	Together with the Condensed Country String, the information required to identify the regulatory domain in which the peer STA is located and to configure its PHY for operation in that regulatory domain.
Channel	Channel field	The field is defined in 8.5.8.3, valid values for the field are defined in Annex E.	The operating channel of the BSS indicated in the received frame
Measurement Pilot Interval	Measurement Pilot Interval field	As defined in 8.5.8.3	The Measurement Pilot interval of the BSS indicated in the received frame
Multiple BSSID element	Multiple BSSID element	As defined in 8.4.2.48	Indicates that the BSS is within a Multiple BSSID Set (see 10.11.14). The range of BSSIDs is determined by the BSSID and Multiple BSSID element.
PHY Type	Integer	As defined in Annex C	The dot11PHYType of the received frame.
RCPIMeasurement	Integer	As defined in 16.4.8.6, or 18.3.10.7, or 17.4.8.6	The RCPI of the received frame.
RSNIMeasurement	Integer	As defined in 16.4.8.6, or 18.3.10.7, or 17.4.8.6	The RSNI of the received frame.

6.3.3.3.3 When generated

This primitive is generated by the MLME as a result of an MLME-SCAN.request primitive to ascertain the operating environment of the STA.

6.3.3.3.4 Effect of receipt

The SME is notified of the results of the scan procedure.

6.3.4 Synchronization

6.3.4.1 Introduction

This mechanism supports the process of selection of a peer in the authentication process.

6.3.4.2 MLME-JOIN.request

6.3.4.2.1 Function

This primitive requests synchronization with a BSS, of which type is infrastructure or independent.

6.3.4.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-JOIN.request(
                        SelectedBSS,
                        JoinFailureTimeout,
                        ProbeDelay,
                        OperationalRateSet,
                        HTOperationalMCSSet,
                        VendorSpecificInfo
                        )
```

Name	Type	Valid range	Description
SelectedBSS	BSSDescription	N/A	The BSSDescription of the BSS to join. The SelectedBSS is a member of the set of descriptions that was returned as a result of a MLME-SCAN.request primitive.
JoinFailureTimeout	Integer	≥ 1	The time limit, in units of beacon intervals, after which the join procedure is terminated.
ProbeDelay	Integer	N/A	Delay (in microseconds) to be used prior to transmitting when changing from Doze to Awake, if no frame sequence is detected by which the NAV can be set.
OperationalRateSet	Set of integers	1–127 inclusive (for each integer in the set)	The set of data rates that the STA desires to use for communication within the BSS. The STA shall be able to receive at each of the data rates listed in the set. This set is a superset of the rates contained in the BSSBasicRateSet parameter.
HTOperationalMCS Set	Set of integers	0–76, representing an MCS index value (for each member of the set)	The set of MCS values that the STA desires to use for communication within the BSS. The STA shall be able to receive at each of the data rates listed in the set. This set is a superset of the MCS values contained in the BSSBasicMCSSet parameter.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.4.2.3 When generated

This primitive is generated by the SME for a STA to establish synchronization with a BSS.

6.3.4.2.4 Effect of receipt

This primitive initiates a synchronization procedure once the current frame exchange sequence is complete. The MLME synchronizes its timing with the specified BSS based on the elements provided in the SelectedBSS parameter. The MLME subsequently issues a MLME-JOIN.confirm primitive that reflects the results.

If an MLME receives an MLME-JOIN.request primitive with the SelectedBSS parameter containing a BSSBasicRateSet element that contains any unsupported rates, the MLME response in the resulting MLME-JOIN.confirm primitive shall contain a ResultCode parameter that is not set to the value SUCCESS.

If the MLME of an HT STA receives an MLME-JOIN.request primitive with the SelectedBSS parameter containing a BSSBasicMCSSet value that contains any unsupported MCSs, the MLME response in the resulting MLME-JOIN.confirm primitive shall contain a ResultCode parameter that is not set to the value SUCCESS.

6.3.4.3 MLME-JOIN.confirm

6.3.4.3.1 Function

This primitive confirms synchronization with a BSS.

6.3.4.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
  MLME-JOIN.confirm(
                                    ResultCode,
                                    VendorSpecificInfo
                                    )
```

Name	Type	Valid range	Description
ResultCode	Enumeration	SUCCESS, INVALID_PARAMETERS	Indicates the result of the MLME-JOIN.request primitive.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.4.3.3 When generated

This primitive is generated by the MLME as a result of an MLME-JOIN.request primitive to establish synchronization with a BSS.

6.3.4.3.4 Effect of receipt

The SME is notified of the results of the synchronization procedure.

6.3.5 Authenticate

6.3.5.1 Introduction

This mechanism supports the process of establishing an authentication relationship with a peer MAC entity.

6.3.5.2 MLME-AUTHENTICATE.request

6.3.5.2.1 Function

This primitive requests authentication with a specified peer MAC entity.

6.3.5.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
  MLME-AUTHENTICATE.request(
                                    PeerSTAAddress,
                                    AuthenticationType,
                                    AuthenticateFailureTimeout,
```

Content of FT Authentication elements,
Content of SAE Authentication Frame,
VendorSpecificInfo
)

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity with which to perform the authentication process.
AuthenticationType	Enumeration	OPEN_SYSTEM, SHARED_KEY, FAST_BSS_TRANSITION, SAE	Specifies the type of authentication algorithm to use during the authentication process.
AuthenticationFailureTimeout	Integer	≥ 1	Specifies a time limit (in TU) after which the authentication procedure is terminated.
Content of FT Authentication elements	Sequence of elements	As defined in 12.8	The set of elements to be included in the first message of the FT authentication sequence, as described in 12.8.2. Present only if dot11FastBSSTransitionActivated is true.
Content of SAE Authentication Frame	Sequence of elements and fields	As defined in 8.4.1.37, 8.4.1.38, 8.4.1.39, 8.4.1.40, 8.4.1.41, and 8.4.1.42	The set of elements and fields to be included in the SAE Commit Message or SAE Confirm Message. Present only if AuthenticationType indicates SAE authentication.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.5.2.3 When generated

This primitive is generated by the SME for a STA to establish authentication with a specified peer MAC entity in order to permit Class 2 frames, or Mesh Peering Management frames for AMPE utilizing SAE authentication (when dot11AuthenticationAlgorithm is simultaneousAuthEquals (4)), to be exchanged between the two STAs. During the authentication procedure, the SME might generate additional MLME-AUTHENTICATE.request primitives.

6.3.5.2.4 Effect of receipt

This primitive initiates an authentication procedure. The MLME subsequently issues a MLME-AUTHENTICATE.confirm primitive that reflects the results.

6.3.5.3 MLME-AUTHENTICATE.confirm

6.3.5.3.1 Function

This primitive reports the results of an authentication attempt with a specified peer MAC entity.

6.3.5.3.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-AUTHENTICATE.confirm(
PeerSTAAddress,
AuthenticationType,
ResultCode,

Content of FT Authentication elements,
Content of SAE Authentication Frame,
VendorSpecificInfo
)

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity with which the authentication process was attempted. This value matches the peerSTAAddress parameter specified in the corresponding MLME-AUTHENTICATE.request primitive.
AuthenticationType	Enumeration	OPEN_SYSTEM, SHARED_KEY FAST_BSS_TRANSITION, SAE	Specifies the type of authentication algorithm that was used during the authentication process. This value matches the authenticationType parameter specified in the corresponding MLME-AUTHENTICATE.request primitive.
ResultCode	Enumeration	SUCCESS, REFUSED, ANTI-CLOGGING TOKEN REQUIRED, FINITE CYCLIC GROUP NOT SUPPORTED, AUTHENTICATION REJECTED	Indicates the result of the MLME-AUTHENTICATE.request primitive.
Content of FT Authentication elements	Sequence of elements	As defined in 12.8	The set of elements included in the second message of the FT authentication sequence, as described in 12.8.3. Present only if dot11FastBSSTransitionActivated is true.
Content of SAE Authentication Frame	Sequence of elements	As defined in 8.4.1.37, 8.4.1.38, 8.4.1.39, 8.4.1.40, 8.4.1.41, and 8.4.1.42	The set of elements to be included in the SAE Commit Message or SAE Confirm Message. Present only if AuthenticationType indicates SAE authentication.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.5.3.3 When generated

This primitive is generated by the MLME as a result of an MLME-AUTHENTICATE.request primitive to authenticate with a specified peer MAC entity.

6.3.5.3.4 Effect of receipt

The SME is notified of the results of the authentication procedure.

6.3.5.4 MLME-AUTHENTICATE.indication

6.3.5.4.1 Function

This primitive indicates receipt of a request from a specific peer MAC entity to establish an authentication relationship with the STA processing this primitive.

6.3.5.4.2 Semantics of the service primitive

The primitive parameters are as follows:

```
   MLME-AUTHENTICATE.indication(
                              PeerSTAAddress,
                              AuthenticationType,
                              Content of FT Authentication elements,
                              Content of SAE Authentication Frame,
                              VendorSpecificInfo
                              )
```

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity with which the authentication relationship was established.
AuthenticationType	Enumeration	OPEN_SYSTEM, SHARED_KEY, FAST_BSS_ TRANSITION, SAE	Specifies the type of authentication algorithm that was used during the authentication process.
Content of FT Authentication elements	Sequence of elements	As defined in 12.8	The set of elements included in the first message of the FT authentication sequence, as described in 12.8.2. Present only if dot11FastBSSTransitionActivated is true.
Content of SAE Authentication Frame	Sequence of elements	As defined in 8.4.1.37, 8.4.1.38, 8.4.1.39, 8.4.1.40, 8.4.1.41, and 8.4.1.42	The set of elements to be included in the SAE Commit Message or SAE Confirm Message. Present only if AuthenticationType indicates SAE authentication.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.5.4.3 When generated

This primitive is generated by the MLME as a result of the receipt of an authentication request from a specific peer MAC entity.

6.3.5.4.4 Effect of receipt

The SME is notified of the receipt of the authentication request.

6.3.5.5 MLME-AUTHENTICATE.response

6.3.5.5.1 Function

This primitive is used to send a response to a specific peer MAC entity that requested authentication with the STA that issued this primitive.

6.3.5.5.2 Semantics of the service primitive

The primitive parameters are as follows:

```
   MLME-AUTHENTICATE.response(
                                 PeerSTAAddress,
                                 ResultCode,
```

Content of FT Authentication elements,
Content of SAE Authentication Frame,
VendorSpecificInfo
)

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity from which the authentication request was received.
ResultCode	Enumeration	SUCCESS, REFUSED, ANTI-CLOGGING TOKEN REQUIRED, FINITE CYCLIC GROUP NOT SUPPORTED, AUTHENTICATION REJECTED	Indicates the result response to the authentication request from the peer MAC entity.
Content of FT Authentication elements	Sequence of elements	As defined in 12.8	The set of elements to be included in the second message of the FT authentication sequence, as described in 12.8.3. Present only if dot11FastBSSTransitionActivated is true.
Content of SAE Authentication Frame	Sequence of elements	As defined in 8.4.1.37, 8.4.1.38, 8.4.1.39, 8.4.1.40, 8.4.1.41, and 8.4.1.42	The set of elements to be included in the SAE Commit Message or SAE Confirm Message. Present only if the AuthenticationType of the MLME-AUTHENTICATE.indication primitive that generated this response indicated SAE authentication.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.5.5.3 When generated

This primitive is generated by the SME of a STA as a response to an MLME-AUTHENTICATE.indication primitive.

6.3.5.5.4 Effect of receipt

This primitive initiates transmission of a response to the specific peer MAC entity that requested authentication.

6.3.6 Deauthenticate

6.3.6.1 Introduction

This mechanism supports the process of invalidating an authentication relationship with a peer MAC entity.

6.3.6.2 MLME-DEAUTHENTICATE.request

6.3.6.2.1 Function

This primitive requests that the authentication relationship with a specified peer MAC entity be invalidated.

6.3.6.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-DEAUTHENTICATE.request(
                                        PeerSTAAddress,
                                        ReasonCode,
                                        VendorSpecificInfo
                                        )
```

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity with which to perform the deauthentication process.
ReasonCode	Reason Code field	As defined in 8.4.1.7	Specifies the reason for initiating the deauthentication procedure.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.6.2.3 When generated

This primitive is generated by the SME for a STA to invalidate authentication with a specified peer MAC entity in order to prevent the exchange of Class 2 frames, or Mesh Peering Management frames for AMPE utilizing SAE authentication (when dot11AuthenticationAlgorithm is simultaneousAuthEquals (4)), between the two STAs. During the deauthentication procedure, the SME might generate additional MLME-DEAUTHENTICATE.request primitives.

6.3.6.2.4 Effect of receipt

This primitive initiates a deauthentication procedure. The MLME subsequently issues a MLME-DEAUTHENTICATE.confirm primitive that reflects the results.

6.3.6.3 MLME-DEAUTHENTICATE.confirm

6.3.6.3.1 Function

This primitive reports the results of a deauthentication attempt with a specified peer MAC entity.

6.3.6.3.2 Semantics of the service primitive

The primitive parameter is as follows:

```
MLME-DEAUTHENTICATE.confirm(
                                        PeerSTAAddress
                                        )
```

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity with which the deauthentication process was attempted.

6.3.6.3.3 When generated

This primitive is generated by the MLME as a result of an MLME-DEAUTHENTICATE.request primitive to invalidate the authentication relationship with a specified peer MAC entity.

6.3.6.3.4 Effect of receipt

The SME is notified of the results of the deauthentication procedure.

6.3.6.4 MLME-DEAUTHENTICATE.indication

6.3.6.4.1 Function

This primitive reports the invalidation of an authentication relationship with a specific peer MAC entity.

6.3.6.4.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-DEAUTHENTICATE.indication(
                                    PeerSTAAddress,
                                    ReasonCode,
                                    VendorSpecificInfo
                                    )
```

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity with which the authentication relationship was invalidated.
ReasonCode	Reason Code field	As defined in 8.4.1.7	Specifies the reason the deauthentication procedure was initiated.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.6.4.3 When generated

This primitive is generated by the MLME as a result of the invalidation of an authentication relationship with a specific peer MAC entity.

6.3.6.4.4 Effect of receipt

The SME is notified of the invalidation of the specific authentication relationship.

6.3.7 Associate

6.3.7.1 Introduction

The following primitives describe how a STA becomes associated with an AP.

6.3.7.2 MLME-ASSOCIATE.request

6.3.7.2.1 Function

This primitive requests association with a specified peer MAC entity that is within an AP.

6.3.7.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-ASSOCIATE.request(
                                PeerSTAAddress,
```

AssociateFailureTimeout,
CapabilityInformation,
ListenInterval,
Supported Channels,
RSN,
QoSCapability,
Content of FT Authentication elements,
SupportedOperatingClasses,
HT Capabilities,
Extended Capabilities,
20/40 BSS Coexistence,
QoSTrafficCapability,
TIMBroadcastRequest,
EmergencyServices,
VendorSpecificInfo
)

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity with which to perform the association process.
AssociateFailureTimeout	Integer	≥ 1	Specifies a time limit (in TU) by dot11AssociationResponseTimeOut, after which the associate procedure is terminated.
CapabilityInformation	Capability Information field	As defined in 8.4.1.4	Specifies the requested operational capabilities to the AP.
ListenInterval	Integer	≥ 0	Specifies how often the STA awakens and listens for the next Beacon frame, if it enters power save mode.
Supported Channels	As defined in the Supported Channels element	As defined in the Supported Channels element	The list of channels in which the STA is capable of operating. Present only if DFS functionality is required, as specified in 10.9.
RSN	RSNE	As defined in 8.4.2.27	A description of the cipher suites and AKM suites selected by the STA.
QoSCapability	QoS Capability element	As defined in 8.4.2.37	Specifies the parameters within the QoS Capability element that are supported by the MAC entity. The parameter is present only if dot11QosOptionImplemented is true.
Content of FT Authentication elements	Sequence of elements	As defined in 12.4	The set of elements to be included in the initial mobility domain association request, as described in 12.4. Present only if dot11FastBSSTransitionActivated is true.
SupportedOperating Classes	As defined in the Supported Operating Classes element	As defined in 8.4.2.56	Specifies the supported operating classes capabilities of the STA. This parameter is present if dot11ExtendedChannelSwitchActivated is true.
HT Capabilities	As defined in frame format	As defined in 8.4.2.58	Specifies the parameters within the HT Capabilities element that are supported by the MAC entity. The parameter is present if dot11HighThroughputOption-Implemented is true and is absent otherwise.

Name	Type	Valid range	Description
Extended Capabilities	As defined in frame format	As defined in 8.4.2.29	Specifies the parameters within the Extended Capabilities element that are supported by the MAC entity.
20/40 BSS Coexistence	As defined in frame format	As defined in 8.4.2.62	Specifies the parameters within the 20/40 BSS Coexistence element that are indicated by the MAC entity. The parameter is present if dot112040BSSCoexistence-ManagementSupport is true.
QoS Traffic Capability	As defined in the QoS Traffic Capability element	As defined in 8.4.2.80	Specifies the QoS Traffic Capability flags of the non-AP STA. This parameter is optionally present if dot11MgmtOptionACStationCountActivated is true, and is not present otherwise.
TIMBroadcastRequest	As defined in the TIM Broadcast Request element	As defined in 8.4.2.85	Specifies the proposed service parameters for TIM Broadcast. This parameter is optionally present if dot11MgmtOptionTIMBroadcastActivated is true, and is not present otherwise.
EmergencyServices	Boolean	True, False	Specifies that the non-AP STA intends to associate for the purpose of unauthenticated access to emergency services. The parameter shall only be present if dot11InterworkingServiceActivated is true.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

Additional parameters needed to perform the association procedure are not included in the primitive parameter list since the MLME already has that data (maintained as internal state).

6.3.7.2.3 When generated

This primitive is generated by the SME when a STA wishes to establish association with an AP.

6.3.7.2.4 Effect of receipt

This primitive initiates an association procedure. The MLME subsequently issues an MLME-ASSOCIATE.confirm primitive that reflects the results.

6.3.7.3 MLME-ASSOCIATE.confirm

6.3.7.3.1 Function

This primitive reports the results of an association attempt with a specified peer MAC entity that is within an AP.

6.3.7.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-ASSOCIATE.confirm(
                                ResultCode,
                                CapabilityInformation,
                                AssociationID,
```

SupportedRates,
EDCAParameterSet,
RCPI.request,
RSNI.request,
RCPI.response,
RSNI.response,
RMEnabledCapabilities,
Content of FT Authentication elements,
SupportedOperatingClasses,
HT Capabilities,
Extended Capabilities,
20/40 BSS Coexistence,
TimeoutInterval,
BSSMaxIdlePeriod,
TIMBroadcastResponse,
QosMapSet,
VendorSpecificInfo
)

Name	Type	Valid range	Description
ResultCode	Enumeration	SUCCESS, REFUSED_REASON_UNSPECIFIED, REFUSED_NOT_AUTHENTICATED, REFUSED_CAPABILITIES_MISMATCH, REFUSED_EXTERNAL_REASON, REFUSED_AP_OUT_OF_MEMORY, REFUSED_BASIC_RATES_MISMATCH, REJECTED_EMERGENCY_SERVICES_NOT_SUPPORTED, Association request rejected temporarily; try again later	Indicates the result of the MLME-ASSOCIATE.request primitive.
Capability-Information	Capability Information field	As defined in 8.4.1.4	Specifies the operational capabilities advertised by the AP.
AssociationID	Integer	1–2007 inclusive	If the association request result was SUCCESS, then AssociationID specifies the association ID value assigned by the AP.
SupportedRates	Set of integers	2–127 inclusive (for each integer in the set), bit 7 is set to 1 to indicate that a rate is a member of the BBSBasicRateSet.	The set of data rates (in units of 500 kb/s) that are supported by AP, including indication of which rates are part of the BSSBasicRateSet (according to 8.4.2.3).
EDCAParameter Set	EDCA Parameter Set element	As defined in 8.4.2.31	Specifies the EDCA parameter set that the STA should use. The parameter is present only if dot11Qos-OptionImplemented is true.
RCPI.request	Integer	As defined in 16.4.8.6, or 18.3.10.7, or 17.4.8.6	The RCPI value represents the measured RCPI of the corresponding Association Request frame. The element is optionally present only if dot11RMRCPIMeasurementActivated is true.

Name	Type	Valid range	Description
RSNI.request	Integer	As defined in 8.4.2.43	RSNI at the time the corresponding Association Request frame was received. The element is optionally present only if dot11RMRSNIMeasurementActivated is true.
RCPI.response	Integer	As defined in 16.4.8.6, or 18.3.10.7, or 17.4.8.6	The RCPI value represents the measured RCPI of the corresponding Association Response frame. The element is optionally present only if dot11RMRCPIMeasurementActivated is true.
RSNI.response	Integer	As defined in 8.4.2.43	RSNI at the time the corresponding Association Response frame was received. The element is optionally present only if dot11RMRSNIMeasurementActivated is true.
RMEnabledCapabilities	RM Enabled Capabilities element	As defined in 8.4.2.47	Specifies the RM enabled capabilities advertised by the AP. The element is present only if dot11RadioMeasurementActivated is true.
Content of FT Authentication elements	Sequence of elements	As defined in 12.4	The set of elements included in the initial mobility domain association response, as described in 12.4. Present only if dot11FastBSSTransitionActivated is true.
SupportedOperatingClasses	As defined in the Supported Operating Classes element	As defined in 8.4.2.56	Specifies the supported operating classes capabilities of the STA. This parameter is present only if dot11ExtendedChannelSwitchActivated is true.
HT Capabilities	As defined in frame format	As defined in 8.4.2.58	Specifies the parameters within the HT Capabilities element that are supported by the MAC entity. The parameter is optionally present if dot11HighThroughputOption-Implemented is true; this parameter is not present otherwise.
Extended Capabilities	As defined in frame format	As defined in 8.4.2.29	Specifies the parameters within the Extended Capabilities element that are supported by the MAC entity.
20/40 BSS Coexistence	As defined in frame format	As defined in 8.4.2.62	Specifies the parameters within the 20/40 BSS Coexistence element that are indicated by the MAC entity. The parameter is present if dot112040BSSCoexistence-ManagementSupport is true.

Name	Type	Valid range	Description
TimeoutInterval	Timeout Interval element, as defined in frame format	As defined in 8.4.2.51	This parameter is present when ResultCode is "Association request rejected temporarily; try again later."
BSSMaxIdlePeriod	As defined in BSS Max Idle Period element	As defined in 8.4.2.81	Indicates the BSS Max idle period parameters of the AP. This parameter is present if dot11WirelessManagementImplemented is true, and is not present otherwise.
TIMBroadcastResponse	As defined in TIM Broadcast Response element	As defined in 8.4.2.86	Specifies the service parameters for TIM Broadcast. This parameter is optionally present if dot11MgmtOptionTIMBroadcastActivated is true and the TIM Broadcast Request element is present in corresponding Association Request frame, and is not present otherwise.
QoSMapSet	As defined in frame format	As defined in 8.4.2.97	Specifies the QoS Map Set the non-AP STA should use.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.7.3.3 When generated

This primitive is generated by the MLME as a result of an MLME-ASSOCIATE.request primitive or receipt of an association response frame from the peer MAC entity to associate with a specified peer MAC entity that is within an AP.

6.3.7.3.4 Effect of receipt

The SME is notified of the results of the association procedure.

6.3.7.4 MLME-ASSOCIATE.indication

6.3.7.4.1 Function

This primitive indicates that a specific peer MAC entity is requesting association with the local MAC entity, which is within an AP.

6.3.7.4.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-ASSOCIATE.indication(
                                PeerSTAAddress,
                                CapabilityInformation,
                                ListenInterval,
                                SSID,
                                SupportedRates,
                                RSN,
                                QoSCapability,
                                RCPI,
                                RSNI,
```

RMEnabledCapabilities,
Content of FT Authentication elements,
SupportedOperatingClasses,
DSERegisteredLocation,
HT Capabilities,
Extended Capabilities,
20/40 BSS Coexistence,
QoSTrafficCapability,
TIMBroadcastRequest,
EmergencyServices,
VendorSpecificInfo
)

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity from which the association was received.
Capability-Information	Capability Information field	As defined in 8.4.1.4	Specifies the operational capability definitions provided by the peer MAC entity as part of the association request.
ListenInterval	Integer	≥ 0	Specifies the listen interval value provided by the peer MAC entity as part of the association request.
SSID	Octet string	0–32 octets	Specifies the SSID provided by the peer MAC entity as part of the association request.
SupportedRates	Set of integers	2–127 inclusive (for each integer in the set)	The set of data rates (in units of 500 kb/s) that are supported by the STA that is requesting association.
RSN	RSNE	As defined in 8.4.2.27	A description of the cipher suites and AKM suites selected by the STA.
QoSCapability	QoS Capability element	As defined in 8.4.2.37	Specifies the parameters within the QoSCapability that are supported by the peer MAC entity. The parameter is optionally present only if dot11QosOption-Implemented is true.
RCPI	Integer	As defined in 16.4.8.6, or 18.3.10.7, or 17.4.8.6	The RCPI value represents the measured RCPI of the corresponding Association Request frame. The element is optionally present only if dot11RMRCPIMeasurementActivated is true.
RSNI	Integer	As defined in 8.4.2.43	The RSNI value represents the measured RSNI at the time the corresponding Association Request frame was received. The element is optionally present only if dot11RMRSNIMeasurementActivated is true.
RMEnabledCapabilities	RM Enabled Capabilities element	As defined in 8.4.2.47	Specifies the RM enabled capabilities advertised by the AP. The element is present only if dot11RadioMeasurementActivated is true.
Content of FT Authentication elements	Sequence of elements	As defined in 12.4	The set of elements included in the initial mobility domain association, as described in 12.4. Present only if dot11FastBSSTransitionActivated is true.
SupportedOperating Classes	As defined in the Supported Operating Classes element	As defined in 8.4.2.56	Indicates the supported operating classes capabilities of the AP. This parameter is present only if dot11ExtendedChannelSwitchActivated is true.
DSERegisteredLocation	As defined in the DSE Registered Location element	As defined in 8.4.2.54	Indicates the DSE registered location including the dependent enablement identifier assigned by the enabling STA. This parameter is optionally present only if dot11LCIDSERequired is true.

Name	Type	Valid range	Description
HT Capabilities	As defined in frame format	As defined in 8.4.2.58	Specifies the parameters within the HT Capabilities element that are supported by the MAC entity. The parameter is optionally present only if dot11HighThroughputOption-Implemented is true.
Extended Capabilities	As defined in frame format	As defined in 8.4.2.29	Specifies the parameters within the Extended Capabilities element that are supported by the MAC entity.
20/40 BSS Coexistence	As defined in frame format	As defined in 8.4.2.62	Specifies the parameters within the 20/40 BSS Coexistence element that are indicated by the MAC entity. The parameter is present if dot112040BSSCoexistenceManagementSupport is true.
QoS Traffic Capability	As defined in the QoS Traffic Capability element	As defined in 8.4.2.80	Specifies the QoS Traffic Capability flags of the non-AP STA. This parameter is optionally present if dot11MgmtOptionACStationCountActivated is true, and is not present otherwise.
TIMBroadcastRequest	As defined in TIM Broadcast Request element	As defined in 8.4.2.85	Specifies the proposed service parameters for TIM Broadcast. This parameter is optionally present if dot11MgmtOptionTIMBroadcastActivated is true, and is not present otherwise.
EmergencyServices	Boolean	True, false	Specifies the setting of the UESA field received from the non-AP STA, if an Interworking element was present in the Associate Request frame. The parameter is present only if dot11InterworkingServiceActivated is true.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.7.4.3 When generated

This primitive is generated by the MLME as a result of the receipt of an association request from a specific peer MAC entity.

6.3.7.4.4 Effect of receipt

The SME is notified of the receipt of the association request.

6.3.7.5 MLME-ASSOCIATE.response

6.3.7.5.1 Function

This primitive is used to send a response to a specific peer MAC entity that requested an association with the STA that issued this primitive, which is within an AP.

6.3.7.5.2 Semantics of the service primitive

The primitive parameters are as follows:

```
   MLME-ASSOCIATE.response(
                                        PeerSTAAddress,
                                        ResultCode,
                                        CapabilityInformation,
```

AssociationID,
EDCAParameterSet,
RCPI,
RSNI,
RMEnabledCapabilities,
Content of FT Authentication elements,
SupportedOperatingClasses,
DSERegisteredLocation,
HTCapabilities,
Extended Capabilities,
20/40 BSS Coexistence,
TimeoutInterval,
BSSMaxIdlePeriod,
TIMBroadcastResponse,
QoSMapSet,
VendorSpecificInfo
)

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity from which the association request was received.
ResultCode	Enumeration	SUCCESS, REFUSED_REASON_UNSPECIFIED, REFUSED_CAPABILITIES_MISMATCH, REFUSED_EXTERNAL_REASON, REFUSED_AP_OUT_OF_MEMORY, REFUSED_BASIC_RATES_MISMATCH, REJECTED_EMERGENCY_SERVICES_NOT_SUPPORTED, REFUSED_TEMPORARILY	Indicates the result response to the association request from the peer MAC entity.
Capability-Information	Capability Information field	As defined in 8.4.1.4	Specifies the operational capabilities advertised by the AP.
AssociationID	Integer	1–2007 inclusive	If the association request result was SUCCESS, then AssociationID specifies the association ID value assigned to the peer MAC entity by the AP.
EDCAParameterSet	EDCA Parameter Set element	As defined in 8.4.2.31	Specifies the EDCA parameter set that the STA should use. The parameter is present only if dot11Qos-OptionImplemented is true.
RCPI	Integer	As defined in 16.4.8.6, or 18.3.10.7, or 17.4.8.6	The RCPI value represents the measured RCPI of the corresponding Association Request frame. The element is optionally present only if dot11RMRCPIMeasurementActivated is true.

Name	Type	Valid range	Description
RSNI	Integer	As defined in 8.4.2.43	The RSNI value represents the measured RSNI at the time the corresponding Association Request frame was received. The element is optionally present only if dot11RMRSNIMeasurementActivated is true.
RMEnabled-Capabilities	RM Enabled Capabilities element	As defined in 8.4.2.47	Specifies the RM enabled capabilities advertised by the AP. The element is present only if dot11RadioMeasurementActivated is true.
Content of FT Authentication elements	Sequence of elements	As defined in 12.4	The set of elements to be included in the initial mobility domain association response, as described in 12.4. Present only if dot11FastBSSTransitionActivated is true.
SupportedOperating Classes	As defined in the Supported Operating Classes element	As defined in 8.4.2.56	Indicates the supported operating classes capabilities of the AP. This parameter is present if dot11ExtendedChannelSwitch Activated is true.
DSERegisteredLocation	As defined in the DSE Registered Location element	As defined in 8.4.2.54	Indicates the DSE registered location including the dependent enablement identifier assigned by the enabling STA. This parameter is optionally present if dot11LCIDSERequired is true.
HT Capabilities	As defined in frame format	As defined in 8.4.2.58	Specifies the parameters within the HT Capabilities element that are supported by the MAC entity. The parameter is present if dot11HighThroughputOption-Implemented is true; otherwise it is not present.
Extended Capabilities	As defined in frame format	As defined in 8.4.2.29	Specifies the parameters within the Extended Capabilities element that are supported by the MAC entity.
20/40 BSS Coexistence	As defined in frame format	As defined in 8.4.2.62	Specifies the parameters within the 20/40 BSS Coexistence element that are indicated by the MAC entity. The parameter is present if dot112040BSSCoexistenceManagementSupport is true.
TimeoutInterval	Timeout Interval element, as defined in frame format	As defined in 8.4.2.51	This parameter is present when ResultCode is “Association request rejected temporarily; try again later.”

Name	Type	Valid range	Description
BSSMaxIdlePeriod	As defined in BSS Max Idle Period element	As defined in 8.4.2.81	Indicates the BSS Max idle period parameters of the AP. This parameter is present if dot11WirelessManagementImplemented is true, and is not present otherwise.
TIMBroadcastResponse	As defined in TIM Broadcast Response element	As defined in 8.4.2.86	Specifies the service parameters for TIM Broadcast. This parameter is optionally present if dot11MgmtOptionTIMBroadcastActivated is true and the TIM Broadcast Request element is present in corresponding Association Request frame, and is not present otherwise.
QoSMapSet	As defined in frame format	As defined in 8.4.2.97	Specifies the QoS Map Set the non-AP STA should use.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

Additional parameters needed to perform the association response procedure are not included in the primitive parameter list since the MLME already has that data (maintained as internal state).

6.3.7.5.3 When generated

This primitive is generated by the SME of a STA that is within an AP as a response to an MLME-ASSOCIATE.indication primitive.

6.3.7.5.4 Effect of receipt

This primitive initiates transmission of an AssociationResponse to the specific peer MAC entity that requested association.

6.3.8 Reassociate

6.3.8.1 Introduction

The following primitives describe how a STA becomes associated with another AP.

6.3.8.2 MLME-REASSOCIATE.request

6.3.8.2.1 Function

This primitive requests a change in association to a specified new peer MAC entity that is within an AP.

6.3.8.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
  MLME-REASSOCIATE.request(
                                    NewAPAddress,
                                    ReassociateFailureTimeout,
                                    CapabilityInformation,
                                    ListenInterval,
                                    Supported Channels
```

RSN,
QoSCapability,
Content of FT Authentication elements,
SupportedOperatingClasses,
HT Capabilities,
Extended Capabilities,
20/40 BSS Coexistence,
QoSTrafficCapability,
TIMBroadcastRequest,
FMSRequest,
DMSRequest,
EmergencyServices,
VendorSpecificInfo
)

Name	Type	Valid range	Description
NewAPAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity with which to perform the reassociation process.
ReassociateFailure Timeout	Integer	≥ 1	Specifies a time limit (in TU) from dot11AssociationResposeTimeout, after which the reassociate procedure is terminated.
Capability-Information	Capability Information field	As defined in 8.4.1.4	Specifies the requested operational capabilities to the AP.
ListenInterval	Integer	≥ 0	Specifies how often the STA awakens and listens for the next Beacon frame, if it enters power save mode.
Supported Channels	As defined in the Supported Channels element	As defined in the Supported Channels element	The list of channels in which the STA is capable of operating. Present only if DFS functionality is required, as specified in 10.9.
RSN	RSNE	As defined in 8.4.2.27	A description of the cipher suites and AKM suites selected by the STA.
QoSCapability	QoS Capability element	As defined in 8.4.2.37	Specifies the parameters within the QoS Capability element that are supported by the MAC entity. The parameter is present only if dot11QosOptionImplemented is true.
Content of FT Authentication elements	Sequence of elements	As defined in 12.8	The set of elements to be included in the third message of the FT authentication sequence, as described in 12.8.4. Present only if dot11FastBSSTransitionActivated is true.
SupportedOperating Classes	As defined in the Supported Operating Classes element	As defined in 8.4.2.56	Specifies the supported operating classes of the STA. This parameter is present if dot11ExtendedChannelSwitchActivated is true.
HT Capabilities	As defined in frame format	As defined in 8.4.2.58	Specifies the parameters within the HT Capabilities element that are supported by the MAC entity. The parameter is present if dot11HighThroughputOption-Implemented is true; otherwise it is not present.
Extended Capabilities	As defined in frame format	As defined in 8.4.2.29	Specifies the parameters within the Extended Capabilities element that are supported by the MAC entity.

Name	Type	Valid range	Description
20/40 BSS Coexistence	As defined in frame format	As defined in 8.4.2.62	Specifies the parameters within the 20/40 BSS Coexistence element that are indicated by the MAC entity. The parameter is present if dot112040BSSCoexistenceManagementSupport is true.
QoS Traffic Capability	As defined in the QoS Traffic Capability element	As defined in 8.4.2.80	Specifies the QoS Traffic Capability flags of the non-AP STA. This parameter is optionally present if dot11MgmtOptionACStationCountActivated is true, and is not present otherwise.
TIMBroadcastRequest	As defined in TIM Broadcast Request element	As defined in 8.4.2.85	Specifies the proposed service parameters for TIM Broadcast. This parameter is optionally present if dot11MgmtOptionTIMBroadcastActivated is true, and is not present otherwise.
FMSRequest	As defined in FMS Request element	As defined in 8.4.2.78	Specifies the proposed multicast parameters for FMS Request. This parameter is optionally present if dot11MgmtOptionFMSActivated is true, and is not present otherwise.
DMSRequest	As defined in DMS Request element	As defined in 8.4.2.90	Specifies the proposed multicast parameters for DMS Request. This parameter is optionally present if dot11MgmtOptionDMSActivated is true, and is not present otherwise.
EmergencyServices	Boolean	True, False	Specifies that the non-AP STA intends to associate for the purpose of unauthenticated access to emergency services. The parameter shall only be present if dot11InterworkingServiceActivated is true.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

Additional parameters needed to perform the reassociation procedure are not included in the primitive parameter list since the MLME already has that data (maintained as internal state).

6.3.8.2.3 When generated

This primitive is generated by the SME for a STA to change association to a specified new peer MAC entity that is within an AP.

6.3.8.2.4 Effect of receipt

This primitive initiates a reassociation procedure. The MLME subsequently issues a MLME-REASSOCIATE.confirm primitive that reflects the results.

6.3.8.3 MLME-REASSOCIATE.confirm

6.3.8.3.1 Function

This primitive reports the results of a reassociation attempt with a specified peer MAC entity that is within an AP.

6.3.8.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-REASSOCIATE.confirm(
                              ResultCode,
                              CapabilityInformation,
                              AssociationID,
                              SupportedRates,
                              EDCAParameterSet,
                              RCPI.request,
                              RSNI.request,
                              RCPI.response,
                              RSNI.response,
                              RMEnabledCapabilities,
                              Content of FT Authentication elements,
                              SupportedOperatingClasses,
                              HT Capabilities,
                              Extended Capabilities,
                              20/40 BSS Coexistence,
                              TimeoutInterval,
                              BSSMaxIdlePeriod,
                              TIMBroadcastResponse,
                              FMSRespone,
                              DMSResponse,
                              QoSMapSet,
                              VendorSpecificInfo
                              )
```

Name	Type	Valid range	Description
ResultCode	Enumeration	SUCCESS, REFUSED_REASON_UNSPECIFIED, REFUSED_NOT_AUTHENTICATED, REFUSED_CAPABILITIES_MISMATCH, REFUSED_EXTERNAL_REASON, REFUSED_AP_OUT_OF_MEMORY, REFUSED_BASIC_RATES_MISMATCH, REJECTED_EMERGENCY_SERVICES_NOT_SUPPORTED, Association request rejected temporarily; try again later	Indicates the result of the MLME-REASSOCIATE.request primitive.
Capability-Information	Capability Information field	As defined in 8.4.1.4	Specifies the operational capabilities advertised by the AP.
AssociationID	Integer	1–2007 inclusive	If the association request result was SUCCESS, then AssociationID specifies the association ID value assigned by the AP.
Supported-Rates	Set of integers	2–127 inclusive (for each integer in the set), bit 7 is set to 1 to indicate that a rate is a member of the BBSBasicRateSet.	The set of data rates (in units of 500 kb/s) that are supported by AP, including indication of which rates are part of the BSSBasicRateSet (according to 8.4.2.3).

Name	Type	Valid range	Description
EDCAParameterSet	EDCA Parameter Set element	As defined in 8.4.2.31	Specifies the EDCA parameter set that the STA should use. The parameter is present only if dot11QosOptionImplemented is true.
RCPI.request	Integer	As defined in 16.4.8.6, or 18.3.10.7, or 17.4.8.6	The RCPI value represents the measured RCPI of the corresponding Association Request frame. The element is optionally present only if dot11RMRCPIMeasurementActivated is true.
RSNI.request	Integer	As defined in 8.4.2.43	RSNI at the time the corresponding Association Request frame was received. The element is optionally present only if dot11RMRSNIMeasurementActivated is true.
RCPI.response	Integer	As defined in 16.4.8.6, or 18.3.10.7, or 17.4.8.6	The RCPI value represents the measured RCPI of the corresponding Association Response frame. The element is optionally present only if dot11RMRCPIMeasurementActivated is true.
RSNI.response	Integer	As defined in 8.4.2.43	RSNI at the time the corresponding Association Response frame was received. The element is optionally present only if dot11RMRSNIMeasurementActivated is true.
RMEnabledCapabilities	RM Enabled Capabilities element	As defined in 8.4.2.47	Specifies the RM enabled capabilities advertised by the AP. The element is present only if dot11RadioMeasurementActivated is true.
Content of FT Authentication elements	Sequence of elements	As defined in 12.8	The set of elements included in the fourth message of the FT authentication sequence, as described in 12.8.5. This includes an optional response to a resource request (RIC). Present only if dot11FastBSSTransitionActivated is true.
SupportedOperatingClasses	As defined in the Supported Operating Classes element	As defined in 8.4.2.56	Specifies the supported operating classes of the STA. This parameter is present only if dot11ExtendedChannelSwitchActivated is true.
HT Capabilities	As defined in frame format	As defined in 8.4.2.58	Specifies the parameters within the HT Capabilities element that are supported by the MAC entity. The parameter is optionally present only if dot11HighThroughputOption-Implemented is true.
Extended Capabilities	As defined in frame format	As defined in 8.4.2.29	Specifies the parameters within the Extended Capabilities element that are supported by the MAC entity.
20/40 BSS Coexistence	As defined in frame format	As defined in 8.4.2.62	Specifies the parameters within the 20/40 BSS Coexistence element that are indicated by the MAC entity. The parameter is present if dot112040BSSCoexistenceManagement Support is true.

Name	Type	Valid range	Description
TimeoutInterval	Timeout Interval element, as defined in frame format	As defined in 8.4.2.51	This parameter is present when ResultCode is “Association request rejected temporarily; try again later.”
BSSMaxIdlePeriod	As defined in BSS Max Idle Period element	As defined in 8.4.2.81	Indicates the BSS Max idle period parameters of the AP. This parameter is present if dot11WirelessManagementImplemented is true, and is not present otherwise.
TIMBroadcastResponse	As defined in TIM Broadcast Response element	As defined in 8.4.2.86	Specifies the service parameters for TIM Broadcast. This parameter is optionally present if dot11MgmtOptionTIMBroadcastActivated is true and the TIM Broadcast Request element is present in corresponding Association Request frame, and is not present otherwise.
FMSResponse	As defined in FMS Response element	As defined in 8.4.2.79	Specifies the multicast parameters for FMS REsponse. This parameter is optionally present if dot11MgmtOptionFMSActivated is true and the FMS Request element is present in corresponding Association Request frame, and is not present otherwise.
DMSResponse	As defined in DMS Response element	As defined in 8.4.2.91	Specifies the multicast parameters for DMS Response. This parameter is optionally present if dot11MgmtOptionDMSActivated is true and the DMS Request element is present in corresponding Association Request frame, and is not present otherwise.
QoSMapSet	As defined in frame format	As defined in 8.4.2.97	Specifies the QoS Map Set the non-AP STA should use.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.8.3.3 When generated

This primitive is generated by the MLME as a result of an MLME-REASSOCIATE.request primitive to reassociate with a specified peer MAC entity that is within an AP.

6.3.8.3.4 Effect of receipt

The SME is notified of the results of the reassociation procedure.

6.3.8.4 MLME-REASSOCIATE.indication

6.3.8.4.1 Function

This primitive indicates that a specific peer MAC entity is requesting reassociation with the local MAC entity, which is within an AP.

6.3.8.4.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-REASSOCIATE.indication(

PeerSTAAddress,
CurrentAPAddress,
CapabilityInformation,
ListenInterval,
SSID,
SupportedRates,
RSN,
QoSCapability,
RCPI,
RSNI,
RMEnabledCapabilities,
Content of FT Authentication elements,
SupportedOperatingClasses,
DSERegisteredLocation,
HT Capabilities,
Extended Capabilities,
20/40 BSS Coexistence,
QoSTrafficCapability,
TIMBroadcastRequest,
FMSRequest,
DMSRequest,
EmergencyServices,
VendorSpecificInfo
)

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity from which the reassociation request was received.
CurrentAPAddress	MACAddress	Any valid individual MAC address	Specifies the address of the AP with which the peer STA is currently associated.
Capability-Information	Capability Information field	As defined in 8.4.1.4	Specifies the operational capability definitions provided by the peer MAC entity as part of the reassociation request.
ListenInterval	Integer	≥ 0	Specifies the listen interval value provided by the peer MAC entity as part of the reassociation request.
SSID	Octet string	0–32 octets	Specifies the desired SSID provided by the peer MAC entity as part of the reassociation request.
SupportedRates	Set of integers	2–127 inclusive (for each integer in the set)	The set of data rates (in units of 500 kb/s) that are supported by the STA that is requesting reassociation.
RSN	RSNE	As defined in 8.4.2.27	A description of the cipher suites and AKM suites selected by the STA.
QoSCapability	QoS Capability element	As defined in 8.4.2.37	Specifies the parameters within the QoS Capability that are supported by the peer MAC entity. The parameter is present only if dot11QosOption-Implemented is true.
RCPI	Integer	As defined in 16.4.8.6, or 18.3.10.7, or 17.4.8.6	The RCPI value represents the measured RCPI of the corresponding Reassociation Request frame. The element is optionally present only if dot11RMRCPIMeasurementActivated is true.

Name	Type	Valid range	Description
RSNI	Integer	As defined in 8.4.2.43	The RSNI value represents the measured RSNI at the time the corresponding Reassociation Request frame was received. The element is optionally present only if dot11RMRSNIMeasurementActivated is true.
RMEnabled-Capabilities	RM Enabled Capabilities element	As defined in 8.4.2.47	Specifies the RM enabled capabilities advertised by the AP. The element is present only if dot11RadioMeasurementActivated is true.
Content of FT Authentication elements	Sequence of elements	As defined in 12.8	The set of elements included in the third message of the FT authentication sequence, as described in 12.8.4. Present only if dot11FastBSSTransitionActivated is true.
SupportedOperatingClasses	As defined in the Supported Operating Classes element	As defined in 8.4.2.56	Specifies the supported operating classes of the STA. This parameter is present only if dot11ExtendedChannelSwitchActivated is true.
DSERegisteredLocation	As defined in the DSE Registered Location element	As defined in 8.4.2.54	Indicates the DSE registered location including the dependent enablement identifier assigned by the enabling STA. This parameter is optionally present only if dot11LCIDSERequired is true.
HT Capabilities	As defined in frame format	As defined in 8.4.2.58	Specifies the parameters within the HT Capabilities element that are supported by the MAC entity. The parameter is optionally present if dot11HighThroughputOptionImplemented is true.
Extended Capabilities	As defined in frame format	As defined in 8.4.2.29	Specifies the parameters within the Extended Capabilities element that are supported by the MAC entity.
20/40 BSS Coexistence	As defined in frame format	As defined in 8.4.2.62	Specifies the parameters within the 20/40 BSS Coexistence element that are indicated by the MAC entity. The parameter is present if dot112040BSSCoexistenceManagementSupport is true.
QoS Traffic Capability	As defined in the QoS Traffic Capability element	As defined in 8.4.2.80	Specifies the QoS Traffic Capability flags of the non-AP STA. This parameter is optionally present if dot11MgmtOptionACStationCountActivated is true, and is not present otherwise.
TIMBroadcastRequest	As defined in TIM Broadcast Request element	As defined in 8.4.2.85	Specifies the proposed service parameters for TIM Broadcast. This parameter is optionally present if dot11MgmtOptionTIMBroadcastActivated is true, and is not present otherwise.
FMSRequest	As defined in FMS Request element	As defined in 8.4.2.78	Specifies the proposed multicast parameters for FMS Request. This parameter is optionally present if dot11MgmtOptionFMSActivated is true, and is not present otherwise.
DMSRequest	As defined in DMS Request element	As defined in 8.4.2.90	Specifies the proposed multicast parameters for DMS Request. This parameter is optionally present if dot11MgmtOptionDMSActivated is true, and is not present otherwise.

Name	Type	Valid range	Description
EmergencyServices	Boolean	True, false	Specifies the setting of the UESA field received from the non-AP STA, if an Interworking element was present in the Reassociate Request frame. The parameter is present only if dot11InterworkingServiceActivated is true.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.8.4.3 When generated

This primitive is generated by the MLME as a result of the establishment of a reassociation with a specific peer MAC entity that resulted from a reassociation procedure that was initiated by that specific peer MAC entity.

6.3.8.4.4 Effect of receipt

The SME is notified of the establishment of the reassociation.

6.3.8.5 MLME-REASSOCIATE.response

6.3.8.5.1 Function

This primitive is used to send a response to a specific peer MAC entity that requested a reassociation with the STA that issued this primitive, which is within an AP.

6.3.8.5.2 Semantics of the service primitive

The primitive parameters are as follows:

```
  MLME-REASSOCIATE.response(
                                    PeerSTAAddress,
                                    ResultCode,
                                    CapabilityInformation,
                                    AssociationID,
                                    EDCAParameterSet,
                                    RCPI,
                                    RSNI,
                                    RMEnabledCapabilities,
                                    Content of FT Authentication elements,
                                    SupportedOperatingClasses,
                                    DSERegisteredLocation,
                                    HT Capabilities,
                                    Extended Capabilities,
                                    20/40 BSS Coexistence,
                                    TimeoutInterval,
                                    BSSMaxIdlePeriod,
                                    TIMBroadcastResponse,
                                    FMSResponse,
                                    DMSResponse,
                                    QoSMapSet,
```

VendorSpecificInfo
)

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity from which the reassociation request was received.
ResultCode	Enumeration	SUCCESS, REFUSED_REASON_UNSPECIFIED, REFUSED_CAPABILITIES_MISMATCH, REFUSED_EXTERNAL_REASON, REFUSED_AP_OUT_OF_MEMORY, REFUSED_BASIC_RATES_MISMATCH REJECTED_EMERGENCY_SERVICES_NOT_SUPPORTED, Association request rejected temporarily; try again later	Indicates the result response to the reassociation request from the peer MAC entity.
Capability-Information	Capability Information field	As defined in 8.4.1.4	Specifies the operational capabilities advertised by the AP.
AssociationID	Integer	1–2007 inclusive	If the reassociation request result was SUCCESS, then AssociationID specifies the association ID value assigned to the peer MAC entity by the AP.
EDCAParameterSet	EDCA Parameter Set element	As defined in 8.4.2.31	Specifies the EDCA parameter set that the STA should use. The parameter is present only if dot11Qos-OptionImplemented is true.
RCPI	Integer	As defined in 16.4.8.6, or 18.3.10.7, or 17.4.8.6	The RCPI value represents the measured RCPI of the corresponding reassociation request frame. The element is optionally present only if dot11RMRCPIMeasurementActivated is true.
RSNI	Integer	As defined in 8.4.2.43	The RSNI value represents the measured RSNI at the time the corresponding reassociation request frame was received. The element is optionally present only if dot11RMRSNIMeasurementActivated is true.
RMEnabled-Capabilities	RM Enabled Capabilities element	As defined in 8.4.2.47	Specifies the RM enabled capabilities advertised by the AP. The element is present only if dot11RadioMeasurementActivated is true.

Name	Type	Valid range	Description
Content of FT Authentication elements	Sequence of elements	As defined in 12.8	The set of elements to be included in the fourth message of the FT authentication sequence, as described in 12.8.5. This includes an optional response to a resource request (RIC). Present only if dot11FastBSSTransitionActivated is true.
SupportedOperatingClasses	As defined in the Supported Operating Classes element	As defined in 8.4.2.56	Specifies the supported operating classes of the STA. This parameter is present if dot11ExtendedChannelSwitchActivated is true.
DSERegisteredLocation	As defined in the DSE Registered Location element	As defined in 8.4.2.54	Indicates the DSE registered location including the dependent enablement identifier assigned by the enabling STA. This parameter is optionally present if dot11LCIDSERequired is true.
HT Capabilities	As defined in frame format	As defined in 8.4.2.58	Specifies the parameters within the HT Capabilities element that are supported by the MAC entity. The parameter is present if dot11HighThroughputOption-Implemented is true; otherwise it is not present.
Extended Capabilities	As defined in frame format	As defined in 8.4.2.29	Specifies the parameters within the Extended Capabilities element that are supported by the MAC entity.
20/40 BSS Coexistence	As defined in frame format	As defined in 8.4.2.62	Specifies the parameters within the 20/40 BSS Coexistence element that are indicated by the MAC entity. The parameter is present if dot112040BSSCoexistenceManagementSupport is true.
TimeoutInterval	Timeout Interval element, as defined in frame format	As defined in 8.4.2.51	This parameter is present when ResultCode is “Association request rejected temporarily; try again later.”
BSSMaxIdlePeriod	As defined in BSS Max Idle Period element	As defined in 8.4.2.81	Indicates the BSS Max idle period parameters of the AP. This parameter is present if dot11WirelessManagementImplemented is true, and is not present otherwise.

Name	Type	Valid range	Description
TIMBroadcastResponse	As defined in TIM Broadcast Response element	As defined in 8.4.2.86	Specifies the service parameters for TIM Broadcast. This parameter is optionally present if dot11MgmtOptionTIMBroadcastActivated is true and the TIM Broadcast Request element is present in corresponding Association Request frame, and is not present otherwise.
FMSResponse	As defined in FMS Response element	As defined in 8.4.2.79	Specifies the multicast parameters for FMS Response. This parameter is optionally present if dot11MgmtOptionFMSActivated is true and the FMS Request element is present in corresponding Association Request frame, and is not present otherwise.
DMSResponse	As defined in DMS Response element	As defined in 8.4.2.91	Specifies the multicast parameters for DMS Response. This parameter is optionally present if dot11MgmtOptionDMSActivated is true and the DMS Request element is present in corresponding Association Request frame, and is not present otherwise.
QoSMapSet	As defined in frame format	As defined in 8.4.2.97	Specifies the QoS Map Set the non-AP STA should use.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

Additional parameters needed to perform the association response procedure are not included in the primitive parameter list since the MLME already has that data (maintained as internal state).

6.3.8.5.3 When generated

This primitive is generated by the SME of a STA that is within an AP as a response to an MLME-REASSOCIATE.indication primitive.

6.3.8.5.4 Effect of receipt

This primitive initiates transmission of a response to the specific peer MAC entity that requested reassociation.

6.3.9 Disassociate

6.3.9.1 MLME-DISASSOCIATE.request

6.3.9.1.1 Function

This primitive requests disassociation with a specified peer MAC entity.

6.3.9.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-DISASSOCIATE.request(
                              PeerSTAAddress,
                              ReasonCode,
                              VendorSpecificInfo
                              )
```

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity with which to perform the disassociation process.
ReasonCode	Reason Code field	As defined in 8.4.1.7	Specifies the reason for initiating the disassociation procedure.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.9.1.3 When generated

This primitive is generated by the SME for a STA to disassociate from a STA with which it has an association.

6.3.9.1.4 Effect of receipt

This primitive initiates a disassociation procedure. The MLME subsequently issues an MLME-DISASSOCIATE.confirm primitive that reflects the results.

6.3.9.2 MLME-DISASSOCIATE.confirm

6.3.9.2.1 Function

This primitive reports the results of a disassociation procedure with a specific peer MAC entity.

6.3.9.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-DISASSOCIATE.confirm(
                              ResultCode,
                              VendorSpecificInfo
                              )
```

Name	Type	Valid range	Description
ResultCode	Enumeration	SUCCESS, INVALID_PARAMETERS	Indicates the result of the MLME-DISASSOCIATE.request primitive.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.9.2.3 When generated

This primitive is generated by the MLME as a result of an MLME-DISASSOCIATE.request primitive to disassociate with a specified peer MAC entity.

6.3.9.2.4 Effect of receipt

The SME is notified of the results of the disassociation procedure.

6.3.9.3 MLME-DISASSOCIATE.indication

6.3.9.3.1 Function

This primitive reports disassociation with a specific peer MAC entity.

6.3.9.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-DISASSOCIATE.indication(
                                        PeerSTAAddress,
                                        ReasonCode,
                                        VendorSpecificInfo
                                        )
```

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity with which the association relationship was invalidated.
ReasonCode	Reason Code field	As defined in 8.4.1.7	Specifies the reason the disassociation procedure was initiated.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.9.3.3 When generated

This primitive is generated by the MLME as a result of the invalidation of an association relationship with a specific peer MAC entity.

6.3.9.3.4 Effect of receipt

The SME is notified of the invalidation of the specific association relationship.

6.3.10 Reset

6.3.10.1 Introduction

This mechanism supports the process of resetting the MAC.

6.3.10.2 MLME-RESET.request

6.3.10.2.1 Function

This primitive requests that the MAC entity be reset.

6.3.10.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-RESET.request(
                                        STAAddress,
```

SetDefaultMIB
)

Name	Type	Valid range	Description
STAAddress	MACAddress	Any valid individual MAC address	Specifies the MAC address that is to be used by the MAC entity that is being reset. This value can be used to provide a locally administered STA address.
SetDefaultMIB	Boolean	true, false	If true, all MIB attributes are set to their default values. The default values are implementation dependent. If false, the MAC is reset, but all MIB attributes retain the values that were in place prior to the generation of the MLME-RESET.request primitive.

6.3.10.2.3 When generated

This primitive is generated by the SME to reset the MAC to initial conditions. The MLME-RESET.request primitive shall be used prior to use of the MLME-START.request primitive.

6.3.10.2.4 Effect of receipt

This primitive sets the MAC to initial conditions, clearing all internal variables to the default values. MIB attributes can be reset to their implementation-dependent default values by setting the SetDefaultMIB flag to true.

If dot11OCBActivated is true and if the SetDefaultMIB parameter is false, MAC operation shall resume in less than 2 TU after the STAAddress parameter is changed.

6.3.11 Start

6.3.11.1 Introduction

This mechanism supports the process of creating a new BSS or becoming a member of an MBSS.

6.3.11.2 MLME-START.request

6.3.11.2.1 Function

This primitive requests that the MAC entity start a new BSS or become a member of an MBSS.

6.3.11.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-START.request(
                    SSID,
                    SSIDEncoding,
                    BSSType,
                    BeaconPeriod,
                    DTIMPeriod,
                    CF parameter set,
                    PHY parameter set,
                    IBSS parameter set,
                    ProbeDelay,
                    CapabilityInformation,
```

BSSBasicRateSet,
OperationalRateSet,
Country,
IBSS DFS Recovery Interval,
EDCAParameterSet,
DSERegisteredLocation,
HT Capabilities,
HT Operation,
BSSMembershipSelectorSet,
BSSBasicMCSSet,
HTOperationalMCSSet,
Extended Capabilities,
20/40 BSS Coexistence,
Overlapping BSS Scan Parameters,
MultipleBSSID,
InterworkingInfo,
AdvertismentProtocolInfo,
RoamingConsortiumInfo,
Mesh ID,
Mesh Configuration,
VendorSpecificInfo
)

Name	Type	Valid range	Description
SSID	Octet string	0–32 octets	The SSID of the BSS.
SSIDEncoding	Enumeration	UNSPECIFIED, UTF8	The encoding used for the SSID
BSSType	Enumeration	INFRASTRUCTURE, INDEPENDENT, MESH	The type of the BSS.
Beacon Period	Integer	≥ 1	The Beacon period (in TU) of the BSS if the BSSType is not MESH, or of the mesh STA if the BSSType is MESH.
DTIM Period	Integer	As defined in 8.4.2.7	The DTIM Period (in beacon periods) of the BSS if the BSSType is not MESH, or of the mesh STA if the BSSType is MESH.
CF parameter set	CF Parameter Set element	As defined in 8.4.2.6	The parameter set for CF periods, if the BSS supports CF mode.
PHY parameter sets	FH Parameter Set element or DSSS Parameter Set element	As defined in 8.4.2.4 or 8.4.2.5	The parameter sets relevant to the PHY.
IBSS parameter set	IBSS Parameter Set element	As defined in 8.4.2.8	The parameter set for the IBSS, if BSS is an IBSS.
ProbeDelay	Integer	N/A	Delay (in microseconds) to be used, while the STA is a member of this BSS, prior to transmitting when changing from Doze to Awake, if no frame sequence is detected by which the NAV can be set.
CapabilityInformation	Capability Information field	As defined in 8.4.1.4	The capabilities to be advertised for the BSS.

Name	Type	Valid range	Description
BSSBasicRateSet	Set of integers	1–127 inclusive (for each integer in the set)	The set of data rates that shall be supported by all STAs to join this BSS. The STA that is creating the BSS shall be able to receive and transmit at each of the data rates listed in the set.
OperationalRateSet	Set of integers	1–127 inclusive (for each integer in the set)	The set of data rates that the STA desires to use for communication within the BSS. The STA shall be able to receive at each of the data rates listed in the set. This set is a superset of the rates contained in the BSSBasicRateSet parameter.
Country	As defined in the Country element	As defined in the Country element	The information required to identify the regulatory domain in which the STA is located and to configure its PHY for operation in that regulatory domain. Present only if TPC functionality is required, as specified in 10.8 or dot11MultiDomainCapabilityActivated is true.
IBSS DFS Recovery Interval	Integer	1–255	Present only if BSSType = INDEPENDENT. The time interval that is used for DFS recovery. Present only if DFS functionality is required, as specified in 10.9.
EDCAParameterSet (QoS only)	EDCA Parameter Set element	As defined in 8.4.2.31	The initial EDCA parameter set values to be used in the BSS. The parameter is present only if dot11QosOptionImplemented is true.
DSERegisteredLocation	As defined in the DSE Registered Location element	As defined in 8.4.2.54	The information for the DSE Registered Location element. The parameter is present if dot11LCIDSERequired is true.
HT Capabilities	As defined in frame format HT Capabilities element	As defined in 8.4.2.58	The HT capabilities to be advertised for the BSS. The parameter is present if dot11HighThroughputOptionImplemented is true; otherwise, this parameter is not present.
HT Operation	As defined in frame format HT Operation element	As defined in 8.4.2.59	The additional HT capabilities to be advertised for the BSS. The parameter is present if BSSType = INFRASTRUCTURE and dot11HighThroughputOptionImplemented is true; otherwise, this parameter is not present.
BSSMembership-SelectorSet	Set of integers	A value from Table 8-55 for each member of the set	The BSS membership selectors that represent the set of features that shall be supported by all STAs to join this BSS. The STA that is creating the BSS shall be able to support each of the features represented by the set.
BSSBasicMCSSet	Set of integers	Each member of the set takes a value in the range 0 to 76, representing an MCS index value	The set of MCS values that shall be supported by all HT STAs to join this BSS. The STA that is creating the BSS shall be able to receive and transmit at each of the MCS values listed in the set. If the HT Operation parameter includes a value of 1 for either the Dual Beacon field or the Dual CTS Protection field, the BSSBasicMCSSet parameter shall include at least one integer value in the range 0 to 7.

Name	Type	Valid range	Description
HTOperationalMCSSet	Set of integers	Each member of the set takes a value in the range 0 to 76, representing an MCS index value	The set of MCS values that the STA desires to use for communication within the BSS. The STA shall be able to receive at each of the data rates listed in the set. This set is a superset of the MCS values contained in the BSSBasicMCSSet parameter.
Extended Capabilities	As defined in frame format	As defined in 8.4.2.29	Specifies the parameters within the Extended Capabilities element that are supported by the MAC entity.
20/40 BSS Coexistence	As defined in frame format	As defined in 8.4.2.62	Specifies the parameters within the 20/40 BSS Coexistence element that are indicated by the MAC entity. The parameter is present if dot112040BSSCoexistence-ManagementSupport is true.
Overlapping BSS Scan Parameters	As defined in frame format	As defined in 8.4.2.61	Specifies the parameters within the Overlapping BSS Scan Parameters element that are indicated by the MAC entity. This parameter is optionally present if dot11FortyMHzOptionImplemented is true; otherwise, it is not present.
MultipleBSSID	As defined in Multiple BSSID Element in 8.4.2.48	As defined in Multiple BSSID Element in 8.4.2.48	This element is optionally present when dot11RMMeasurementPilotCapability is a value between 2 and 7 and the AP is a member of a Multiple BSSID Set (see 10.11.14) with two or more members, or if dot11MgmtOptionMultiBSSIDActivated is true.
InterworkingInfo	As defined in frame format	As defined in Interworking element in 8.4.2.94	Specifies the Interworking capabilities of STA. This field is present when dot11InterworkingServiceActivated is true.
AdvertisementProtocolInfo	Integer or Sequence of Integers	As defined in Advertisement Protocol element in Table 8-175	Identifies zero or more Advertisement Protocols and advertisement control to be used in the BSSs. This field is present when dot11InterworkingServiceActivated is true.
RoamingConsortiumInfo	As defined in frame format	As defined in roaming consortium element in 8.4.2.98	Specifies identifying information for SSPs whose security credentials can be used to authenticate with the AP. This field may be present when dot11InterworkingServiceActivated is true.
Mesh ID	Octet string	0–32 octets	The value of MeshID. This element is present only if the BSSType = MESH.
Mesh Configuration	As defined in frame format	As defined in 8.4.2.100	Specifies the configuration of the mesh STA. This element is present only if the BSSType = MESH.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.11.2.3 When generated

This primitive is generated by the SME to start an infrastructure BSS (with the MAC entity within an AP), an IBSS (with the MAC entity acting as the first STA in the IBSS), or an MBSS (with the MAC entity acting as the first mesh STA in the MBSS) or to become a member of an existing MBSS. In an MBSS, this primitive starts the process of mesh beaconing.

An MLME-START.request primitive may be generated in an infrastructure BSS or IBSS only after an MLME-RESET.request primitive has been used to reset the MAC entity and before an MLME-JOIN.request primitive has been used to successfully join an existing infrastructure BSS or IBSS.

An MLME-START.request primitive may be generated in an MBSS only after an MLME-RESET.request primitive has been used to reset the MAC entity and before any synchronization and mesh peering have been established. When the mesh STA uses the default synchronization method and the default mesh peering protocol, the MLME-START.request primitive shall be generated before an MLME-MESHNEIGHBOROFFSETSYNCSTART.request primitive and MLME-MESHPEERINGMANAGEMENT.request primitive have been used.

The MLME-START.request primitive shall not be used after successful use of the MLME-START.request primitive or successful use of the MLME-JOIN.request primitive without generating an intervening MLME-RESET.request primitive.

6.3.11.2.4 Effect of receipt

This primitive initiates the BSS initialization procedure once the current frame exchange sequence is complete. The MLME subsequently issues an MLME-START.confirm primitive that reflects the results of the creation procedure.

If an MLME receives an MLME-START.request primitive with a BSSBasicRateSet parameter containing any unsupported rates, the MLME response in the resulting MLME-START.confirm primitive shall contain a ResultCode parameter that is not set to the value SUCCESS.

If the MLME of an HT STA receives an MLME-START.request primitive with a BSSBasicMCSSet parameter containing any unsupported MCSs, the MLME response in the resulting MLME-START.confirm primitive shall contain a ResultCode parameter that is not set to the value SUCCESS.

If an MLME receives an MLME-START.request primitive with the SSIDEncoding parameter value UTF8, the MLME shall set the UTF-8 SSID subfield of the Extended Capabilities element to one in Beacon and Probe Response frames.

6.3.11.3 MLME-START.confirm

6.3.11.3.1 Function

This primitive reports the results of a BSS creation procedure or a procedure to become a member of an MBSS.

6.3.11.3.2 Semantics of the service primitive

The primitive parameter is as follows:

```
  MLME-START.confirm(
                                    ResultCode
                                    )
```

Name	Type	Valid range	resetDescription
ResultCode	Enumeration	SUCCESS, BSS_ALREADY_STARTED_OR_JOINED, RESET_REQUIRED_BEFORE_ START, NOT_SUPPORTED	Indicates the result of the MLME-START.request primitive.

6.3.11.3.3 When generated

This primitive is generated by the MLME as a result of an MLME-START.request primitive to create a new BSS or to become a member of an MBSS.

6.3.11.3.4 Effect of receipt

The SME is notified of the results of the BSS creation procedure or a procedure to become a member of an MBSS.

6.3.12 Stop

6.3.12.1 General

This mechanism supports the process of terminating an existing BSS.

6.3.12.2 MLME-STOP.request

6.3.12.2.1 Function

This primitive requests that the MAC entity stop a BSS previously started by using an MLME-START.request primitive

6.3.12.2.2 Semantics of the service primitive

The primitive parameter is as follows:

```
          MLME-STOP.request(
                              SSID
                              )
```

Name	Type	Valid range	Description
SSID	Octet string	0–32 octets	The SSID of the BSS to be stopped.

6.3.12.2.3 When generated

This primitive is generated by the SME to terminate an infrastructure BSS (with the MAC entity within an AP). The MLME-STOP.request primitive shall be generated only after successful use of an MLME-START.confirm primitive.

The MLME-STOP termination procedure does not reset the MAC to initial conditions. An MLME-RESET.request primitive shall be issued prior to use of the MLME-START.request primitive and subsequent to the use of an MLME-STOP.request primitive.

The SME should notify associated non-AP STAs of imminent infrastructure BSS termination before issuing the MLME-STOP.request. This can be done with the BSS Transition Management procedure, using the Termination information.

6.3.12.2.4 Effect of receipt

This primitive initiates the termination of the BSS. All services provided by the AP to an infrastructure BSS, including Beacons, Probe Responses and access to the DS, are stopped by the termination. All STAs in an infrastructure BSS are deauthenticated by the termination.

6.3.13 Protocol layer model for spectrum management and radio measurement

The layer management extensions for measurement, TPC, and channel switching assume a certain partition of functionality between the MLME and SME. This partitioning assumes that policy decisions (e.g., regarding measurement and channel switching) reside in the SME, while the protocol for measurement, switch timing, and the associated frame exchanges resides within the MLME (see Figure 6-2).

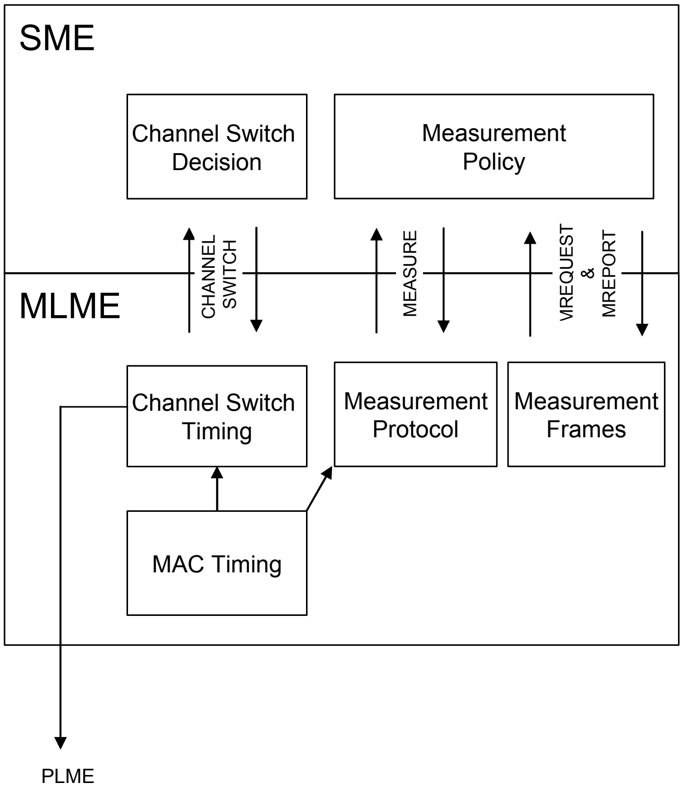


Figure 6-2—Layer management model

The informative diagrams within this subclause further illustrate the protocol layer model adopted. Figure 6-3 and Figure 6-4 depict the measurement process for a peer STA to accept and reject a measurement request, respectively. Figure 6-5 illustrates the TPC adaptation process. Lastly, Figure 6-6 depicts the management process for a channel switch using a Channel Switch Announcement frame.

It should be noted that these diagrams are intended as examples and do not depict all possible protocol scenarios, e.g., a measurement request may result in more than one measurement report frame as described in 10.9.7 and 10.11.

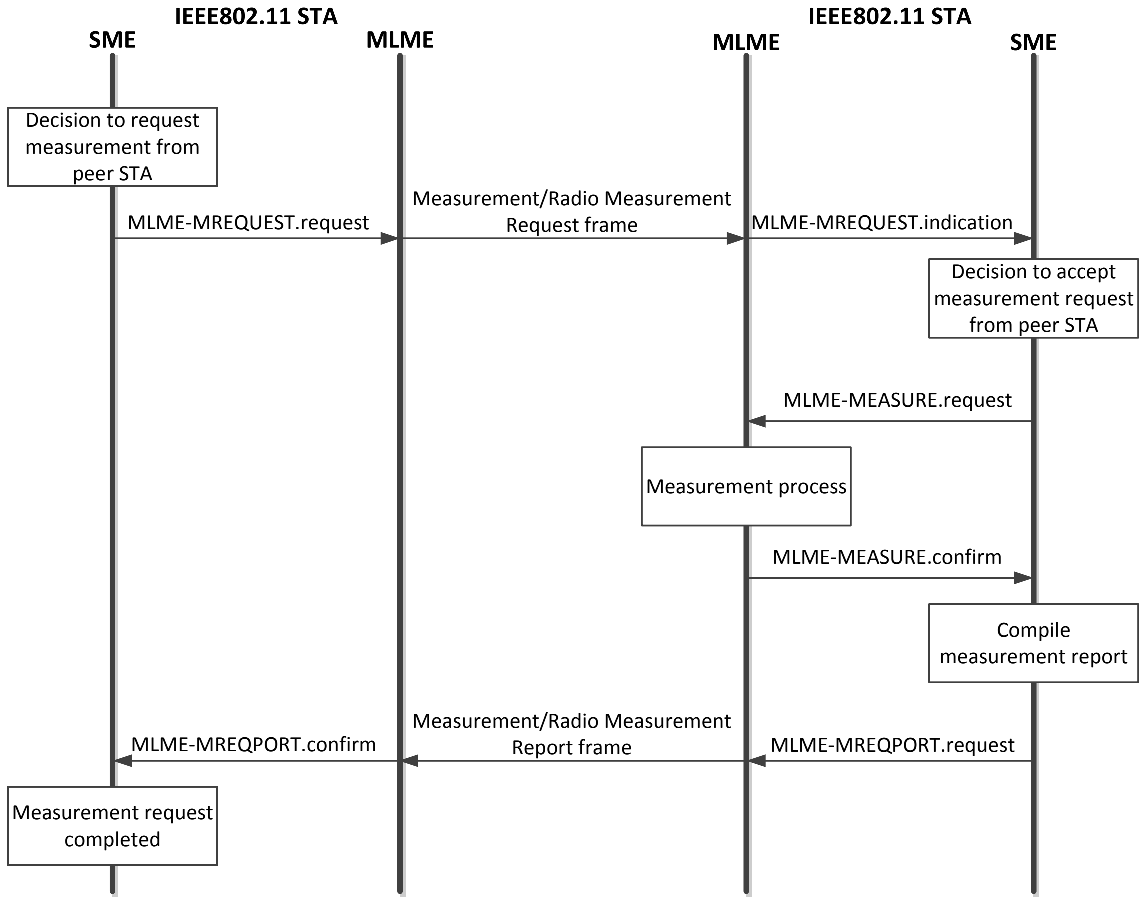

IEEE802.11 STA
SME
MLME
IEEE802.11 STA
MLME
SME
Decision to request measurement from peer STA
MLME-MREQUEST.request
Measurement/Radio Measurement Request frame
MLME-MREQUEST.indication
Decision to accept measurement request from peer STA
MLME-MEASURE.request
Measurement process
MLME-MEASURE.confirm
Compile measurement report
MLME-MREQPORT.confirm
Measurement/Radio Measurement Report frame
MLME-MREQPORT.request
Measurement request completed

Figure 6-3—Measurement request—accepted

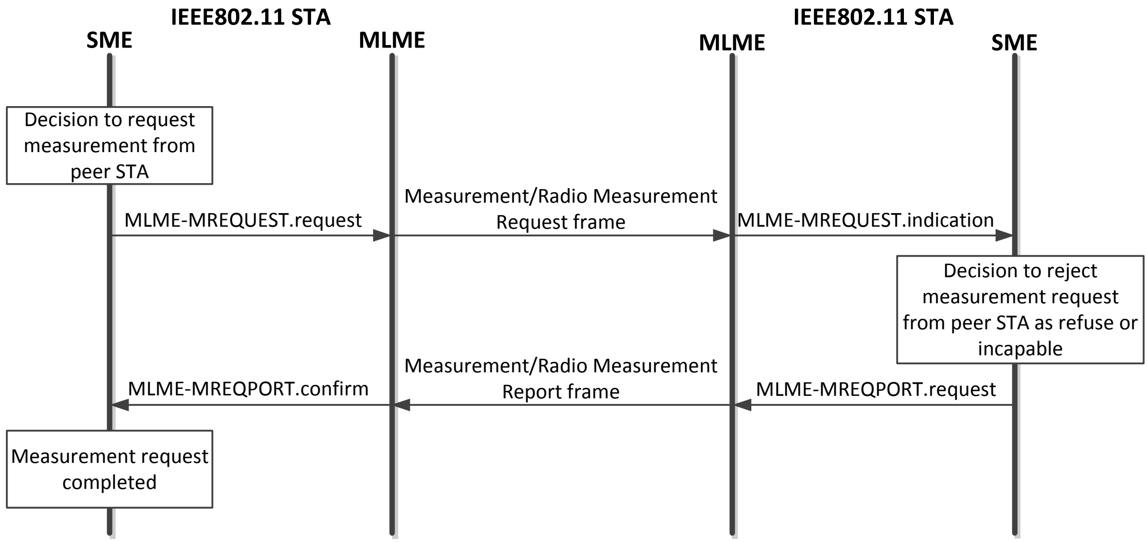

IEEE802.11 STA
SME
MLME
IEEE802.11 STA
MLME
SME
Decision to request measurement from peer STA
MLME-MREQUEST.request
Measurement/Radio Measurement Request frame
MLME-MREQUEST.indication
Decision to reject measurement request from peer STA as refuse or incapable
MLME-MREQPORT.confirm
Measurement/Radio Measurement Report frame
MLME-MREQPORT.request
Measurement request completed

Figure 6-4—Measurement request—rejected

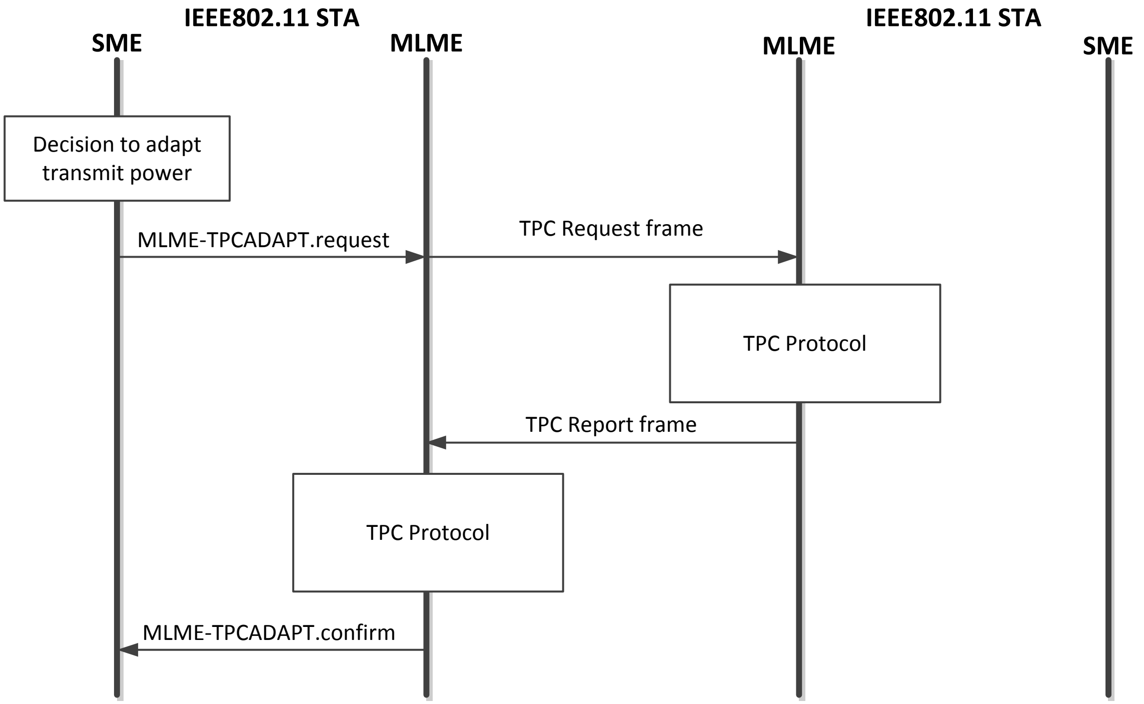


Figure 6-5—TPC adaptation

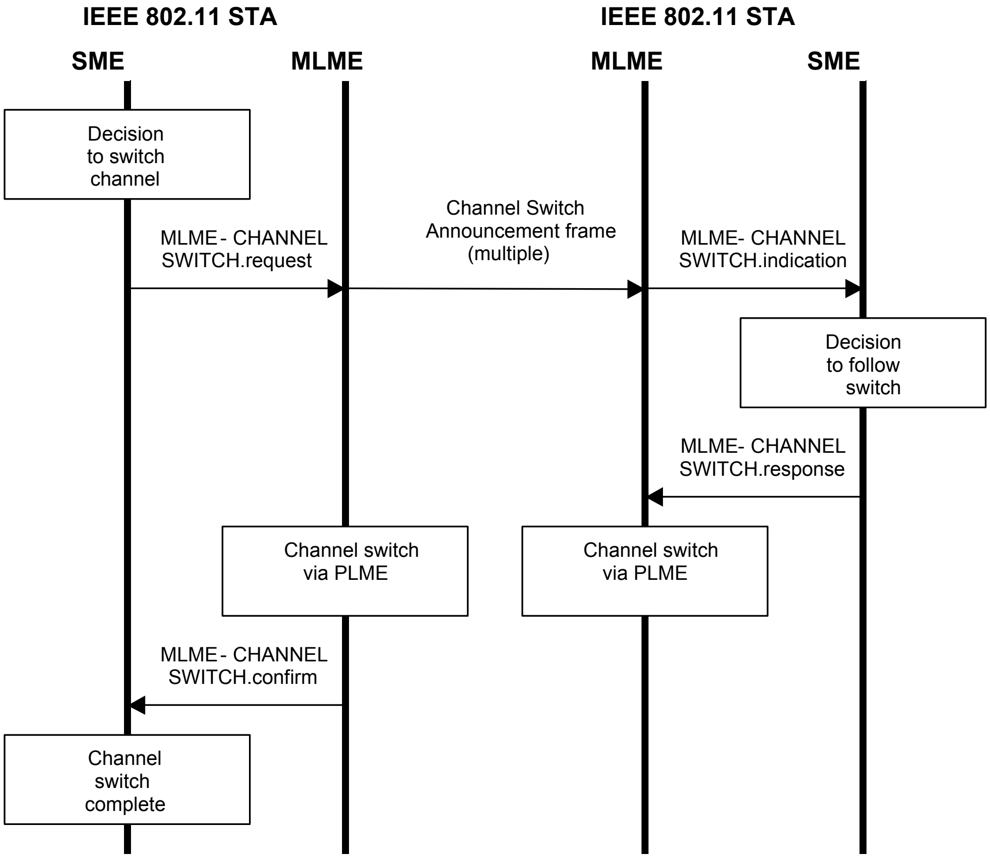

IEEE 802.11 STA
IEEE 802.11 STA
SME
MLME
MLME
SME
Decision to switch channel
MLME- CHANNEL SWITCH.request
Channel Switch Announcement frame (multiple)
MLME- CHANNEL SWITCH.indication
Decision to follow switch
MLME- CHANNEL SWITCH.response
Channel switch via PLME
Channel switch via PLME
MLME- CHANNEL SWITCH.confirm
Channel switch complete

Figure 6-6—Channel switch

6.3.14 Measurement request

6.3.14.1 Introduction

This set of primitives supports the signaling of measurement requests between peer SMEs.

6.3.14.2 MLME-MREQUEST.request

6.3.14.2.1 Function

This primitive requests the transmission of a measurement request to a peer entity.

6.3.14.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MREQUEST.request(
                              Peer MAC Address,
                              Dialog Token,
                              Measurement Request Set,
                              Number of Repetitions,
                              Measurement Category,
                              VendorSpecificInfo
                              )
```

Name	Type	Valid range	Description
Peer MAC Address	MACAddress	Any valid individual or group MAC address	The address of the peer MAC entity to which the measurement request is transmitted.
Dialog Token	Integer	1–255	The dialog token to identify the measurement transaction.
Measurement Request Set	Set of measurement requests, each as defined in the Measurement Request element format	Set of measurement requests, each as defined in 8.4.2.23	A set of measurement requests, each containing a Measurement Token, Measurement Request Mode, Measurement Type, and Measurement Request.
Number of Repetitions	Integer	0 – 65 535	The number of times the Measurement Request Set is to be repeated. The parameter is present only if Measurement Category is Radio Measurement and dot11RadioMeasurementActivated is true.
Measurement Category	Enumeration	SPECTRUM MANAGEMENT, or RADIO MEASUREMENT	Indicates whether the Measurement Report Set is a set of Spectrum Management or Radio Measurement measurement requests. The parameter is present only if dot11RadioMeasurementActivated is true.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.14.2.3 When generated

This primitive is generated by the SME to request that a Measurement Request frame be sent to a peer entity to initiate one or more measurements.

6.3.14.2.4 Effect of receipt

On receipt of this primitive, the MLME constructs a Measurement Request frame containing the set of Measurement Request elements specified. This frame is then scheduled for transmission.

6.3.14.3 MLME-MREQUEST.indication

6.3.14.3.1 Function

This primitive indicates that a Measurement Request frame has been received requesting the measurement of one or more channels.

6.3.14.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-MREQUEST.indication(
                                    Peer MAC Address,
                                    Dialog Token,
                                    Measurement Request Set,
                                    Number of Repetitions,
                                    Measurement Category,
                                    VendorSpecificInfo
                                    )
```

Name	Type	Valid range	Description
Peer MAC Address	MACAddress	Any valid individual MAC address	The address of the peer MAC entity from which the measurement request was received.
Dialog Token	Integer	1–255	The dialog token to identify the measurement transaction.
Measurement Request Set	Set of measurement requests, each as defined in the Measurement Request element format	Set of measurement requests, each as defined in 8.4.2.23	A set of measurement requests, each containing a Measurement Token, Measurement Request Mode, Measurement Type, and Measurement Request.
Number of Repetitions	Integer	0 – 65 535	The number of times the Measurement Request Set is to be repeated. The parameter is present only if Measurement Category is Radio Measurement and dot11RadioMeasurementActivated is true.
Measurement Category	Enumeration	SPECTRUM MANAGEMENT, or RADIO MEASUREMENT	Indicates whether the Measurement Report Set is a set of Spectrum Management or Radio Measurement measurement requests. The parameter is present only if dot11RadioMeasurementActivated is true.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.14.3.3 When generated

This primitive is generated by the MLME when a valid Measurement Request frame is received.

6.3.14.3.4 Effect of receipt

On receipt of this primitive, the SME either rejects the request or commences the requested measurements.

6.3.15 Channel measurement

6.3.15.1 Introduction

This set of primitives supports the requesting and reporting of measurement data.

6.3.15.2 MLME-MEASURE.request

6.3.15.2.1 Function

This primitive is generated by the SME to request that the MLME initiate specified measurements.

6.3.15.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MEASURE.request(
                              Dialog Token,
                              Measurement Request Set,
                              VendorSpecificInfo
                              )
```

Name	Type	Valid range	Description
Dialog Token	Integer	0–255	The Dialog Token to identify the measurement transaction.
Measurement Request Set	Set of measurement requests, each as defined in the Measurement Request element	Set of measurement requests, each as defined in 8.4.2.23	A set of measurement requests, each containing a Measurement Token, Measurement Request Mode, Measurement Type, and Measurement Request.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.15.2.3 When generated

This primitive is generated by the SME to request that the MLME initiate the specified measurements.

6.3.15.2.4 Effect of receipt

On receipt of this primitive, the MLME commences the measurement process.

6.3.15.3 MLME-MEASURE.confirm

6.3.15.3.1 Function

This primitive reports the result of a measurement.

6.3.15.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MEASURE.confirm(
                              ResultCode,
                              Dialog Token,
                              Measurement Report Set,
                              VendorSpecificInfo
                              )
```

Name	Type	Valid range	Description
ResultCode	Enumeration	SUCCESS, UNSPECIFIED_FAILURE	The outcome of the measurement request.
Dialog Token	Integer	0–255	The dialog token to identify the measurement transaction.
Measurement Report Set	Set of measurement reports, each as defined in the Measurement Report element	Set of measurement reports, each as defined in 8.4.2.24	A set of measurement reports, each containing a Measurement Token, Measurement Report Mode, Measurement Type, and Measurement Report.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.15.3.3 When generated

This primitive is generated by the MLME to report the results when a measurement set completes.

6.3.15.3.4 Effect of receipt

On receipt of this primitive, the SME evaluates the result code and, if appropriate, stores the channel measurements pending communication to the requesting entity or for local use.

6.3.16 Measurement report

6.3.16.1 Introduction

This set of primitives supports the signaling of measurement reports.

6.3.16.2 MLME-MREPORT.request

6.3.16.2.1 Function

This primitive supports the signaling of measurement reports between peer SMEs.

6.3.16.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MREPORT.request(
                              Peer MAC Address,
                              Dialog Token,
                              Measurement Report Set,
                              Measurement Category,
                              VendorSpecificInfo
                              )
```

Name	Type	Valid range	Description
Peer MAC Address	MACAddress	Any valid individual MAC address	The address of the peer MAC entity to which the measurement report is sent.
Dialog Token	Integer	0–255	The dialog token to identify the measurement transaction. Set to 0 for an autonomous report.
Measurement Report Set	Set of measurement reports, each as defined in the Measurement Report element format	Set of measurement reports, each as defined in 8.4.2.24	A set of measurement reports, each containing a Measurement Token, Measurement Report Mode, Measurement Type, and Measurement Report.
Measurement Category	Enumeration	SPECTRUM MANAGEMENT, or RADIO MEASUREMENT	Indicates whether the Measurement Report Set is a set of Spectrum Management or Radio Measurement reports. The parameter is present only if dot11RadioMeasurementActivated is true.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.16.2.3 When generated

This primitive is generated by the SME to request that a frame be sent to a peer entity to report the results of measuring one or more channels.

6.3.16.2.4 Effect of receipt

On receipt of this primitive, the MLME constructs a Measurement Report frame containing the set of measurement reports. This frame is then scheduled for transmission.

6.3.16.3 MLME-MREPORT.indication

6.3.16.3.1 Function

This primitive indicates that a Measurement Report frame has been received from a peer entity. This management report is either a response to an earlier measurement request (e.g., MLME-MREQUEST.request primitive) or an autonomous report.

6.3.16.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MREPORT.indication(
                              Peer MAC Address,
                              Dialog Token,
                              Measurement Report Set,
                              Measurement Category,
                              VendorSpecificInfo
                              )
```

Name	Type	Valid range	Description
Peer MAC Address	MACAddress	Any valid individual MAC address	The address of the peer MAC entity from which the Measurement Report frame was received.

Name	Type	Valid range	Description
Dialog Token	Integer	0–255	The dialog token to identify the measurement transaction. Set to 0 for an autonomous report.
Measurement Report Set	Set of measurement reports, each as defined in the Measurement Report element format	Set of measurement reports, each as defined in 8.4.2.24	A set of measurement reports, each containing a Measurement Token, Measurement Report Mode, Measurement Type, and Measurement Report.
Measurement Category	Enumeration	SPECTRUM MANAGEMENT, or RADIO MEASUREMENT	Indicates whether the Measurement Report Set is a set of Spectrum Management or Radio Measurement reports. The parameter is present only if dot11RadioMeasurementActivated is true.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.16.3.3 When generated

This primitive is generated by the MLME when a valid Measurement Report frame is received.

6.3.16.3.4 Effect of receipt

On receipt of this primitive, measurement data might be available for SME processes, such as channel selection.

6.3.17 Channel switch

6.3.17.1 MLME-CHANNELSWITCH.request

6.3.17.1.1 Function

This primitive requests a switch to a new operating channel.

6.3.17.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-CHANNELSWITCH.request(
                                Mode,
                                Channel Number,
                                Secondary Channel Offset,
                                Channel Switch Count,
                                Mesh Channel Switch Parameters,
                                VendorSpecificInfo
                                )
```

Name	Type	Valid range	Description
Mode	Integer	0, 1	Channel switch mode, as defined for the Channel Switch Announcement element.
Channel Number	Integer	As defined in 18.3.8.4.3	Specifies the new channel number.

Name	Type	Valid range	Description
Secondary Channel Offset	Integer	As in Table 8-57	Specifies the position of secondary channel in relation to the primary channel. The parameter is present if dot11FortyMHzOperationImplemented is true; otherwise, the parameter is not present.
Channel Switch Count	As defined in 8.4.2.21	As defined in 8.4.2.21	Specifies the time period until the channel switch event, as described in 8.4.2.21
Mesh Channel Switch Parameters	As defined in 8.4.2.105	As defined in 8.4.2.105	Specifies MBSS Channel Switch Parameters used by a mesh STA. This parameter is present if the dot11MeshActivated is true; otherwise, the parameter is not present.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

The Secondary Channel Offset parameter may be present for HT STAs.

6.3.17.1.3 When generated

This primitive is generated by the SME to schedule a channel switch and announce this switch to peer entities in the BSS.

6.3.17.1.4 Effect of receipt

On receipt of this primitive, the MLME schedules the channel switch event and announces this switch to other STAs in the BSS using the Channel Switch Announcement frame or element. The MLME sets the timing of the frame transmission taking into account the the activation delay.

6.3.17.2 MLME-CHANNELSWITCH.confirm

6.3.17.2.1 Function

This primitive reports the result of a request to switch channel.

6.3.17.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-CHANNELSWITCH.confirm(
                              ResultCode,
                              VendorSpecificInfo
                              )
```

Name	Type	Valid range	Description
ResultCode	Enumeration	SUCCESS, UNSPECIFIED FAILURE	Reports the result of a channel switch request.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.17.2.3 When generated

This primitive is generated by the MLME when a channel switch request completes. Possible unspecified failure causes include an inability to schedule a channel switch announcement.

6.3.17.2.4 Effect of receipt

The SME is notified of the results of the channel switch procedure.

6.3.17.3 MLME-CHANNELSWITCH.indication

6.3.17.3.1 Function

This primitive indicates that a channel switch announcement has been received from a peer entity.

6.3.17.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-CHANNELSWITCH.indication(
                                    Peer MAC Address,
                                    Mode,
                                    Channel Number,
                                    Secondary Channel Offset,
                                    Channel Switch Count,
                                    Mesh Channel Switch Parameters,
                                    VendorSpecificInfo
                                    )
```

Name	Type	Valid range	Description
Peer MAC Address	MACAddress	Any valid individual MAC address	The address of the peer MAC entity from which the Measurement Report frame was received.
Mode	Integer	0, 1	Channel switch mode, as defined for the Channel Switch Announcement element.
Channel Number	Integer	As defined in 18.3.8.4.3	Specifies the new channel number.
Secondary Channel Offset	Integer	As in Table 8-57	Specifies the position of secondary channel in relation to the primary channel. The parameter is optionally present only if dot11FortyMHzOperationImplemented is true.
Channel Switch Count	As defined in 8.4.2.21	As defined in 8.4.2.21	Specifies the time period until the channel switch event, as described in 8.4.2.21
Mesh Channel Switch Parameters	As defined in 8.4.2.105	As defined in 8.4.2.105	Specifies MBSS Channel Switch Parameters used by a mesh STA. This parameter is present if the dot11MeshActivated is true; otherwise, the parameter is not present.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.17.3.3 When generated

This primitive is generated by the MLME when a valid Channel Switch Announcement frame is received.

6.3.17.3.4 Effect of receipt

On receipt of this primitive, the SME decides whether to accept the switch.

6.3.17.4 MLME-CHANNELSWITCH.response

6.3.17.4.1 Function

This primitive is used to schedule an accepted channel switch.

6.3.17.4.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-CHANNELSWITCH.response(
                              Mode,
                              Channel Number,
                              Secondary Channel Offset,
                              Channel Switch Count,
                              Mesh Channel Switch Parameters,
                              VendorSpecificInfo
                              )
```

Name	Type	Valid range	Description
Mode	Integer	0, 1	Channel switch mode, as defined for the Channel Switch Announcement element.
Channel Number	Integer	As defined in 18.3.8.4.3	Specifies the new channel number.
Secondary Channel Offset	Integer	As in Table 8-57	Specifies the position of secondary channel in relation to the primary channel. The parameter is optionally present only if dot11FortyMHzOperationImplemented is true.
Channel Switch Count	As defined in 8.4.2.21	As defined in 8.4.2.21	Specifies the time period until the channel switch event, as described in 8.4.2.21
Mesh Channel Switch Parameters	As defined in 8.4.2.105	As defined in 8.4.2.105	Specifies MBSS Channel Switch Parameters used by a mesh STA. This parameter is present if the dot11MeshActivated is true; otherwise, the parameter is not present.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.17.4.3 When generated

This primitive is generated by the SME to schedule an accepted channel switch request.

6.3.17.4.4 Effect of receipt

On receipt of this primitive, the MLME schedules the channel switch.

6.3.18 TPC request

6.3.18.1 Introduction

This set of primitives supports the adaptation of transmit power between peer entities as described in 10.8.6.

6.3.18.2 MLME-TPCADAPT.request

6.3.18.2.1 Function

This primitive supports the adaptation of transmit power between peer entities as specified in 10.8.6.

6.3.18.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-TPCADAPT.request(
                              Peer MAC Address,
                              Dialog Token,
                              VendorSpecificInfo
                              )
```

Name	Type	Valid range	Description
Peer MAC Address	MACAddress	Any valid individual or group MAC address	The address of the peer MAC entity to which the TPC request is sent.
Dialog Token	Integer	1–255	The dialog token to identify the TPC transaction.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.18.2.3 When generated

This primitive is generated by the SME to request that a TPC Request frame be sent to a peer entity to request that entity to report transmit power and link margin information.

6.3.18.2.4 Effect of receipt

On receipt of this primitive, the MLME constructs a TPC Request frame. This frame is then scheduled for transmission.

6.3.18.3 MLME-TPCADAPT.confirm

6.3.18.3.1 Function

This primitive reports the result of the TPC adaptation procedure.

6.3.18.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-TPCADAPT.confirm(
                              ResultCode
                              Transmit Power,
                              Link Margin,
                              Rate,
                              VendorSpecificInfo
                              )
```

Name	Type	Valid range	Description
ResultCode	Enumeration	SUCCESS, UNSPECIFIED FAILURE	Reports the outcome of a request to send a TPC Request frame.
Transmit Power	Integer	–127 to 127	Value of the Transmit Power field of the TPC Report element of the TPC Report frame.
Link Margin	Integer	–127 to 127	Value of the Link Margin field of the TPC Report element of the TPC Report frame.
Rate	Integer	As defined in 8.4.2.3	The rate at which the TPC Request frame was sent.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.18.3.3 When generated

This primitive is generated by the MLME when the TPC adaptation procedure completes.

6.3.18.3.4 Effect of receipt

The SME is notified of the results of the TPC adaptation procedure.

6.3.19 SetKeys

6.3.19.1 MLME-SETKEYS.request

6.3.19.1.1 Function

This primitive causes the keys identified in the parameters of the primitive to be set in the MAC and enabled for use.

6.3.19.1.2 Semantics of the service primitive

The primitive parameter is as follows:

```
   MLME-SETKEYS.request(
                                    Keylist
                                    )
```

Name	Type	Valid range	Description
Keylist	A set of SetKeyDescriptors	N/A	The list of keys to be used by the MAC.

Each SetKeyDescriptor consists of the following elements:

Name	Type	Valid range	Description
Key	Bit string	N/A	The temporal key value
Length	Integer	N/A	The number of bits in the Key to be used.
Key ID	Integer	0–3 shall be used with WEP, TKIP, and CCMP; 4–5 with BIP; and 6–4095 are reserved	Key identifier
Key Type	Integer	Group, Pairwise, PeerKey, IGTK	Defines whether this key is a group key, pairwise key, PeerKey, or Integrity Group key.

Name	Type	Valid range	Description
Address	MACAddress	Any valid individual MAC address	This parameter is valid only when the Key Type value is Pairwise, when the Key Type value is Group and the STA is in IBSS, or when the Key Type value is PeerKey.
Receive Sequence Count	8 octets	N/A	Value to which the RSC(s) is initialized
Is Authenticator	Boolean	true, false	Indicates whether the key is configured by the Authenticator or Supplicant. The value true indicates Authenticator.
Cipher Suite Selector	4 octets	As defined in 8.4.2.27	The cipher suite required for this association.
Direction	Integer	Receive, Transmit, Both	Indicates the direction for which the keys are to be installed. Receive indicates that the keys are being installed for the receive direction. Transmit indicates that the keys are being installed for the transmit direction. Both indicates that the keys are being installed for both the receive and transmit directions.

6.3.19.1.3 When generated

This primitive is generated by the SME at any time when one or more keys are to be set in the MAC.

6.3.19.1.4 Effect of receipt

Receipt of this primitive causes the MAC to apply the keys as follows provided the MLME-SETPROTECTION.request primitive has been issued:

- If the Direction element of the SetKeyDescriptor indicates Transmit or Both then the MAC uses the key information for the transmission of all subsequent frames to which the key applies.
- If the Direction element of the SetKeyDescriptor indicates Receive or Both then the MAC installs the key with the associated Key ID such that received frames of the appropriate type and containing the matching Key ID are processed using that key and its associated state information.

6.3.20 DeleteKeys

6.3.20.1 MLME-DELETEKEYS.request

6.3.20.1.1 Function

This primitive causes the keys identified in the parameters of the primitive to be deleted from the MAC and thus disabled for use.

6.3.20.1.2 Semantics of the service primitive

The primitive parameter is as follows:

```
MLME-DELETEKEYS.request(
                              Keylist
                              )
```

Name	Type	Valid range	Description
Keylist	A set of DeleteKeyDescriptors	N/A	The list of keys to be deleted from the MAC.

Each DeleteKeyDescriptor consists of the following elements:

Name	Type	Valid range	Description
Key ID	Integer	N/A	Key identifier.
Key Type	Integer	Group, Pairwise, PeerKey, IGTK	Defines whether this key is a group key, pairwise key, PeerKey, or Integrity Group key.
Address	MACAddress	Any valid individual MAC address	This parameter is valid only when the Key Type value is Pairwise, or when the Key Type value is Group and is from an IBSS STA, or when the Key Type value is PeerKey.

6.3.20.1.3 When generated

This primitive is generated by the SME at any time when keys for a security association are to be deleted in the MAC.

6.3.20.1.4 Effect of receipt

Receipt of this primitive causes the MAC to delete the temporal keys identified by the Keylist Address, including Group, Pairwise and PeerKey, and to cease using them.

6.3.21 MIC (Michael) failure event

6.3.21.1 MLME-MICHAELMICFAILURE.indication

6.3.21.1.1 Function

This primitive reports that a MIC failure event was detected.

6.3.21.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MICHAELMICFAILURE.indication (
                                   Count,
                                   Address,
                                   Key Type,
                                   Key ID,
                                   TSC
                                   )
```

Name	Type	Valid range	Description
Count	Integer	1 or 2	The current number of MIC failure events.
Address	MACAddress	Any valid individual MAC address	The source MAC address of the frame.
Key Type	Integer	Group, Pairwise, PeerKey	The key type that the received frame used.
Key ID	Integer	0–3	Key identifier.
TSC	6 octets	N/A	The TSC value of the frame that generated the MIC failure.

6.3.21.1.3 When generated

This primitive is generated by the MAC when it has detected a MIC failure.

6.3.21.1.4 Effect of receipt

The SME is notified that the MAC has detected a MIC failure; see 11.4.2.4.

6.3.22 EAPOL

6.3.22.1 MLME-EAPOL.request

6.3.22.1.1 Function

This primitive is generated by the SME when the SME has an IEEE 802.1X EAPOL-Key frame to send.

6.3.22.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-EAPOL.request (
                        Source Address,
                        Destination Address,
                        Data
                        )
```

Name	Type	Valid range	Description
Source Address	MACAddress	N/A	The MAC sublayer address from which the EAPOL-Key frame is being sent.
Destination Address	MACAddress	N/A	The MAC sublayer entity address to which the EAPOL-Key frame is being sent.
Data	IEEE 802.1X EAPOL-Key frame	N/A	The EAPOL-Key frame to be transmitted.

6.3.22.1.3 When generated

This primitive is generated by the SME when the SME has a 802.1X EAPOL-Key frame to send.

6.3.22.1.4 Effect of receipt

The MAC sends this EAPOL-Key frame.

6.3.22.2 MLME-EAPOL.confirm

6.3.22.2.1 Function

This primitive indicates that this EAPOL-Key frame has been transmitted by the IEEE 802.11 MAC.

6.3.22.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-EAPOL.confirm (
                        ResultCode
                        )
```

Name	Type	Valid range	Description
ResultCode	Enumeration	SUCCESS, TRANSMISSION_FAILURE	Indicates whether the EAPOL-Key frame has been transmitted to the target STA.

6.3.22.2.3 When generated

This primitive is generated by the MAC as a result of an MLME-EAPOL.request primitive being generated to send an EAPOL-Key frame.

6.3.22.2.4 Effect of receipt

The SME is always notified whether this EAPOL-Key frame has been transmitted to the IEEE 802.11 MAC.

This primitive communicates that the frame has been transmitted; see the procedures in 11.4.2.4.1.

6.3.23 MLME-PeerKeySTART

6.3.23.1 MLME- PeerKeySTART.request

6.3.23.1.1 Function

This primitive is generated by the SME to start a PeerKey Handshake with a peer.

6.3.23.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```
    MLME-PeerKeySTART.request (
                                   PeerSTAAddress,
                                   RSN
                                   )
```

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity with which to perform the PeerKey Handshake process.
RSN	RSNE	As defined in 8.4.2.27	A description of the cipher suites supported by initiator STA.

6.3.23.1.3 When generated

This primitive is generated by the SME for a STA to initiate a PeerKey Handshake with a specified peer MAC entity in order to create a secure link between the two STAs.

6.3.23.1.4 Effect of receipt

This primitive initiates the SMK Handshake as part of PeerKey Handshake by sending an EAPOL-Key message.

6.3.24 SetProtection

6.3.24.1 MLME-SETPROTECTION.request

6.3.24.1.1 Function

This primitive indicates whether protection is required for frames sent to and received from the indicated MAC address.

6.3.24.1.2 Semantics of the service primitive

The primitive parameter is as follows:

```
MLME-SETPROTECTION.request(
                                    Protectlist
                                    )
```

Name	Type	Valid range	Description
Protectlist	A set of protection elements	N/A	The list of how each key is being used currently.

Each Protectlist consists of the following elements:

Name	Type	Valid range	Description
Address	MACAddress	Any valid individual MAC address	This parameter is valid only when the Key Type value is Pairwise or when the Key Type value is Group and is from an IBSS STA or PeerKey.
ProtectType	Enumeration	None, Rx, Tx, Rx_Tx	The protection value for this MAC.
Key Type	Integer	Group, Pairwise, PeerKey, IGTK	Defines whether this key is a group key, pairwise key, PeerKey, or Integrity Group key.

6.3.24.1.3 When generated

This primitive is generated by the SME when protection is required for frames sent to and received from the indicated MAC address.

6.3.24.1.4 Effect of receipt

Receipt of this primitive causes the MAC to set the protection and to protect data frames as indicated in the ProtectType element of the Protectlist parameter:

- None: Specifies that data frames neither from the MAC address nor to the MAC address are protected.
- Rx: Specifies that data frames from the MAC address are protected (i.e., any data frames without protection received from the MAC address are discarded).
- Tx: Specifies that data frames to the MAC address are protected.
- Rx_Tx: Specifies that data frames to and from the MAC address are protected.

Once data frames are protected to and/or from the specified MAC address, the MLME-SETPROTECTION.request primitive is used to reset the prior setting. Invocation of the MLME-SETPROTECTION.request primitive with a ProtectType of None deletes a protection state.

6.3.25 MLME-PROTECTEDFRAMEDROPPED

6.3.25.1 MLME- PROTECTEDFRAMEDROPPED.indication

6.3.25.1.1 Function

This primitive notifies the SME that a frame has been dropped because a temporal key was unavailable.

6.3.25.1.2 Semantics of the service primitive

This primitive has two parameters, the MAC addresses of the two STAs.

The primitive parameters are as follows:

```
   MLME-PROTECTEDFRAMEDROPPED.indication (
                                                    Address1,
                                                    Address2
                                                    )
```

Name	Type	Valid range	Description
Address1	MACAddress	Any valid individual MAC address	MAC address of TA.
Address2	MACAddress	Any valid individual MAC address	MAC address of RA.

6.3.25.1.3 When generated

This primitive is generated by the MAC when a frame is dropped because no temporal key is available for the frame.

6.3.25.1.4 Effect of receipt

The SME is notified that a frame was dropped. The SME can use this information in an IBSS to initiate a security association to the peer STA.

6.3.26 TS management interface

6.3.26.1 General

This mechanism supports the process of adding, modifying, or deleting a TS in a BSS using the procedures defined in 10.4.

The primitives used for this mechanism are called *TS Management primitives*, which include MLME-ADDTS.xxx and MLME-DELTS.xxx primitives, where xxx denotes request, confirm, indication, or response. Each primitive contains parameters that correspond to a QoS Action frame. Requests and responses may cause the transmission of the corresponding QoS Action frames. Confirms and indications are issued upon the receipt of the appropriate QoS Action frame.

Table 6-1 defines which primitives are supported by which type of STA.

Table 6-1—Supported TS management primitives

Primitive	Request	Confirm	Indication	Response
ADDTS	non-AP QoS STA	non-AP QoS STA	HC	HC
DELTS	non-AP QoS STA and HC	non-AP QoS STA and HC	non-AP QoS STA and HC	—

6.3.26.2 MLME-ADDTS.request

6.3.26.2.1 Function

This primitive requests addition (or modification) of a TS. It requests the HC to admit the new or changed TS.

6.3.26.2.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-ADDTS.request (
DialogToken,
TSPEC,
TCLAS,
TCLASProcessing,
ADDTSFailureTimeout,
U-APSD Coexistence,
EBR,
VendorSpecificInfo
)

Name	Type	Valid range	Description
DialogToken	Integer	0–127	Specifies a number unique to the QoS Action primitives and frames used in adding (or modifying) the TS of concern.
TSPEC	As defined in TSPEC element with the exception of the surplus bandwidth allowance, minimum PHY rate, and maximum and minimum SIs, which are optionally specified	As defined in 8.4.2.32 with the exception of the surplus bandwidth allowance, minimum PHY rate, and maximum and minimum SIs, which are optionally specified	Specifies the TSID, traffic characteristics, and QoS requirements of the TS of concern.
TCLAS	TCLAS element	As defined in 8.4.2.33	Zero or more TCLAS elements. Specifies the rules and parameters by which an MSDU may be classified to the specified TS.
TCLASProcessing	TCLAS Processing element	As defined in 8.4.2.35	Specifies how the TCLAS elements are to be processed when there are multiple TCLAS elements.

Name	Type	Valid range	Description
ADDTSFailure-Timeout	Integer	Greater than or equal to 1	Specifies a time limit (in TU) after which the TS setup procedure is terminated.
U-APSD Coexistence	U-APSD Coexistence element	As defined in 8.4.2.93.	Indicates the coexistence parameters for requested transmission during a U-APSD service period.
EBR	Expedited Bandwidth Request element	As defined in 8.4.2.96	Specifies the precedence level of the TS request. This element is optionally present when dot11EBRActivated is true.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.26.2.3 When generated

This primitive is generated by the SME to request the addition of a new (or modification of an existing) TS in order to support parameterized QoS transport of the MSDUs belonging to this TS when a higher layer protocol or mechanism signals the STA to initiate such an addition (or modification).

6.3.26.2.4 Effect of receipt

The STA operates according to the procedures defined in 10.4.

6.3.26.3 MLME-ADDTS.confirm

6.3.26.3.1 Function

This primitive reports the results of a TS addition (or modification) attempt.

6.3.26.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-ADDTS.confirm(
                        ResultCode,
                        DialogToken,
                        TSDelay,
                        TSPEC,
                        Schedule,
                        TCLAS,
                        TCLASProcessing,
                        EBR,
                        VendorSpecificInfo
                        )
```

Name	Type	Valid range	Description
ResultCode	Enumeration	SUCCESS, INVALID_PARAMETERS, REJECTED_WITH_SUGGESTED_ CHANGES, REJECTED_FOR_DELAY_PERIOD, REJECTED_WITH_SUGGESTED_BSS_ TRANSITION, REQUESTED_TCLAS_NOT_SUPPORTED, TCLAS_RESOURCES_EXHAUSTED, REJECTED_HOME_WITH_ SUGGESTED_CHANGES, REJECTED_FOR_SSP_PERMISSIONS	Indicates the results of the corresponding MLME-ADDTS.request primitive.
DialogToken	Integer	As defined in the corresponding MLME-ADDTS.request primitive	Specifies a number unique to the QoS Action primitives and frames used in adding (or modifying) the TS.
TSDelay	Integer	≥ 0	When the result code is REJECTED_FOR_DELAY_ PERIOD, provides the amount of time a STA should wait before attempting another ADDTS request frame.
TSPEC	TSPEC element	As defined in 8.4.2.32	Specifies the TS information, traffic characteristics, and QoS requirements of the TS.
Schedule	Schedule element	As defined in 8.4.2.36	Specifies the schedule information, service start time, SI, and the specification interval.
TCLAS	TCLAS element	As defined in 8.4.2.33	Zero or more TCLAS elements. Specifies the rules and parameters by which an MSDU may be classified to the specified TS.
TCLASProcessing	TCLAS Processing element	As defined in 8.4.2.35	Specifies how the TCLAS elements are to be processed when there are multiple TCLAS elements.
EBR	Expedited Bandwidth Request element	As defined in 8.4.2.96	Specifies the precedence level of the TS request. This element is optionally present when dot11EBRActivated is true.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

For the ResultCode value of SUCCESS, the TSPEC and the optional TCLAS parameters describe the characteristics of the TS that has been created (or modified); and the specified (nonzero) parameters [with the exception of Service Start Time, Medium Time, and any possibly unspecified minimum set of parameters (see 9.19.4.3) in the TSPEC in ADDTS Request frame] exactly match those of the matching MLME-ADDTS.request primitive.

For other values of ResultCode, no new TS has been created. In the case of REJECTED_WITH_ SUGGESTED_CHANGES, the TSPEC represents an alternative proposal by the HC based on information about the current status of the MAC entity. In the case of REJECTED_HOME_WITH_SUGGESTED_CHANGES, the TSPEC represents an alternative proposal by the HC based on information received from the SSPN interface. A TS is not created with this definition. If the suggested changes are acceptable to the STA, it is the responsibility of the STA to set up the TS with the suggested changes.

In the case of REJECTED_WITH_SUGGESTED_BSS_TRANSITION, non-AP STA should retry TS setup process with the newly associated AP once the transition is done.

If this is the result of a modification of an existing TS, the status of that TS remains unchanged.

6.3.26.3.3 When generated

This primitive is generated by the MLME as a result of an MLME-ADDTS.request primitive indicating the results of that request.

This primitive is generated when that MLME-ADDTS.request primitive is found to contain invalid parameters, when a timeout occurs, or when the STA receives a response in the form of an ADDTS Response frame in the corresponding QoS Action frame from the HC.

6.3.26.3.4 Effect of receipt

The SME is notified of the results of the TS addition (or modification) procedure.

The SME should operate according to the procedures defined in 10.4.

6.3.26.4 MLME-ADDTS.indication

6.3.26.4.1 Function

This primitive reports to the HC’s SME the request for adding (or modifying) a TS.

6.3.26.4.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-ADDTS.indication (
                                    DialogToken,
                                    STAAddress,
                                    TSPEC,
                                    TCLAS,
                                    TCLASProcessing,
                                    U-APSD Coexistence,
                                    EBR,
                                    VendorSpecificInfo
                                    )
```

Name	Type	Valid range	Description
DialogToken	Integer	As defined in the received ADDTS request frame	Specifies a number unique to the QoS Action primitives and frames used in adding (or modifying) the TS.
STAAddress	MACAddress		Contains the MAC address of the STA that initiated the MLME-ADDTS.request primitive.

Name	Type	Valid range	Description
TSPEC	As defined in the TPEC element, with the exception of the surplus bandwidth allowance, minimum PHY rate, and maximum and minimum SIs, which are optionally specified	As defined in 8.4.2.32 with the exception of the surplus bandwidth allowance, minimum PHY rate, and maximum and minimum SIs, which are optionally specified	Specifies the TSID, traffic characteristics, and QoS requirements of the TS.
TCLAS	TCLAS element	As defined in 8.4.2.33	Zero or more TCLAS elements. Specifies the rules and parameters by which an MSDU may be classified to the specified TS.
TCLASProcessing	TCLAS Processing element	As defined in 8.4.2.35	Specifies how the TCLAS elements are to be processed when there are multiple TCLAS elements.
U-APSD Coexistence	U-APSD Coexistence element	As defined in 8.4.2.93	Indicates the coexistence parameters for requested transmission during a U-APSD service period.
EBR	Expedited Bandwidth Request element	As defined in 8.4.2.96	Specifies the precedence level of the TS request. This element is optionally present when dot11EBRActivated is true.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

The TCLAS is optional at the discretion of the STA that originated the request.

6.3.26.4.3 When generated

This primitive is generated by the MLME as a result of receipt of a request to add (or modify) a TS by a specified STA in the form of an ADDTS request frame.

6.3.26.4.4 Effect of receipt

The SME is notified of the request for a TS addition (or modification) by a specified STA.

This primitive solicits an MLME-ADDTS.response primitive from the SME that reflects the results of admission control at the HC on the TS requested to be added (or modified).

The SME should operate according to the procedures defined in 10.4.

The SME generates an MLME-ADDTS.response primitive within a dot11ADDTSResponseTimeout.

6.3.26.5 MLME-ADDTS.response

6.3.26.5.1 Function

This primitive responds to the request for a TS addition (or modification) by a specified STA's MAC entity.

6.3.26.5.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-ADDTS.response(
ResultCode,
DialogToken,
STAAddress,
TSDelay,
TSPEC,
Schedule,
TCLAS,
TCLASProcessing,
EBR,
VendorSpecificInfo
)

Name	Type	Valid range	Description
ResultCode	Enumeration	SUCCESS, INVALID_PARAMETERS, REJECTED_WITH_SUGGESTED_CHANGES, REJECTED_FOR_DELAY_PERIOD, REJECTED_WITH_SUGGESTED_BSS_TRANSITION, REQUESTED_TCLAS_NOT_SUPPORTED, TCLAS_RESOURCES_EXHAUSTED, REJECTED_HOME_WITH_SUGGESTED_CHANGES, REJECTED_FOR_SSP_PERMISSIONS	Indicates the results of the corresponding MLME-ADDTS.indication primitive.
DialogToken	Integer	As defined in the corresponding MLME-ADDTS.indication	DialogToken of the matching MLME-ADDTS.indication primitive.
STAAddress	MACAddress		Contains the STA address of the matching MLME-ADDTS.indication primitive.
TSDelay	Integer	≥ 0	When the result code is REJECTED_FOR_DELAY_PERIOD, provides the amount of time a STA should wait before attempting another ADDTS request.
TSPEC	TSPEC element	As defined in 8.4.2.32	Specifies the QoS parameters of the TS.
Schedule	Schedule element	As defined in 8.4.2.36	Specifies the schedule information, service start time, SI, and the specification interval.
TCLAS	TCLAS element	As defined in 8.4.2.33	Zero or more TCLAS elements. Specifies the rules and parameters by which an MSDU may be classified to the specified TS.

Name	Type	Valid range	Description
TCLASProcessing	TCLAS Processing element	As defined in 8.4.2.35	Specifies how the TCLAS elements are to be processed when there are multiple TCLAS elements.
EBR	Expedited Bandwidth Request element	As defined in 8.4.2.96	Specifies the precedence level of the TS request. This element is optionally present when dot11EBRActivated is true.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

The DialogToken and STAAddress parameters contain the values from the matching MLME-ADDTS.indication primitive.

If the result code is SUCCESS, the TSPEC and (optional) TCLAS parameters contain the values from the matching MLME-ADDTS-indication.

If the result code is REJECTED_WITH_SUGGESTED_CHANGES or REJECTED_HOME_WITH_SUGGESTED_CHANGES, the TSPEC and TCLAS parameters represent an alternative proposed TS either based on information local to the MAC entity or using additional information received across the SSPN interface. The TS, however, is not created. The TSID and direction values within the TSPEC are as in the matching MLME-ADDTS.indication primitive. The difference may lie in the QoS (e.g., minimum data rate, mean data rate, and delay bound) values, as a result of admission control performed at the SME of the HC on the TS requested to be added (or modified) by the STA. If sufficient bandwidth is not available, the QoS values may be reduced. In one extreme, the minimum data rate, mean data rate, and delay bound may be all set to 0, indicating that no QoS is to be provided to this TS.

If the result code is REJECTED_WITH_SUGGESTED_BSS_TRANSITION, the non-AP STA should initiate a transition query as defined in 10.23.6. Once the transition is completed, the STA should retry TS setup process, as defined in 10.4.4.

6.3.26.5.3 When generated

This primitive is generated by the SME at the HC as a result of an MLME-ADDTS.indication primitive to initiate addition (or modification) of a TS with a specified peer MAC entity or entities.

6.3.26.5.4 Effect of receipt

This primitive approves addition (or modification) of a TS requested by a specified STA's MAC entity, with or without altering the TSPEC.

This primitive causes the MAC entity at the HC to send an ADDTS Response frame in the corresponding QoS Action management frame to the requesting STA containing the specified parameters.

6.3.26.6 MLME-DELTS.request

6.3.26.6.1 Function

This primitive requests deletion of a TS with a specified peer MAC.

This primitive may be generated at either a non-AP STA or the HC.

6.3.26.6.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-DELTS.request(
                        STAAddress,
                        TSInfo,
                        ReasonCode,
                        VendorSpecificInfo
                        )
```

Name	Type	Valid range	Description
STAAddress	MACAddress		Specifies the MAC address of the STA that initiated this TS. Present only at the HC.
TSInfo	TS Info field	As defined in 8.4.2.32	Specifies the TS to be deleted.
ReasonCode	Enumeration	STA_LEAVING, END_TS, UNKNOWN_TS, TIMEOUT, SERVICE_CHANGE_PRECLUDES_TS	Indicates the reason why the TS is being deleted.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.26.6.3 When generated

This primitive is generated by the SME at a STA to initiate deletion of a TS when a higher layer protocol or mechanism signals the STA to initiate such a deletion.

6.3.26.6.4 Effect of receipt

This primitive initiates a TS deletion procedure.

This primitive causes the local MAC entity to send out a DELTS frame containing the specified parameters. If this primitive was generated at the HC, the frame is sent to the specified STA's MAC address. If this primitive was generated at the non-AP STA, the frame is sent to its HC. In either case, the DELTS frame does not solicit a response from the recipient frame other than an acknowledgment to receipt of the frame.

6.3.26.7 MLME-DELTS.indication

6.3.26.7.1 Function

This primitive reports the deletion of a TS by a specified peer MAC entity or deletion of the TS due to an inactivity timeout (HC only).

6.3.26.7.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-DELTS.indication(
                        STAAddress,
                        TSInfo,
                        ReasonCode,
                        VendorSpecificInfo
                        )
```

Name	Type	Valid range	Description
STAAddress	MACAddress		Specifies the MAC address of the STA for which the TS is being deleted. Present only at the HC.
TSInfo	TS Info field	As defined in 8.4.2.32	Specifies the TS information of the TS of concern.
ReasonCode	Enumeration	STA_LEAVING, END_TS, UNKNOWN_TS, TIMEOUT, SERVICE_CHANGE_PRECLUDES_TS	Indicates the reason why the TS is being deleted.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.26.7.3 When generated

This primitive is generated by the MLME when it receives a DELTS frame from the specified peer MAC entity.

This primitive may also be generated by the MLME at the HC as a result of inactivity of a particular TS. Inactivity results when a period equal to the inactivity interval in the TSPEC for the TS elapses

— Without arrival of an MSDU belonging to that TS at the MAC entity of the HC via an MA-UNITDATA.request primitive when the HC is the source STA of that TS or

— Without reception of an MSDU belonging to that TS by the MAC entity of the HC when the HC is not the source STA of that TS.

This primitive is generated after any other state concerning the TSID/direction within the MAC has been destroyed.

6.3.26.7.4 Effect of receipt

The SME is notified of the initiation of a TS deletion by a specified peer MAC entity.

6.3.27 Management of direct links

6.3.27.1 Introduction

This subclause describes the management procedures associated with direct links.

6.3.27.2 MLME-DLS.request

6.3.27.2.1 Function

This primitive requests the setup of a direct link with a specified peer MAC entity.

6.3.27.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-DLS.request(
                              PeerMACAddress,
                              DLSTimeoutValue,
                              DLSResponseTimeout,
                              VendorSpecificInfo
                              )
```

Name	Type	Valid range	Description
PeerMACAddress	MACAddress	Any valid individual MAC address	Specifies the MAC address of the STA that is the intended immediate recipient of the data flow.
DLSTimeoutValue	Integer	≥ 0	Specifies a time limit (in seconds) after which the direct link is terminated if there are no frame exchange sequences with the peer. A value of 0 implies that the direct link is never to be terminated based on a timeout.
DLSResponseTimeout	Integer	≥ 1	Specifies a time limit (in TU) after which the DLS procedure is terminated, from dot11DLSResponseTimeout.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.27.2.3 When generated

This primitive is generated by the SME at a STA to set up a direct link with another STA.

6.3.27.2.4 Effect of receipt

This primitive initiates a DLS procedure. The MLME subsequently issues an MLME-DLS.confirm primitive that reflects the results.

6.3.27.3 MLME-DLS.confirm

6.3.27.3.1 Function

This primitive reports the results of a DLS attempt with a specified peer MAC entity.

6.3.27.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-DLS.confirm(
                              PeerMACAddress,
                              ResultCode,
                              CapabilityInformation,
                              DLSTimeoutValue,
                              SupportedRates,
                              HT Capabilities,
                              VendorSpecificInfo
                              )
```

Name	Type	Valid range	Description
PeerMACAddress	MACAddress	Any valid individual MAC address	Specifies the MAC address of the STA that is the intended immediate recipient of the data flow.
ResultCode	Enumeration	SUCCESS, DLS_NOT_ALLOWED, NOT_PRESENT, NOT_QOS_STA, REFUSED	Indicates the results of the corresponding MLME-DLS.request primitive.
CapabilityInformation	Capability Information field	As defined in 8.4.1.4	Specifies the capabilities of the peer MAC entity.

Name	Type	Valid range	Description
DLSTimeoutValue	Integer	≥ 0	Specifies a time limit (in seconds) after which the direct link is terminated if there are no frame exchange sequences with the peer. A value of 0 implies that the direct link is never to be terminated based on a timeout.
SupportedRates	Set of integers	1–127 inclusive (for each integer in the set)	The set of data rates that are supported by the peer MAC entity.
HT Capabilities	As defined in frame format	As defined in 8.4.2.58	Specifies the parameters within the HT Capabilities element that are supported by the MAC entity. The parameter is optionally present only if dot11HighThroughputOptionImplemented is true.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.27.3.3 When generated

This primitive is generated by the MLME as a result of an MLME-DLS.request primitive to establish a direct link with a specified peer MAC entity.

6.3.27.3.4 Effect of receipt

The SME is notified of the results of the DLS procedure.

6.3.27.4 MLME-DLS.indication

6.3.27.4.1 Function

This primitive reports the establishment of a direct link with a specified peer MAC entity.

6.3.27.4.2 Semantics of the service primitive

The primitive parameters are as follows:

```
  MLME-DLS.indication(
                                    PeerMACAddress,
                                    CapabilityInformation,
                                    DLSTimeoutValue,
                                    SupportedRates,
                                    HT Capabilities,
                                    VendorSpecificInfo
                                    )
```

Name	Type	Valid range	Description
PeerMACAddress	MACAddress	Any valid individual MAC address	Specifies the MAC address of the STA that is the sender of the data flow.
CapabilityInformation	Capability Information field	As defined in 8.4.1.4	Specifies the operational capability definitions to be used by the peer MAC entity.

Name	Type	Valid range	Description
DLSTimeoutValue	Integer	≥ 0	Specifies a time limit (in seconds) after which the direct link is terminated if there are no frame exchange sequences with the peer. A value of 0 implies that the direct link is never to be terminated based on a timeout.
SupportedRates	Set of integers	1–127 inclusive (for each integer in the set)	The set of data rates that are supported by the peer MAC entity.
HT Capabilities	As defined in frame format	As defined in 8.4.2.58	Specifies the parameters within the HT Capabilities element that are supported by the MAC entity. The parameter is optionally present only if dot11HighThroughputOptionImplemented is true.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.27.4.3 When generated

This primitive is generated by the MLME as result of the establishment of a direct link with a specific peer MAC entity that resulted from a DLS procedure that was initiated by that specific peer MAC entity.

6.3.27.4.4 Effect of receipt

The SME is notified of the establishment of the DLS.

6.3.27.5 MLME-DLSTeardown.request

6.3.27.5.1 Function

When initiated by a non-AP STA, this primitive requests the teardown of the direct link with a specified peer MAC entity. When initiated by an AP, this primitive requests the teardown of direct link between two specified MAC entities.

6.3.27.5.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-DLSTeardown.request(
PeerMACAddress1,
PeerMACAddress2,
ReasonCode,
VendorSpecificInfo
)

Name	Type	Valid range	Description
PeerMACAddress1	MACAddress	Any valid individual MAC address	Specifies the MAC address of the STA that is the intended immediate recipient of the teardown.
PeerMACAddress2	MACAddress	Any valid individual MAC address	Specifies the MAC address of the STA with which the DLS is being torn down. This parameter is applicable only at the AP.

Name	Type	Valid range	Description
ReasonCode	Enumeration	STA_LEAVING, END_DLS, PEERKEY_MISMATCH, UNKNOWN_DLS, TIMEOUT, PEER_INITIATED, AP_INITIATED	Indicates the reason why the direct link is being torn down.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.27.5.3 When generated

This primitive is generated by the SME at a STA for tearing down a direct link with another STA.

6.3.27.5.4 Effect of receipt

This primitive initiates a direct-link teardown procedure. The MLME subsequently issues an MLME-DLSTeardown.confirm primitive that reflects the results.

6.3.27.6 MLME-DLSTeardown.indication

6.3.27.6.1 Function

This primitive indicates the teardown of an already established direct link with a specific peer MAC entity.

6.3.27.6.2 Semantics of the service primitive

The primitive parameters are as follows:

```
  MLME-DLSTeardown.indication(
                                        PeerMACAddress,
                                        ReasonCode,
                                        VendorSpecificInfo
                                        )
```

Name	Type	Valid range	Description
PeerMACAddress	MACAddress	Any valid individual MAC address	Specifies the MAC address of the STA that is the sender of the data flow.
ReasonCode	Enumeration	STA_LEAVING, END_DLS, PEERKEY_MISMATCH, UNKNOWN_DLS, TIMEOUT, PEER_INITIATED, AP_INITIATED	Indicates the reason why the direct link is being torn down.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.27.6.3 When generated

This primitive is generated by the MLME as result of the teardown of a direct link with a specific peer MAC entity that resulted from a direct link teardown procedure that was initiated either by that specific peer MAC entity or by the local MAC entity.

6.3.27.6.4 Effect of receipt

The SME is notified of the teardown of the DLS.

6.3.28 Higher layer synchronization support

6.3.28.1 Introduction

This mechanism supports the process of synchronization among higher layer protocol entities residing within different wireless STAs. The actual synchronization mechanism in the higher layer is out of the scope of this standard. In principle, the MLME indicates the transmission/reception of frames with a specific group address in the Address 1 field of a data MPDU.

6.3.28.2 MLME-HL-SYNC.request

6.3.28.2.1 Function

This primitive requests activation of the synchronization support mechanism.

6.3.28.2.2 Semantics of the service primitive

The primitive parameter is as follows:

```
MLME-HL-SYNC.request(
                              GroupAddress
                              )
```

Name	Type	Valid range	Description
GroupAddress	MACAddress	A group MAC address	Specifies the group MAC address to which the synchronization frames are addressed. A synchronization frame is a data frame with higher layer synchronization information.

6.3.28.2.3 When generated

This primitive is generated by the SME when a higher layer protocol initiates a synchronization process.

6.3.28.2.4 Effect of Receipt

This request activates the synchronization support mechanism at the STA. The MLME issues an MLME-HL-SYNC.indication primitive when a higher layer synchronization frame, which is a data frame with the specified group address in Address 1 field, is received or transmitted.

6.3.28.3 MLME-HL-SYNC.indication

6.3.28.3.1 Function

This primitive indicates the last symbol on air of a higher layer synchronization frame, whether transmitted or received by the MAC.

6.3.28.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-HL-SYNC.indication(
                              SourceAddress,
                              SequenceNumber
                              )
```

Name	Type	Valid range	Description
SourceAddress	MACAddress	Any valid individual MAC address	Specifies the SA of the STA that transmitted the higher layer synchronization frame.
SequenceNumber	Integer	As defined in 8.2.4.4.2	Specifies the sequence number of the higher layer synchronization frame received or transmitted.

6.3.28.3.3 When generated

This primitive is generated by the MLME when the successful reception or transmission of a higher layer synchronization frame is detected, as indicated by the PHY_RXEND.indication or PHY_TXEND.confirm primitives generated by the PHY. The higher layer synchronization frame is identified by the group MAC address registered by an earlier MLME-HL-SYNC.request primitive in the Address 1 field of a data frame.

6.3.28.3.4 Effect of Receipt

The SME is notified of the reception or transmission of a higher layer synchronization frame.

6.3.29 Block Ack

6.3.29.1 General

This mechanism supports the initiation (or modification) and termination of Block Ack.

The primitives used for this mechanism are called *Block Ack primitives*, which include MLME-ADDBA.xxx and MLME-DELBA.xxx primitives, where xxx denotes request, confirm, indication, or response. Each primitive contains parameters that correspond to a Block Ack Action frame body. Requests and responses may cause these frames to be sent. Confirms and indications are emitted when an appropriate Block Ack Action frame is received.

6.3.29.2 MLME-ADDBA.request

6.3.29.2.1 Function

This primitive requests the initiation (or modification) of Block Ack with a peer MAC entity.

6.3.29.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-ADDBA.request(
                                PeerSTAAddress,
                                DialogToken,
                                TID,
                                BlockAckPolicy,
                                BufferSize,
                                BlockAckTimeout,
                                ADDBAFailureTimeout,
                                BlockAckStartingSequenceControl,
                                VendorSpecificInfo
                                )
```

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	N/A	Specifies the address of the peer MAC entity with which to perform the Block Ack initiation (or modification).
DialogToken	Integer	0–255	Specifies a number unique to the Block Ack Action primitives and frames used in defining the Block Ack.
TID	Integer	0–15	Specifies the TID of the data.
BlockAckPolicy	Enumeration	Immediate, Delayed	Specifies the Block Ack policy.
BufferSize	Integer	0–127	Specifies the number of MPDUs that can be held in its buffer.
BlockAckTimeout	Integer	0 – 65 535	Specifies the number of TUs without a frame exchange between peers after which the Block Ack is considered to be torn down.
ADDBAFailureTimeout	Integer	Greater than or equal to 1	Specifies a time limit (in TU) after which the Block Ack setup procedure is terminated, from dot11ADDBAResponseTimeout.
BlockAckStartingSequenceControl	Block Ack Starting Sequence Control field	As defined in 8.3.1.8	Specifies the value of Block Ack starting sequence control.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.29.2.3 When generated

This primitive is generated by the SME at a STA to request initiation (or modification) of Block Ack with the specified peer MAC entity.

6.3.29.2.4 Effect of receipt

The STA sends the ADDBA Request frame to the specified peer MAC entity.

6.3.29.3 MLME-ADDBA.confirm

6.3.29.3.1 Function

The primitive reports the results of initiation (or modification) of the Block Ack attempt with the specified peer MAC entity.

6.3.29.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-ADDBA.confirm(
                              PeerSTAAddress,
                              DialogToken,
                              TID,
                              ResultCode,
                              BlockAckPolicy,
                              BufferSize,
                              BlockAckTimeout,
                              VendorSpecificInfo
                              )
```

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	N/A	Specifies the address of the peer MAC entity with which the Block Ack initiation (or modification) was attempted. This value matches the PeerSTAAddress parameter specified in MLME-ADDBA.request primitive.
DialogToken	Integer	0–255	Specifies a number unique to the Block Ack Action primitives and frames used in defining the Block Ack. This value matches the DialogToken parameter specified in MLME-ADDBA.request primitive.
TID	Integer	0–15	Specifies the TID of the data. This value matches the TID specified in MLME-ADDBA.request primitive.
ResultCode	Enumeration	SUCCESS, INVALID_ PARAMETERS, REFUSED	Indicates the result of the corresponding MLME-ADDBA.request primitive.
BlockAckPolicy	Enumeration	Immediate, Delayed	Specifies the Block Ack policy.
BufferSize	Integer	0–127	Specifies the maximum number of MPDUs in the block for the specified TID.
BlockAckTimeout	Integer	0 – 65 535	Specifies the number of TUs without a frame exchange between peers after which the Block Ack is considered to be torn down.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.29.3.3 When generated

This primitive is generated by the MLME as a result of an MLME-ADDBA.request primitive to indicate the results of that request.

6.3.29.3.4 Effect of receipt

The SME is notified of the results of the Block Ack initiation (or modification).

6.3.29.4 MLME-ADDBA.indication

6.3.29.4.1 Function

This primitive reports the initiation (or modification) of Block Ack by a peer MAC entity.

6.3.29.4.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-ADDBA.indication(
                              PeerSTAAddress,
                              DialogToken,
                              TID,
                              BlockAckPolicy,
                              BufferSize,
                              BlockAckTimeout,
                              VendorSpecificInfo
                              )
```

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	N/A	Specifies the address of the peer MAC entity that requested the Block Ack initiation (or modification).
DialogToken	Integer	0–255	Specifies a number unique to the Block Ack Action primitives and frames used in defining the Block Ack.
TID	Integer	0–15	Specifies the TID of the data.
BlockAckPolicy	Enumeration	Immediate, Delayed	Specifies the Block Ack policy.
BufferSize	Integer	0–127	Specifies the number of MPDUs that can be held in peerSTAAddress buffer.
BlockAckTimeout	Integer	0 – 65 535	Specifies the number of TUs without a frame exchange between peers after which the Block Ack is considered to be torn down.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.29.4.3 When generated

This primitive is generated by the MLME as a result of receipt of a Block Ack initiation (or modification) by the specified peer MAC entity in the form of an ADDBA Request frame.

6.3.29.4.4 Effect of receipt

The SME is notified of the initiation (or modification) of the Block Ack by the specified peer MAC entity.

6.3.29.5 MLME-ADDBA.response

6.3.29.5.1 Function

The primitive responds to the initiation (or modification) by a specified peer MAC entity.

6.3.29.5.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-ADDBA.response(
                                PeerSTAAddress,
                                DialogToken,
                                TID,
                                ResultCode,
                                BlockAckPolicy,
                                BufferSize,
                                BlockAckTimeout,
                                VendorSpecificInfo
                                )
```

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	N/A	Specifies the address of the peer MAC entity that attempted the Block Ack initiation (or modification). This value matches the PeerSTAAddress parameter specified in MLME-ADDBA.indication primitive.

Name	Type	Valid range	Description
DialogToken	Integer	0–255	Specifies a number unique to the Block Ack Action primitives and frames used in defining the Block Ack. This value matches the DialogToken parameter specified in MLME-ADDBA.indication primitive.
TID	Integer	0–15	Specifies the TID of the data. This value matches the TID specified in MLME-ADDBA.indication primitive.
ResultCode	Enumeration	SUCCESS, REFUSED, INVALID_ PARAMETERS	Indicates the result of the corresponding MLME-ADDBA.indication primitive.
BlockAckPolicy	Enumeration	Immediate, Delayed	Specifies the Block Ack policy. Undefined when the result code is REFUSED.
BufferSize	Integer	0–127	Specifies the maximum number of MPDUs in the block for the specified TID. Undefined when the result code is REFUSED.
BlockAckTimeout	Integer	0 – 65 535	Specifies the number of TUs without a frame exchange between peers after which the Block Ack is considered to be torn down.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.29.5.3 When generated

This primitive is generated by the MLME as a result of an MLME-ADDBA.indication primitive to initiate Block Ack by the specified peer MAC entity.

6.3.29.5.4 Effect of receipt

The primitive causes the MAC entity to send an ADDBA Response frame to the specified peer MAC entity.

6.3.29.6 MLME-DELBA.request

6.3.29.6.1 Function

This primitive requests the deletion of Block Ack with a peer MAC entity.

6.3.29.6.2 Semantics of the service primitive

The primitive parameters are as follows:

```
  MLME-DELBA.request(
                              PeerSTAAddress,
                              Direction,
                              TID,
                              ReasonCode,
                              VendorSpecificInfo
                              )
```

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	N/A	Specifies the address of the peer MAC entity with which to perform the Block Ack deletion.

Name	Type	Valid range	Description
Direction	Enumeration	Originator, Recipient	Specifies if the MAC entity initiating the MLME-DELBA.request primitive is the originator or the recipient of the data stream that uses the Block Ack.
TID	Integer	0–15	Specifies the TID of the MSDUs for which this Block Ack has been set up.
ReasonCode	Enumeration	STA_LEAVING, END_BA, UNKNOWN_BA, TIMEOUT	Indicates the reason why the Block Ack is being deleted.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.29.6.3 When generated

This primitive is generated by the SME at a STA to request deletion of Block Ack with the specified peer MAC entity.

6.3.29.6.4 Effect of receipt

The STA sends the DELBA frame to the specified peer MAC entity.

6.3.29.7 MLME-DELBA.indication

6.3.29.7.1 Function

This primitive reports the deletion of Block Ack by a peer MAC entity.

6.3.29.7.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-DELBA.indication(
                                PeerSTAAddress,
                                Direction,
                                TID,
                                ReasonCode,
                                VendorSpecificInfo
                                )
```

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	N/A	Specifies the address of the peer MAC entity with which to perform the Block Ack deletion.
Direction	Enumeration	Originator, Recipient	Specifies if the MAC entity initiating the MLME-DELBA.request primitive is the originator or the recipient of the data stream that uses the Block Ack.
TID	Integer	0–15	Specifies the TID of the MSDUs for which this Block Ack has been set up.
ReasonCode	Enumeration	STA_LEAVING, END_BA, UNKNOWN_BA, TIMEOUT	Indicates the reason why the Block Ack is being deleted.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.29.7.3 When generated

This primitive is generated by the MLME as a result of receipt of a Block Ack deletion by the specified peer MAC entity in the form of a DELBA frame.

6.3.29.7.4 Effect of receipt

The SME is notified of the deletion of the Block Ack by the specified peer MAC entity.

6.3.30 Schedule element management

6.3.30.1 Introduction

This subclause describes the management procedures associated with the QoS Schedule element.

The primitives defined are MLME-SCHEDULE.request and MLME-SCHEDULE.indication.

6.3.30.2 MLME-SCHEDULE.request

6.3.30.2.1 Function

This primitive requests transmission of a Schedule frame. It is valid at the HC.

6.3.30.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
   MLME-SCHEDULE.request(
                                        STAAddress,
                                        Schedule
                                        )
```

Name	Type	Valid range	Description
STAAddress	MACAddress	Any valid individual address	MAC address of the STA to which the Schedule frame shall be sent.
Schedule	Schedule element	As defined in 8.4.2.36	Specifies the schedule for the STA, including the SI (minimum and maximum), TXOP duration (minimum and maximum), and specification interval.

6.3.30.2.3 When generated

This primitive is generated by the SME at the HC to send the schedule information, in the form of a Schedule frame, to a specified STA when the schedule information for the STA is changed.

6.3.30.2.4 Effect of receipt

This primitive causes the MAC entity at the HC to send a Schedule frame to the STA specified in the primitive containing the specified Schedule parameters.

6.3.30.3 MLME-SCHEDULE.indication

6.3.30.3.1 Function

This primitive reports the reception of a new schedule by the STA in the form of a Schedule frame.

6.3.30.3.2 Semantics of the service primitive

The primitive parameter is as follows:

```
MLME-SCHEDULE.indication(
                                        Schedule
                                        )
```

Name	Type	Valid range	Description
Schedule	Schedule element	As defined in 8.4.2.36	Specifies the schedule for the STA, including the SI (minimum and maximum), TXOP duration (minimum and maximum), and specification interval.

6.3.30.3.3 When generated

This primitive is generated by the MLME as a result of receipt of a new schedule in the form of a Schedule frame.

6.3.30.3.4 Effect of receipt

The SME is notified of the receipt of QoS schedule in the form of a Schedule frame. The new Schedule parameters overwrite the previously stored values.

6.3.31 Vendor-specific action

6.3.31.1 Introduction

This set of primitives supports the signaling of (Protected) Vendor Specific Action frames among peer SMEs.

6.3.31.2 MLME-VSPECIFIC.request

6.3.31.2.1 Function

This primitive requests transmission of a Vendor Specific Action frame.

6.3.31.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-VSPECIFIC.request(
                                        PeerMACAddress,
                                        Protected,
                                        Organization Identifier,
                                        VendorSpecificContent
                                        )
```

Name	Type	Valid range	Description
PeerMACAddress	MACAddress	Any valid individual MAC address or any valid group MAC address	The address of the peer MAC entity or group of entities to which the Vendor Specific Action frame is sent.

Name	Type	Valid range	Description
Protected	Boolean	true, false	Specifies whether the request is sent using a Robust Management frame. If true, the request is sent using a Protected Vendor Specific frame. If false, the request is sent using a Vendor Specific frame.
Organization Identifier	As defined in 8.4.1.31	As defined in 8.4.1.31	Contains a public value assigned by the IEEE to identify the organization that has defined the content of the particular vendor-specific action.
VendorSpecificContent	Determined by the entity to whom the Organization Identifier is registered	Determined by the entity to whom the Organization Identifier is registered	Vendor-specific content.

6.3.31.2.3 When generated

This primitive is generated by the SME to request that a Vendor Specific Action frame be sent.

6.3.31.2.4 Effect of receipt

On receipt of this primitive, the MLME constructs a Vendor Specific Action frame containing the set of elements and vendor-specific fields. The STA then attempts to transmit the frame.

6.3.31.3 MLME-VSPECIFIC.indication

6.3.31.3.1 Function

This primitive indicates that a Vendor Specific Action frame has been received from a peer entity.

6.3.31.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-VSPECIFIC.indication(
                              PeerMACAddress,
                              Protected,
                              Organization Identifier,
                              RCPI,
                              VendorSpecificContent
                              )
```

Name	Type	Valid range	Description
PeerMACAddress	MACAddress	Any valid individual MAC address	The address of the peer MAC entity from which the Vendor Specific Action frame was received.

Name	Type	Valid range	Description
Protected	Boolean	true, false	Specifies whether the request was received using a Robust Management frame. If true, the request was received using a Protected Vendor Specific frame. If false, the request was received using a Vendor Specific frame.
Organization Identifier	As defined in 8.4.1.31	As defined in 8.4.1.31	Contains a public value assigned by the IEEE to identify the organization that has defined the content of the particular vendor-specific action.
RCPI	As defined in 8.4.2.40	As defined in 8.4.2.40	Present when dot11OCBActivated is true. RCPI is the measured value of received channel power on the received Vendor Specific Action frame.
VendorSpecificContent	Determined by the entity to whom the Organization Identifier is registered	Determined by the entity to whom the Organization Identifier is registered	Vendor-specific content.

6.3.31.3.3 When generated

This primitive is generated by the MLME when a valid Vendor Specific Action frame is received.

6.3.31.3.4 Effect of receipt

On receipt of this primitive, the Vendor Specific Content can be made available for SME processes.

6.3.32 Neighbor report request

6.3.32.1 General

The following MLME primitives support the signaling of neighbor report requests.

6.3.32.2 MLME-NEIGHBORREPREQ.request

6.3.32.2.1 Function

This primitive requests that a Neighbor Report Request frame be sent to the AP with which the STA is associated. It is valid only at a Radio Measurement capable STA.

6.3.32.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-NEIGHBORREPREQ.request(
                            DialogToken,
                            SSID,
                            VendorSpecificInfo
                            )
```

Name	Type	Valid range	Description
DialogToken	Integer	1–255	The Dialog Token to identify the neighbor report transaction.
SSID	As defined in the SSID element	As defined in the SSID element	Optional SSID element to request a neighbor list for a specific SSID.
Vendor-SpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.32.2.3 When generated

This primitive is generated by the SME to request that a Neighbor Report Request frame be sent to the AP with which the STA is associated to request a neighbor report.

6.3.32.2.4 Effect of receipt

On receipt of this primitive, the MLME constructs a Neighbor Report Request frame. The STA then attempts to transmit this to the AP with which it is associated.

6.3.32.3 MLME-NEIGHBORREPREQ.indication

6.3.32.3.1 Function

This primitive indicates that a Neighbor Report Request frame was received. It is valid only at a Radio Measurement capable AP.

6.3.32.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-NEIGHBORREPREQ.indication(
                                   PeerSTAAddress,
                                   DialogToken,
                                   SSID,
                                   VendorSpecificInfo
                                   )
```

Name	Type	Valid range	Description
PeerSTA-Address	MACAddress	Any valid individual MAC address	The address of the STA's MAC entity from which a Neighbor Report Request frame was received.
DialogToken	Integer	1–255	The Dialog Token in the Neighbor Report Request frame that was received.
SSID	As defined in the SSID element	As defined in the SSID element	Optional SSID element to request a neighbor list for a specific SSID.
Vendor-SpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.32.3.3 When generated

This primitive is generated by the MLME when a valid Neighbor Report Request frame is received.

6.3.32.3.4 Effect of receipt

On receipt of this primitive, the SME operates according to the procedure in 10.11.10.3.

6.3.33 Neighbor report response

6.3.33.1 General

The following MLME primitives support the signaling of neighbor report responses.

6.3.33.2 MLME-NEIGHBORREPRESP.request

6.3.33.2.1 Function

This primitive requests that a neighbor report response be sent. This may be in response to an MLME-NEIGHBORREPREQ.indication primitive or an autonomous request. It is valid only at a Radio Measurement capable AP.

6.3.33.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-NEIGHBORREPRESP.request(
                              PeerSTAAddress,
                              DialogToken,
                              NeighborListSet,
                              VendorSpecificInfo
                              )
```

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC address	The address of the STA's MAC entity to which a Neighbor Report Response frame is to be sent.
DialogToken	Integer	0–255	The Dialog Token to identify the neighbor report transaction. Set to the value received in the corresponding MLME-NEIGHBORREPREQ.indication primitive or to 0 for an autonomous report.
NeighborListSet	Set of Neighbor List elements each as defined in the Neighbor Report element format	As defined in 8.4.2.39	A set of Neighbor List elements, each representing a neighboring AP being reported as defined in the Neighbor Report element format.
Vendor-SpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.33.2.3 When generated

This primitive is generated by the SME to request a neighbor report be sent. This may be in response to an earlier MLME-NEIGHBORREPREQ.indication primitive or a request to transmit an autonomous report.

6.3.33.2.4 Effect of receipt

On receipt of this primitive, the MLME constructs a Neighbor Report Response frame. The STA then attempts to transmit this to the STA indicated by the PeerSTAAddress parameter.

6.3.33.3 MLME-NEIGHBORREPRESP.indication

6.3.33.3.1 Function

This primitive indicates that a neighbor report response has been received. This may be in response to an earlier neighbor report request (MLME-NEIGHBORREPREQ.request) or an autonomous report. It is valid only at a Radio Measurement capable STA.

6.3.33.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
   MLME-NEIGHBORREPRESP.indication(
                                   PeerSTAAddress,
                                   DialogToken,
                                   NeighborListSet,
                                   VendorSpecificInfo
                                   )
```

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC address	The address of the AP from which the Neighbor Report Response frame was received.
DialogToken	Integer	0–255	The Dialog Token received in the Neighbor Report Response frame to identify the neighbor report transaction.
NeighborListSet	Set of Neighbor List elements, each as defined in the Neighbor Report element format	As defined in 8.4.2.39	A set of Neighbor List elements derived from the MIB table dot11RMNeighborReportTable, each representing a neighboring AP being reported as defined in the Neighbor Report element format.
Vendor-SpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.33.3.3 When generated

This primitive is generated by the MLME when a valid Neighbor Report Response frame is received.

6.3.33.3.4 Effect of receipt

On receipt of this primitive, neighbor report data may be available to the SME.

6.3.34 Link Measure Request

6.3.34.1 General

The following primitives support the measurement of link path loss and the estimation of link margin between peer entities.

6.3.34.2 MLME-LINKMEASURE.request

6.3.34.2.1 Function

This primitive supports the measurement of link path loss and the estimation of link margin between peer entities.

NOTE—The layer management model used assumes that the handling of a received Link Measurement Request frame is entirely within the MLME. Correspondingly there are no MLME-SME primitives specified for the peer side of a link measurement request transaction.

6.3.34.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-LINKMEASURE.request(
                              PeerMACAddress,
                              DialogToken,
                              Transmit Power,
                              Max Transmit Power,
                              VendorSpecificInfo
                              )
```

Name	Type	Valid range	Description
PeerMAC Address	MACAddress	Any valid individual MAC address	The address of the peer MAC entity to which the Link Measure Request shall be sent.
DialogToken	Integer	1–255	The dialog token to identify the Link Measure transaction.
Transmit Power	Integer	As defined in 8.5.7.4	The transmit power to be used when transmitting the Link Measurement Request frame and included in the frame body. See 8.5.7.4.
Max Transmit Power	Integer	As defined in 8.5.7.4	The maximum transmit power to be used by the transmitting STA on its operating channel. See 8.5.7.5.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.34.2.3 When generated

This primitive is generated by the SME to request that a Link Measurement Request frame be sent to the peer entity to request that entity to report transmit power and link margin information.

6.3.34.2.4 Effect of receipt

On receipt of this primitive, the MLME constructs a Link Measurement Request frame. The STA then attempts to transmit this to the STA indicated in the PeerMACAddress parameter.

6.3.34.3 MLME-LINKMEASURE.confirm

6.3.34.3.1 Function

This primitive reports the result of a Link Measurement request.

6.3.34.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-LINKMEASURE.confirm(
                              ResultCode,
                              DialogToken,
                              TransmitPower,
                              LinkMargin,
                              RCPI.request,
                              RSNI.request,
                              RCPI.report,
                              RSNI.report,
                              ReceiveAntennaID,
                              TransmitAntennaID,
                              VendorSpecificInfo
                              )
```

Name	Type	Valid range	Description
ResultCode	Enumeration	SUCCESS, UNSPECIFIED FAILURE	Indicates the result of the corresponding MLME-LINKMEASURE.request primitive.
DialogToken	Integer	As defined in the corresponding MLME-LINK-MEASURE.request primitive	The Dialog Token to identify the link measurement transaction.
TransmitPower	As defined in the TPC Report element	As defined in the TPC Report element	The contents of the Transmit Power field of the received Link Measurement Report frame. Present only if ResultCode = SUCCESS.
LinkMargin	As defined in the TPC Report element	As defined in the TPC Report element	The contents of the Link Margin field of the received Link Measurement Report frame. Present only if ResultCode = SUCCESS.
RCPI.request	Integer	As defined in 16.4.8.6, or 18.3.10.7, or 17.4.8.6	The RCPI level of the corresponding Link Measurement Request frame received at the reporting STA. Present only if ResultCode = SUCCESS.
RSNI.request	Integer	As defined in 8.4.2.43	The RSNI of the corresponding Link Measurement Request frame received at the reporting STA. Present only if ResultCode = SUCCESS
RCPI.report	Integer	As defined in 16.4.8.6, or 18.3.10.7, or 17.4.8.6	The RCPI level of the corresponding Link Measurement Report frame received at the requesting STA. Present only if ResultCode = SUCCESS.
RSNI.report	Integer	As defined in 8.4.2.43	The RSNI of the corresponding Link Measurement Report frame received at the requesting STA. Present only if ResultCode = SUCCESS
Receive Antenna ID	Integer	0–255	The Antenna ID corresponding to the antenna on which the Link Measurement Request frame was received at the reporting STA. Antenna ID is defined in 8.4.2.31.
Transmit Antenna ID	Integer	0–255	The Antenna ID corresponding to the antenna used to transmit the Link Measurement Report frame. Antenna ID is defined in 8.4.2.31.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.34.3.3 When generated

This primitive is generated by the MLME when a valid Link Measurement Report frame is received from the requested STA.

6.3.34.3.4 Effect of receipt

On receipt of this primitive, the SME evaluates the ResultCode and may use the reported data.

6.3.35 MLME SAP interface for resource request

6.3.35.1 MLME-RESOURCE-REQUEST.request

6.3.35.1.1 Function

This primitive is used to perform the over-the-air resource request of an FT Resource Request Protocol. The over-the-air resource request is performed using Authentication frames, with an authentication algorithm of FT authentication and transaction sequence number of 3 or 4.

6.3.35.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-RESOURCE-REQUEST.request(
                                        PeerMACAddress,
                                        Contents of FT Authentication elements
                                        )
```

Name	Type	Valid range	Description
PeerMACAddress	MACAddress	Any valid individual MAC address	Specifies the MAC address of the AP that is the intended immediate recipient of the resource request.
Content of FT Authentication elements	Sequence of elements	As defined in 12.8	The set of elements to be included in the FT Confirm frame, as described in 12.8.4.

6.3.35.1.3 When generated

This primitive is generated by the SME to send the third frame of the over-the-air FT Resource Request Protocol. The third frame is an Authentication frame, with an authentication algorithm of FT authentication and transaction sequence value of 3.

6.3.35.1.4 Effect of receipt

Upon receipt of this primitive, the MLME constructs the appropriate Authentication frame and causes it to be transmitted to the peer MAC address.

6.3.35.2 MLME-RESOURCE-REQUEST.indication

6.3.35.2.1 Function

This primitive is used to enact the security and QoS resource request with a specified peer MAC entity.

6.3.35.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-RESOURCE-REQUEST.indication(
                                  PeerMACAddress,
                                  Content of FT Authentication elements
                                  )
```

Name	Type	Valid range	Description
PeerMACAddress	MACAddress	Any valid individual MAC address	Specifies the MAC address of the STA that was the sender of the resource request.
Content of FT Authentication elements	Sequence of elements	As defined in 12.8	The set of elements included in the FT Confirm frame, as described in 12.8.4.

6.3.35.2.3 When generated

This primitive is generated by the MLME at an AP to indicate that the third frame of the over-the-air FT Resource Request Protocol has been received. The third frame is an Authentication frame, with an authentication algorithm of FT authentication and transaction sequence value of 3.

6.3.35.2.4 Effect of receipt

Upon receipt of this primitive, the SME examines the Transition element and RSNE contents and responds to the peer MAC address using the MLME-RESOURCE-REQUEST.response primitive.

6.3.35.3 MLME-RESOURCE-REQUEST.response

6.3.35.3.1 Function

This primitive is used to enact the security and QoS resource request protocol with a specified peer MAC entity.

6.3.35.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-RESOURCE-REQUEST.response(
                                PeerMACAddress,
                                Content of FT Authentication elements
                                )
```

Name	Type	Valid range	Description
PeerMACAddress	MACAddress	Any valid individual MAC address	Specifies the MAC address of the STA that is the intended immediate recipient of the resource response.
Content of FT Authentication elements	Sequence of elements	As defined in 12.8	The set of elements to be included in the FT Ack frame, as described in 12.8.5. This includes an optional response to a resource request (RIC).

6.3.35.3.3 When generated

This primitive is generated by the SME at an AP to cause the transmission of the fourth frame in the over-the-air FT Resource Request Protocol. The fourth frame is an Authentication frame, with an authentication algorithm of FT authentication and transaction sequence value of 4.

6.3.35.3.4 Effect of receipt

Upon receipt of this primitive, the MLME constructs the appropriate Authentication frame and causes it to be transmitted to the peer MAC address.

6.3.35.4 MLME-RESOURCE-REQUEST.confirm

6.3.35.4.1 Function

This primitive is used to enact the security and QoS resource request protocol with a specified peer MAC entity.

6.3.35.4.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-RESOURCE-REQUEST.confirm(
                                        PeerMACAddress,
                                        Content of FT Authentication elements
                                        )
```

Name	Type	Valid range	Description
PeerMACAddress	MACAddress	Any valid individual MAC address	Specifies the MAC address of the AP that was the sender of the resource response.
Content of FT Authentication elements	Sequence of elements	As defined in 12.8	The set of elements included in the FT Ack frame, as described in 12.8.5. This includes an optional response to a resource request (RIC).

6.3.35.4.3 When generated

This primitive is generated by the MLME on receipt of the fourth frame in the FT Resource Request Protocol.

6.3.35.4.4 Effect of receipt

Upon receipt of this primitive, the SME examines the content of the message and completes its processing of the resource request.

6.3.35.5 MLME-RESOURCE-REQUEST-LOCAL.request

6.3.35.5.1 Function

This primitive is used to enact the over-the-DS FT Resource Request Protocol for a specified peer MAC entity. The over-the-DS FT Resource Request Protocol is performed by communication between the STA and the SME of the target AP, bypassing the MAC of the target AP. This MLME function is used to allow the MAC of the target AP to process the resource requests.

6.3.35.5.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-RESOURCE-REQUEST-LOCAL.request(
                                        MACAddress,
                                        Content of Resource Descriptor(s)
                                        )
```

Name	Type	Valid range	Description
MACAddress	MACAddress	Any valid individual MAC address	Specifies the MAC address of the STA that is making the resource request.
Content of Resource Descriptor(s)	Sequence of elements	As defined in 12.11.2	Specifies the resource(s) that are being requested.

6.3.35.5.3 When generated

This primitive is generated by the SME at a target AP upon receiving an over-the-DS resource request to request resources within the local MAC.

6.3.35.5.4 Effect of receipt

Upon receipt of this primitive, the MAC checks for resource availability and allocates resources as requested.

6.3.35.6 MLME-RESOURCE-REQUEST-LOCAL.confirm

6.3.35.6.1 Function

This primitive is used to respond to a local resource request for resources from the SME.

6.3.35.6.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-RESOURCE-REQUEST-LOCAL.confirm(
                                                  MACAddress,
                                                  Content of Resource Descriptor(s),
                                                  ResultCode
                                                  )
```

Name	Type	Valid range	Description
MACAddress	MACAddress	Any valid individual MAC address	Specifies the MAC address of the STA that is making the resource request.
Content of Resource Descriptor(s)	Sequence of elements	As defined in 12.11.2	Specifies the resource(s) that were allocated or could have been allocated.
ResultCode	Enumeration	SUCCESS, REFUSED, UNSPECIFIED FAILURE	Indicates the result of the outcome of a resource request.

6.3.35.6.3 When generated

This primitive is generated by the MAC in response to a local resource request for resources via MLME-RESOURCE-REQUEST-LOCAL.request primitive.

6.3.35.6.4 Effect of receipt

Upon receipt of this primitive, the SME prepares a success or failure response to be sent to the STA via the current AP.

6.3.36 MLME SAP interface for remote requests

6.3.36.1 MLME-REMOTE-REQUEST.request

6.3.36.1.1 Function

This primitive is used by the SME of a non-AP STA (to send over-the-DS requests) and the SME of an AP (to send over-the-DS responses) to request the MAC to send an FT Action frame.

6.3.36.1.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-REMOTE-REQUEST.request(
PeerMACAddress,
Content of FT Action Frame
)

Name	Type	Valid range	Description
PeerMACAddress	MACAddress	Any valid individual MAC address	Specifies the MAC address of the STA that is the destination of the Action frame
Content of FT Action Frame	Sequence of octets	As defined in 8.5.9	The Action frame to send to the STA.

6.3.36.1.3 When generated

This primitive is generated by the SME to send an FT Action frame to a specific peer MAC entity.

6.3.36.1.4 Effect of receipt

Upon receipt of this primitive, the MAC forwards the Action frame to the STA identified in the Action frame.

6.3.36.2 MLME-REMOTE-REQUEST.indication

6.3.36.2.1 Function

This primitive is used by the MAC to indicate to the SME the reception of an FT Action frame.

6.3.36.2.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-REMOTE-REQUEST.indication(
PeerMACAddress,
Contents of FT Action Frame
)

Name	Type	Valid range	Description
PeerMACAddress	MACAddress	Any valid individual MAC address	Specifies the MAC address of the STA that issued the Action frame.
Content of FT Action Frame	Sequence of octets	As defined in 8.5.9	The Action frame received from the STA.

6.3.36.2.3 When generated

This primitive is generated by the MAC as a result of the receipt of an FT Action frame from a specific peer MAC entity.

6.3.36.2.4 Effect of receipt

Upon receipt of this primitive, the remote request broker (RRB) in the SME of the current AP forwards the Action frame to the target AP identified in the Action frame.

6.3.37 Extended channel switch announcement

6.3.37.1 General

The following MLME primitives support the signaling of extended channel switch announcement.

6.3.37.2 MLME-EXTCHANNELSWITCH.request

6.3.37.2.1 Function

This primitive requests that a (Protected) Extended Channel Switch Announcement frame be sent by an AP or mesh STA in an MBSS.

6.3.37.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-EXTCHANNELSWITCH.request(
                                    Mode,
                                    OperatingClass,
                                    ChannelNumber,
                                    ChannelSwitchCount,
                                    Protected,
                                    Mesh Channel Switch Parameters,
                                    VendorSpecificInfo
                                    )
```

Name	Type	Valid range	Description
Mode	Integer	0,1	Channel switch mode, as defined for the Extended Channel Switch Announcement element.
OperatingClass	Integer	As defined in Annex E	Specifies the new operating class.
ChannelNumber	Integer	As defined in Annex E	Specifies the new channel number.
ChannelSwitchCount	As defined in 8.4.2.55	As defined in 8.4.2.55	Specifies the time period until the channel switch event, as described in 8.4.2.55
Protected	Boolean	true, false	Specifies whether the request is sent using a Robust Management frame. If true, the request is sent using the Protected Extended Channel Switch Announcement frame. If false, the request is sent using the Extended Channel Switch Announcement frame.

Name	Type	Valid range	Description
Mesh Channel Switch Parameters	As defined in 8.4.2.105	As defined in 8.4.2.105	Specifies MBSS Channel Switch Parameters used by a mesh STA. This parameter is present if the dot11MeshActivated is true; otherwise, the parameter is not present.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.37.2.3 When generated

This primitive is generated by the STA management entity (SME) to request that a (Protected) Extended Channel Switch Announcement frame be sent to a STA that is associated to the AP or to peer mesh STAs in the MBSS.

6.3.37.2.4 Effect of receipt

On receipt of this primitive, the MLME constructs and transmits a (Protected) Extended Channel Switch Announcement frame.

6.3.37.3 MLME-EXTCHANNELSWITCH.confirm

6.3.37.3.1 Function

This primitive reports the result of a request to switch channel.

6.3.37.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-EXTCHANNELSWITCH.confirm(
                                ResultCode,
                                VendorSpecificInfo
                                )
```

Name	Type	Valid range	Description
ResultCode	Enumeration	SUCCESS, UNSPECIFIED_FAILURE	Reports the result of an extended channel switch request.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.37.3.3 When generated

This primitive is generated by the MLME when an extended channel switch request completes. Possible unspecified failure causes include an inability to schedule an extended channel switch announcement.

6.3.37.3.4 Effect of receipt

The SME is notified of the results of the extended channel switch procedure.

6.3.37.4 MLME-EXTCHANNELSWITCH.indication

6.3.37.4.1 Function

This primitive indicates that a (Protected) Extended Channel Switch Announcement frame was received from an AP or from a peer mesh STA.

6.3.37.4.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-EXTCHANNELSWITCH.indication(
                                        Peer MAC Address,
                                        Mode,
                                        OperatingClass,
                                        ChannelNumber,
                                        ChannelSwitchCount,
                                        Protected,
                                        Mesh Channel Switch Parameters,
                                        VendorSpecificInfo
                                        )
```

Name	Type	Valid range	Description
Peer MAC Address	MACAddress	Any valid individual MAC address	The address of the peer MAC entity from which the Extended Channel Switch Announcement frame was received.
Mode	Integer	0, 1	Channel switch mode, as defined for the Channel Switch Announcement element.
OperatingClass	Integer	As defined in Annex E	Specifies the new operating class.
ChannelNumber	Integer	As defined in Annex E	Specifies the new channel number.
ChannelSwitchCount	As defined in 8.4.2.55	As defined in 8.4.2.55	Specifies the time period until the channel switch event, as described in 8.4.2.55
Protected	Boolean	true, false	Specifies whether the request was received using a Robust Management frame. If true, the request was received using the Protected Extended Channel Switch Announcement frame. If false, the request was received using the Extended Channel Switch Announcement frame.
Mesh Channel Switch Parameters	As defined in 8.4.2.105	As defined in 8.4.2.105	Specifies MBSS Channel Switch Parameters used by a mesh STA. This parameter is present if the dot11MeshActivated is true; otherwise, the parameter is not present.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.37.4.3 When generated

This primitive is generated by the MLME when a valid (Protected) Extended Channel Switch Announcement frame is received.

6.3.37.4.4 Effect of receipt

On receipt of this primitive, the SME decides whether to accept the switch request.

6.3.37.5 MLME-EXTCHANNELSWITCH.response

6.3.37.5.1 Function

This primitive is used to schedule an accepted extended channel switch.

6.3.37.5.2 Semantics of the service primitive

The primitive parameters are as follows:

```
   MLME-EXTCHANNELSWITCH.response(
                                                    Mode,
                                                    OperatingClass,
                                                    ChannelNumber,
                                                    ChannelSwitchCount,
                                                    Mesh Channel Switch Parameters,
                                                    VendorSpecificInfo
                                                    )
```

Name	Type	Valid range	Description
Mode	Integer	0, 1	Channel switch mode, as defined for the Channel Switch Announcement element.
OperatingClass	Integer	As defined in Annex E	Specifies the new operating class.
ChannelNumber	Integer	As defined in Annex E	Specifies the new channel number.
ChannelSwitchCount	As defined in 8.4.2.55	As defined in 8.4.2.55	Specifies the time period until the channel switch event, as described in 8.4.2.55
Mesh Channel Switch Parameters	As defined in 8.4.2.105	As defined in 8.4.2.105	Specifies MBSS Channel Switch Parameters used by a mesh STA. This parameter is present if the dot11MeshActivated is true; otherwise, the parameter is not present.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.37.5.3 When generated

This primitive is generated by the SME to schedule an accepted extended channel switch request.

6.3.37.5.4 Effect of receipt

On receipt of this primitive, the MLME schedules the extended channel switch.

6.3.38 DSE power constraint announcement

6.3.38.1 General

The following MLME primitives support the signaling of DSE power constraint to dependent STAs.

6.3.38.2 MLME-DSETPC.request

6.3.38.2.1 Function

This primitive requests that a (Protected) DSE Power Constraint frame be sent by an enabling STA.

6.3.38.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-DSETPC.request(
                        RequesterSTAAddress,
                        ResponderSTAAddress,
                        DSELocalPowerConstraint,
                        Protected,
                        VendorSpecificInfo
                        )
```

Name	Type	Valid range	Description
RequesterSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the MAC entity of the enabling STA.
ResponderSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the MAC entity that initiates the enablement process.
DSELocalPowerConstraint	Integer	0–255	Specifies the local power constraint, as described in the DSE Power Constraint frame (see 8.5.8.10).
Protected	Boolean	true, false	Specifies whether the request is sent using a Robust Management frame. If true, the request is sent using the Protected DSE Power Constraint frame. If false, the request is sent using the DSE Power Constraint frame.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.38.2.3 When generated

This primitive is generated by the SME to request that a (Protected) DSE Power Constraint Announcement frame be sent to a dependent STA.

6.3.38.2.4 Effect of receipt

Upon receipt of this primitive, the MLME constructs a (Protected) DSE Power Constraint Announcement frame. The enabling STA then schedules this frame for transmission.

6.3.38.3 MLME-DSETPC.confirm

6.3.38.3.1 Function

This primitive reports the results of a request to send a (Protected) DSE Power Constraint Announcement frame.

6.3.38.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-DSETPC.confirm(
                        ResultCode,
                        VendorSpecificInfo
                        )
```

Name	Type	Valid range	Description
ResultCode	Enumeration	SUCCESS, REFUSED	Indicates the result of MLME-DSETPC.request primitive.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.38.3.3 When generated

This primitive is generated by the MLME when a DSE power constraint announcement completes.

6.3.38.3.4 Effect of receipt

The SME is notified of the results of the DSE power constraint procedure.

6.3.38.4 MLME-DSETPC.indication

6.3.38.4.1 Function

This primitive indicates that a DSE Power Constraint Announcement frame was received from an enabling STA.

6.3.38.4.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-DSETPC.indication(
                        RequesterSTAAddress,
                        ResponderSTAAddress,
                        DSELocalPowerConstraint,
                        Protected,
                        VendorSpecificInfo
                        )
```

Name	Type	Valid range	Description
RequesterSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity that initiated the enablement process.
ResponderSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity that is the enabling STA.
DSELocalPowerConstraint	Integer	0–255	Specifies the local power constraint, as described in the DSE Power Constraint frame (see 8.5.8.10).

Name	Type	Valid range	Description
Protected	Boolean	true, false	Specifies whether the request was received using a Robust Management frame. If true, the request was received using the Protected DSE Power Constraint frame. If false, the request was received using the DSE Power Constraint frame.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.38.4.3 When generated

This primitive is generated by the MLME when a valid (Protected) DSE Power Constraint Announcement frame is received.

6.3.38.4.4 Effect of receipt

On receipt of this primitive, the SME performs the DSE power constraint procedure (see 10.12.5).

6.3.38.5 MLME-DSETPC.response

6.3.38.5.1 Function

This primitive is used to report the result of the DSE power constraint procedure.

6.3.38.5.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-DSETPC.response(
                        RequesterSTAAddress,
                        ResponderSTAAddress,
                        ResultCode,
                        Protected,
                        VendorSpecificInfo
                        )
```

Name	Type	Valid range	Description
RequesterSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the MAC entity of the enabling STA.
ResponderSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity that initiates the enabling process.
ResultCode	Enumeration	SUCCESS, REFUSED	Reports the result of a DSE power constraint procedure.
Protected	Boolean	true, false	Specifies whether the response is sent using a Robust Management frame. If true, the response is sent using the Protected DSE Power Constraint frame. If false, the response is sent using the DSE Power Constraint frame.

Name	Type	Valid range	Description
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.38.5.3 When generated

This primitive is generated by the SME to schedule a response to DSE power constraint announcement.

6.3.38.5.4 Effect of receipt

On receipt of this primitive, the MLME schedules the transmission of a (Protected) DSE power constraint result to the enabling STA that sent the DSE power constraint announcement.

6.3.39 Enablement

6.3.39.1 General

This mechanism supports the process of establishing an enablement relationship with a peer MAC entity.

6.3.39.2 MLME-ENABLEMENT.request

6.3.39.2.1 Function

This primitive requests enablement with a specified peer MAC entity.

6.3.39.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-ENABLEMENT.request(
                              RequesterSTAAddress,
                              ResponderSTAAddress,
                              EnablementTimeLimit,
                              Protected,
                              VendorSpecificInfo
                              )
```

Name	Type	Valid range	Description
RequesterSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the MAC entity that initiates the enablement process.
ResponderSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the MAC entity of the enabling STA.
EnablementTimeLimit	Integer	> 0	Specifies a time limit (in TU) after which the enablement process is terminated.
Protected	Boolean	true, false	Specifies whether the request is sent using a Robust Management frame. If true, the request is sent using the Protected DSE Enablement frame. If false, the request is sent using the DSE Enablement frame.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.39.2.3 When generated

This primitive is generated by the SME for a STA to establish enablement with a specified peer MAC entity. During the enablement procedure, the SME can generate additional MLME-ENABLEMENT.request primitives.

6.3.39.2.4 Effect of receipt

This primitive initiates an enablement procedure. The MLME subsequently issues a MLME-ENABLEMENT.confirm primitive that reflects the results.

6.3.39.3 MLME-ENABLEMENT.confirm

6.3.39.3.1 Function

This primitive reports the results of an enablement attempt with a specified peer MAC entity.

6.3.39.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-ENABLEMENT.confirm(
                              RequesterSTAAddress,
                              ResponderSTAAddress,
                              ResultCode,
                              Protected,
                              EnablementIdentifier,
                              VendorSpecificInfo
                              )
```

Name	Type	Valid range	Description
RequesterSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the MAC entity that initiated the enablement process. This value matches the RequesterSTAAddress parameter specified in the corresponding MLME-ENABLEMENT.request primitive.
ResponderSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peerMAC entity with which the enablement process was attempted.This value matches the ResponderSTAAddress parameter specified in the corresponding MLME-ENABLEMENT.request primitive.
ResultCode	Enumeration	SUCCESS, REFUSED, INVALID_PARAMETERS, TOO_MANY_SIMULTANEOUS_REQUESTS	Indicates the result of MLME-ENABLEMENT.request primitive.
EnablementIdentifier	Integer	0 – 65 535	Specifies the dependent enablement identifier.

Name	Type	Valid range	Description
Protected	Boolean	true, false	Specifies whether the response was received using a Robust Management frame. If true, the response was received using the Protected DSE Enablement frame. If false, the response was received using the DSE Enablement frame.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.39.3.3 When generated

This primitive is generated by the MLME as a result of an MLME-ENABLEMENT.request primitive for enablement with a specified peer MAC entity.

6.3.39.3.4 Effect of receipt

The SME is notified of the results of the enablement procedure.

6.3.39.4 MLME-ENABLEMENT.indication

6.3.39.4.1 Function

This primitive indicates receipt of a request from a specific peer MAC entity to establish an enablement relationship with the STA processing this primitive.

6.3.39.4.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-ENABLEMENT.indication(
                                   RequesterSTAAddress,
                                   ResponderSTAAddress,
                                   Protected,
                                   VendorSpecificInfo
                                   )
```

Name	Type	Valid range	Description
RequesterSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity that initiated the enablement process.
ResponderSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity that is the enabling STA.
Protected	Boolean	true, false	Specifies whether the request was sent using a Robust Management frame. If true, the request was sent using the Protected DSE Enablement frame. If false, the request was sent using the DSE Enablement frame.

Name	Type	Valid range	Description
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.39.4.3 When generated

This primitive is generated by the MLME as a result of the receipt of an enablement request from a specific peer MAC entity.

6.3.39.4.4 Effect of receipt

The SME is notified of the receipt of this enablement request.

6.3.39.5 MLME-ENABLEMENT.response

6.3.39.5.1 Function

This primitive is used to send a response to a specified peer MAC entity that requested enablement with the STA that issued this primitive.

6.3.39.5.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-ENABLEMENT.response(
                              RequesterSTAAddress,
                              ResponderSTAAddress,
                              ResultCode,
                              EnablementIdentifier,
                              Protected,
                              VendorSpecificInfo
                              )
```

Name	Type	Valid range	Description
RequesterSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the MAC entity that initiated the enablement process.
ResponderSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peerMAC entity that is the enabling STA.
ResultCode	Enumeration	SUCCESS, REFUSED, INVALID_PARAMETERS, TOO_MANY_SIMULTANEOUS_REQUESTS	Indicates the result response to the enablement request from the peer MAC entity.
EnablementIdentifier	Integer	0-65535	Specifies the dependent enablement identifier.
Protected	Boolean	true, false	Specifies whether the response is sent using a Robust Management frame. If true, the response is sent using the Protected DSE Enablement frame. If false, the response is sent using the DSE Enablement frame.

Name	Type	Valid range	Description
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.39.5.3 When generated

This primitive is generated by the SME of a STA as a response to an MLME-ENABLEMENT.indication primitive.

6.3.39.5.4 Effect of receipt

This primitive initiates transmission of a response to the specific peer MAC entity that requested enablement.

6.3.40 Deenablement

6.3.40.1 MLME-DEENABLEMENT.request

6.3.40.1.1 Function

This primitive requests that the enablement relationship with a specified peer MAC entity be invalidated.

6.3.40.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```
   MLME-DEENABLEMENT.request(
                                        RequesterSTAAddress,
                                        ResponderSTAAddress,
                                        ReasonCode,
                                        Protected,
                                        VendorSpecificInfo
                                        )
```

Name	Type	Valid range	Description
RequesterSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity that requests the deenablement process.
ResponderSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the MAC entity that becomes deenabled in the process.
ReasonCode	Reason Result Code field	As defined in 8.5.8.5	Specifies the reason code for initiating the deenablement process.
Protected	Boolean	true, false	Specifies whether the request is sent using a Robust Management frame. If true, the request is sent using the Protected DSE Deenablement frame. If false, the request is sent using the DSE Deenablement frame.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.40.1.3 When generated

This primitive is generated by the SME for a STA to invalidate enablement with a specified peer MAC entity in order to prevent the exchange of (Protected dual of) Public Action frames between the two STAs. During the deenablement procedure, the SME can generate additional MLME-DEENABLEMENT.request primitives.

6.3.40.1.4 Effect of receipt

This primitive initiates a deenablement procedure.

6.3.40.2 MLME-DEENABLEMENT.indication

6.3.40.2.1 Function

This primitive reports the invalidation of an enablement relationship with a specified peer MAC entity.

6.3.40.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-DEENABLEMENT.indication(
                                        RequesterSTAAddress,
                                        ResponderSTAAddress,
                                        ReasonCode,
                                        Protected,
                                        VendorSpecificInfo
                                        )
```

Name	Type	Valid range	Description
RequesterSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity with which the enablement relationship was invalidated.
ResponderSTAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the MAC entity with which the enablement relationship was invalidated.
ReasonCode	Reason Result Code field	As defined in 8.5.8.5	Specifies the reason the deenablement procedure was initiated.
Protected	Boolean	true, false	Specifies whether the request was received using a Robust Management frame. If true, the request was received using the Protected DSE Deenablement frame. If false, the request was received using the DSE Deenablement frame.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.40.2.3 When generated

This primitive is generated by the MLME as a result of the invalidation of an enablement relationship with a specific peer MAC entity.

6.3.40.2.4 Effect of receipt

The SME is notified of the invalidation of the specific enablement relationship.

6.3.41 SA Query support

6.3.41.1 MLME-SAQuery.request

6.3.41.1.1 Function

This primitive requests that a SA Query Request frame be sent to a specified peer STA to which the STA is associated.

6.3.41.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-SAQuery.request(
                    PeerSTAAddress,
                    TransactionIdentifier
                    )
```

Name	Type	Valid range	Description
PeerSTA Address	MAC Address	Any valid individual MAC Address	Specifies the address of the peer MAC entity for the SA Query
TransactionIdentifier	2 octets	As defined in 8.5.10.2	The Transaction Identifier to identify the SA Query Request and Response transaction

6.3.41.1.3 When generated

This primitive is generated by the SME to request that a SA Query Request frame be sent to a specified peer STA with which the STA is associated.

6.3.41.1.4 Effect of receipt

On receipt of this primitive, the MLME constructs a SA Query Request frame. The STA then attempts to transmit this to the peer STA with which it is associated.

6.3.41.2 MLME-SAQuery.confirm

6.3.41.2.1 Function

This primitive reports the result of a SA Query procedure.

6.3.41.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-SAQuery.confirm(
                    PeerSTAAddress,
                    TransactionIdentifier
                    )
```

Name	Type	Valid Range	Description
PeerSTA Address	MAC Address	Any valid individual MAC Address	Specifies the address of the peer MAC entity for the SA Query
TransactionIdentifier	2 octets	As defined in 8.5.10.2	The Transaction Identifier to identify the SA Query Request and Response transaction

6.3.41.2.3 When generated

This primitive is generated by the MLME as a result of the receipt of a valid SA Query Response frame.

6.3.41.2.4 Effect of receipt

On receipt of this primitive, the SME may use the response as a sign of liveness of the peer STA.

6.3.41.3 MLME-SAQuery.indication

6.3.41.3.1 Function

This primitive indicates that a SA Query Request frame was received from a STA.

6.3.41.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-SAQuery.indication(
                         PeerSTAAddress,
                         TransactionIdentifier
                         )
```

Name	Type	Valid range	Description
PeerSTA Address	MAC Address	Any valid individual MAC Address	Specifies the address of the peer MAC entity for the SA Query
TransactionIdentifier	2 octets	As defined in 8.5.10.2	The Transaction Identifier to identify the SA Query Request and Response transaction

6.3.41.3.3 When generated

This primitive is generated by the MLME when a valid SA Query Request frame is received.

6.3.41.3.4 Effect of receipt

On receipt of this primitive, the SME operates according to the procedure in 10.3.

6.3.41.4 MLME-SAQuery.response

6.3.41.4.1 Function

This primitive is generated in response to an MLME-SAQuery.indication requesting a SA Query Response frame be sent to a STA.

6.3.41.4.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-SAQuery.response(
                        PeerSTAAddress,
                        TransactionIdentifier
                        )
```

Name	Type	Valid range	Description
PeerSTA Address	MAC Address	Any valid individual MAC Address	Specifies the address of the peer MAC entity for the SA Query
TransactionIdentifier	2 octets	As defined in 8.5.10.2	The Transaction Identifier to identify the SA Query Request and Response transaction

6.3.41.4.3 When generated

This primitive is generated by the SME, in response to an MLME-SAQuery.indication primitive, requesting a SA Query Response frame be sent to a STA.

6.3.41.4.4 Effect of receipt

On receipt of this primitive, the MLME constructs a SA Query Response frame. The STA then attempts to transmit this to the STA indicated by the PeerSTAAddress parameter.

6.3.42 Get TSF timer

6.3.42.1 General

This mechanism is used to request the current value of the TSF timer that the STA maintains.

6.3.42.2 MLME-GETTSFTIME.request

6.3.42.2.1 Function

This primitive is generated by the SME to request that the MLME returns the value of its TSF timer. The value returned in TSFtime (as specified in 6.3.42.2.2) is the value of the TSF timer at the instant the MLME-GETTSFTIME.request primitive is received.

6.3.42.2.2 Semantics of the service primitive

This primitive has no parameters.

6.3.42.2.3 When generated

This primitive is generated by the SME to request the value of the TSF timer from the MLME.

6.3.42.2.4 Effect of receipt

The MLME issues an MLME-GETTSFTIME.confirm.

6.3.42.2 MLME-GETTSFTIME.confirm

6.3.42.2.1 Function

This primitive is generated by the MLME to report to the SME the result of a request to get the value of the TSF timer.

6.3.42.2.2 Semantics of the service primitive

This primitive uses the following parameters:

```
   MLME-GETTSFTIME.confirm(
                              ResultCode,
                              TSFtime
                              )
```

Name	Type	Valid range	Description
ResultCode	Enumeration	SUCCESS, FAILURE	Reports the outcome of an MLME-GETTSFTIME.request primitive
TSFtime	Integer	0 – (2^{64} –1)	The value of the TSF timer. Present only if ResultCode is SUCCESS.

6.3.42.2.3 When generated

This primitive is generated by the MLME to report to the SME the result of an MLME-GETTSFTIME.request.

6.3.42.2.4 Effect of receipt

The SME is notified of the result of an MLME-GETTSFTIME.request primitive and, if successful, has the value of the TSF timer at the instant the MLME-GETTSFTIME.request was received by the MLME. If the result of an MLME-GETTSFTIME.request is failure, the TSFtime parameter is not included in the MLME-GETTSFTIME.confirm primitive.

NOTE—The TSF timer value can be used, along with other information, by the SME to compute an offset between an external time standard such as a version of Universal Coordinated Time (UTC) from a Global Positioning System (GPS) unit and the TSF timer.

6.3.43 Timing Advertisement

6.3.43.1 General

The Timing Advertisement primitives are used to communicate timing and other information from the higher layers or the SME of one STA to the higher layers or SME of other STAs.

6.3.43.2 MLME-TIMING_ADVERTISEMENT.request

6.3.43.2.1 Function

This primitive is generated by the SME to request that the MLME generate a Timing Advertisement frame to transmit timing and, optionally, higher layer information.

6.3.43.2.2 Semantics of the service primitive

This primitive provides the following parameters:

```
MLME-TIMING_ADVERTISEMENT.request(
                                   PeerMACAddress,
                                   Capability Information,
                                   Country,
                                   Power Constraint,
                                   Time Advertisement,
                                   Extended Capabilities,
                                   VendorSpecificInfo
                                   )
```

Name	Type	Valid range	Description
PeerMACAddress	MACAddress	Any valid individual or group MAC address	The address of the peer MAC entity or group of entities to which the Timing Advertisement frame is sent.
Capability Information	As defined in 8.4.1.4	As defined in 8.4.1.4	The announced capabilities of the STA.
Country	As defined in 8.4.2.10	As defined in 8.4.2.10	The information required to identify the regulatory domain in which the STA is located and to configure its PHY for operation in that regulatory domain. Present only when TPC functionality is required, as specified in 10.8 or when dot11MultiDomainCapabilityActivated is true.
Power Constraint	As defined in 8.4.2.16	As defined in 8.4.2.16	Optional. The Power Constraint element contains the information necessary to allow a STA to determine the local maximum transmit power in the current channel.
Time Advertisement	As defined in 8.4.2.63	As defined in 8.4.2.63	Timing announced by the STA.
Extended Capabilities	As defined in 8.4.2.29	As defined in 8.4.2.29	Optional. The Extended Capabilities element may be present if any of the fields in this element are nonzero.
Vendor-SpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.43.2.3 When generated

This primitive is generated by the SME to request that the MLME generates a Timing Advertisement frame for transmission.

6.3.43.2.4 Effect of receipt

Upon the receipt of this primitive, the MLME attempts to transmit a Timing Advertisement frame to the specified MAC address, using the procedures defined in 10.21.

6.3.43.3 MLME-TIMING_ADVERTISEMENT.indication

6.3.43.3.1 Function

This primitive is generated by the MLME to indicate to the SME the reception of a Timing Advertisement frame.

6.3.43.3.2 Semantics of the service primitive

This primitive provides the following parameters:

```
MLME-TIMING_ADVERTISEMENT.indication(
                                      Timestamp,
                                      Capability Information,
                                      Local Time,
                                      Country,
                                      Power Constraint,
                                      Time Advertisement,
                                      Extended Capabilities,
                                      RCPI,
                                      Source MAC address,
                                      VendorSpecificInfo
                                      )
```

Name	Type	Valid range	Description
Timestamp	Integer	N/A	The timestamp of the received frame.
Capability Information	As defined in 8.4.1.4	As defined in 8.4.1.4	The announced capabilities of the STA.
Local Time	Integer	N/A	Local Time is the value of a station's TSF timer at the start of reception of the first octet of the timestamp field of the received Timing Advertisement frame.
Country	As defined in 8.4.2.10	As defined in 8.4.2.10	The information required to identify the regulatory domain in which the STA is located and to configure its PHY for operation in that regulatory domain. Present only when TPC functionality is required, as specified in 10.8 or when dot11MultiDomainCapabilityActivated is true.
Power Constraint	As defined in 8.4.2.16	As defined in 8.4.2.16	The Power Constraint element contains the information necessary to allow a STA to determine the local maximum transmit power in the current channel.
Time Advertisement	As defined in 8.4.2.63	As defined in 8.4.2.63	Timing announced by the STA.
Extended Capabilities	As defined in 8.4.2.29	As defined in 8.4.2.29	The Extended Capabilities element may be present if any of the fields in this element are nonzero.
RCPI	Integer as defined in 8.4.2.40	As defined in 8.4.2.40	RCPI is the measured value of received channel power on the received Timing Advertisement frame.
Source MAC Address	As defined in 8.2.4.3.6	As defined in 8.2.4.3.6	The SA field of the MAC header from the received Timing Advertisement frame.
Vendor-SpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.43.3.3 When generated

This primitive is generated by the MLME when a Timing Advertisement frame is received.

6.3.43.3.4 Effect of receipt

Upon the receipt of this primitive, the SME is notified that a Timing Advertisement frame has been received.

6.3.44 TDLS Discovery

6.3.44.1 General

The following MLME primitives support the signaling of TDLS Discovery.

6.3.44.2 MLME-TDLSDISCOVERY.request

6.3.44.2.1 Function

This primitive requests that a TDLS Discovery Request frame be sent through the AP path.

6.3.44.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-TDLSDISCOVERY.request(
                              DestinationAddress,
                              TDLSDiscoveryRequest
                              )
```

Name	Type	Valid range	Description
DestinationAddress	MAC Address	Any valid individual MAC Address	Specifies the DA to which a TDLS Discovery Request frame is transmitted.
TDLSDiscoveryRequest	Sequence of octets	As defined in TDLS Discovery Request frame	Specifies the proposed service parameters for the TDLS Discovery Request frame.

6.3.44.2.3 When generated

This primitive is generated by the SME to request that a TDLS Discovery Request frame be sent through the AP.

6.3.44.2.4 Effect of receipt

On receipt of this primitive, the MLME constructs a TDLS Discovery Request frame. The STA then attempts to transmit this frame.

6.3.44.3 MLME-TDLSDISCOVERY.confirm

6.3.44.3.1 Function

This primitive is generated when a valid TDLS Discovery Response frame is received.

6.3.44.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-TDLSDISCOVERY.confirm(
                              TDLSPeerSTAAddress,
                              TDLSDiscoveryResponse
                              )
```

Name	Type	Valid range	Description
TDLSPeerSTAAddress	MAC Address	Any valid individual MAC Address	Specifies the MAC address of the TDLS peer STA from which a TDLS Discovery Response frame was received.
TDLSDiscoveryResponse	Sequence of octets	As defined in TDLS Discovery Response frame	Specifies the service parameters contained in the received TDLS Discovery Response frame.

6.3.44.3.3 When generated

This primitive is generated when a valid TDLS Discovery Response frame is received.

6.3.44.3.4 Effect of receipt

On receipt of this primitive, the SME evaluates the MLME-TDLSDISCOVERY.confirm primitive and may use the reported data.

6.3.44.4 MLME-TDLSDISCOVERY.indication

6.3.44.4.1 Function

This primitive indicates that a TDLS Discovery Request frame was received from a TDLS peer STA.

6.3.44.4.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-TDLSDISCOVERY.indication(
                                        TDLSPeerSTAAddress,
                                        TDLSDiscoveryRequest
                                        )
```

Name	Type	Valid range	Description
TDLSPeerSTAAddress	MACAddress	Any valid individual MAC Address	Specifies the MAC address of the TDLS peer STA from which a TDLS Discovery Request frame was received.
TDLSDiscoveryRequest	Sequence of octets	As defined in TDLS Discovery Request frame	Specifies the proposed service parameters of the TDLS Discovery Request frame.

6.3.44.4.3 When generated

This primitive is generated by the MLME when a valid TDLS Discovery Request frame is received.

6.3.44.4.4 Effect of receipt

On receipt of this primitive, the SME operates according to the procedure in 10.22.

6.3.44.5 MLME-TDLSDISCOVERY.response

6.3.44.5.1 Function

This primitive requests that a TDLS Discovery Response frame be sent directly to the TDLS peer STA from which a TDLS Discovery Request frame was received.

6.3.44.5.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-TDLSDISCOVERY.response(
                                    TDLSPeerSTAAddress,
                                    TDLSDiscoveryResponse
                                    )
```

Name	Type	Valid range	Description
TDLSPeerSTAAddress	MAC Address	Any valid individual MAC Address	Specifies the MAC address of the TDLS peer STA to which a TDLS Discovery Response frame is transmitted.
TDLSDiscoveryResponse	Sequence of octets	As defined in TDLS Discovery Response frame	Specifies the proposed service parameters for the TDLS Discovery Response frame.

6.3.44.5.3 When generated

This primitive is generated by the SME to request that a TDLS Discovery Response frame be sent to the TDLS peer STA from which a TDLS Discovery Request frame was received.

6.3.44.5.4 Effect of receipt

On receipt of this primitive, the MLME constructs a TDLS Discovery Response frame. The STA then attempts to transmit this frame to the TDLS peer STA.

6.3.45 TDLS direct-link establishment

6.3.45.1 General

The following MLME primitives support the signaling of tunneled direct-link setup. Figure 6-7 depicts the TDLS direct-link establishment process. The figure is only an example of the basic procedure and is not meant to be exhaustive of all possible uses of the protocol.

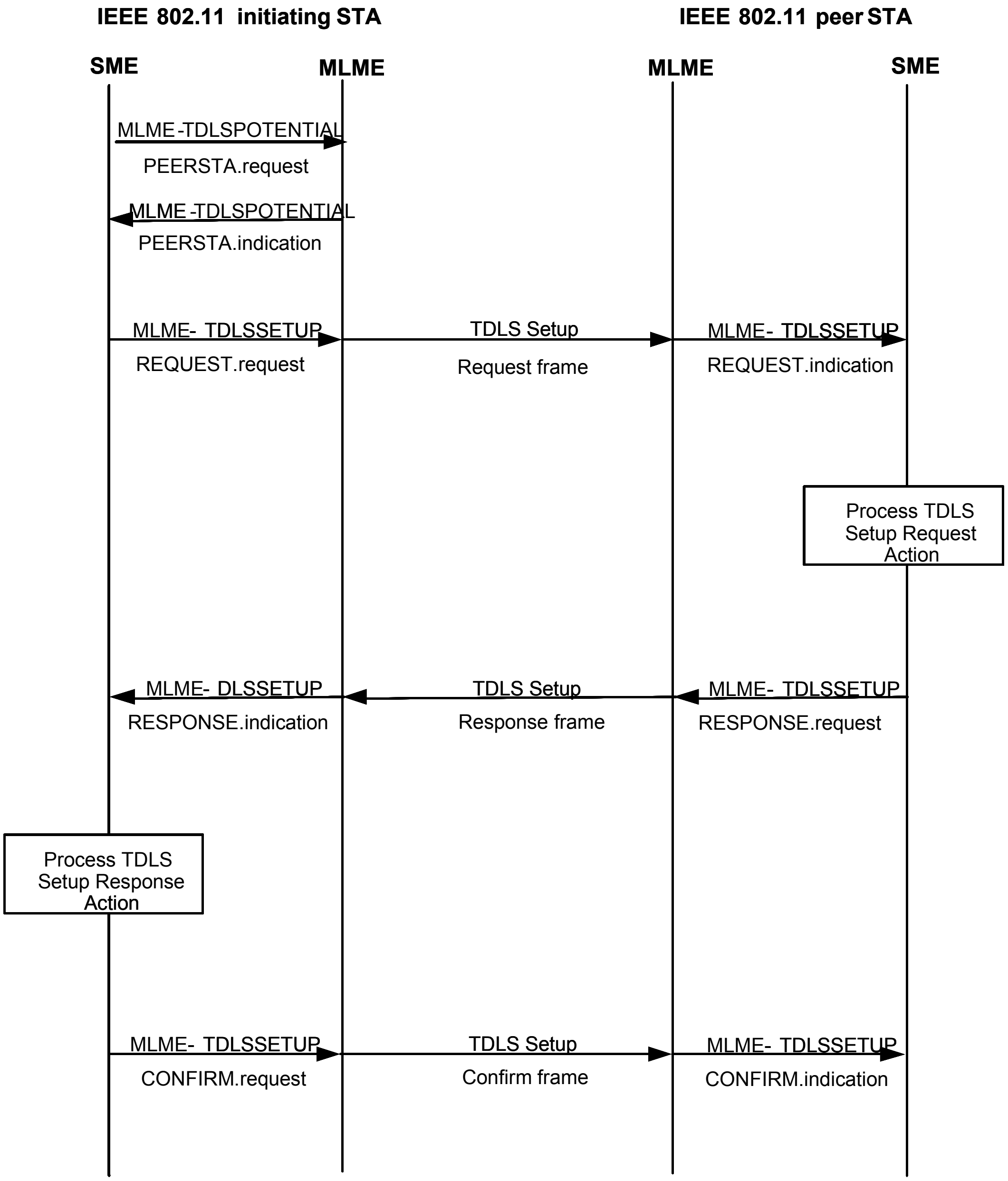


Figure 6-7—TDLS direct-link establishment

6.3.45.2 MLME-TDLSSETUPREQUEST.request

6.3.45.2.1 Function

This primitive requests that a TDLS Setup Request frame be sent to a candidate TDLS responder STA.

6.3.45.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-TDLSSETUPREQUEST.request(
                                          TDLSResponderAddress,
                                          TDLSSetupRequest
                                          )
```

Name	Type	Valid range	Description
TDLSResponderAddress	MAC Address	Any valid individual MAC Address	Specifies the MAC address of the STA to which the TDLS Setup Request frame is transmitted.
TDLSSetupRequest	Sequence of octets	As defined in TDLS Setup Request frame	Specifies the proposed service parameters for the TDLS Setup.

6.3.45.2.3 When generated

This primitive is generated by the SME to request that a TDLS Setup Request frame be sent to a candidate TDLS responder STA.

6.3.45.2.4 Effect of receipt

On receipt of this primitive, the MLME constructs a TDLS Setup Request frame. The STA then attempts to transmit this frame to the candidate TDLS responder STA.

6.3.45.3 MLME-TDLSSETUPREQUEST.indication

6.3.45.3.1 Function

This primitive indicates that a TDLS Setup Request frame was received.

6.3.45.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-TDLSSETUPREQUEST.indication(
                                          TDLSInitiatorAddress,
                                          TDLSSetupRequest
                                          )
```

Name	Type	Valid range	Description
TDLSInitiatorAddress	MACAddress	Any valid individual MAC Address	Specifies the MAC address of the TDLS initiator STA from which a TDLS Setup Request frame was received.
TDLSSetupRequest	Sequence of octets	As defined in TDLS Setup Request frame	Specifies the proposed service parameters for the TDLS Setup.

6.3.45.3.3 When generated

This primitive is generated by the MLME when a valid TDLS Setup Request frame is received.

6.3.45.3.4 Effect of receipt

On receipt of this primitive, the SME operates according to the procedure in 10.22.

6.3.45.4 MLME-TDLSSETUPRESPONSE.request

6.3.45.4.1 Function

This primitive requests that a TDLS Setup Response frame be sent to the TDLS initiator STA.

6.3.45.4.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-TDLSSETUPRESPONSE.request(
                               TDLSInitiatorAddress,
                               TDLSSetupResponse
                               )
```

Name	Type	Valid range	Description
TDLSInitiatorAddress	MAC Address	Any valid individual MAC Address	Specifies the MAC address of the TDLS initiator STA to which a TDLS Setup Response frame is transmitted.
TDLSSetupResponse	Sequence of octets	As defined in TDLS Setup Response frame	Specifies the proposed service parameters for the TDLS Setup.

6.3.45.4.3 When generated

This primitive is generated by the SME to request that a TDLS Setup Response frame be sent to the TDLS initiator STA.

6.3.45.4.4 Effect of receipt

On receipt of this primitive, the MLME constructs a TDLS Setup Response frame. The STA then attempts to transmit this to the TDLS initiator STA.

6.3.45.5 MLME-TDLSSETUPRESPONSE.indication

6.3.45.5.1 Function

This primitive indicates that a TDLS Setup Response frame was received from the TDLS responder STA.

6.3.45.5.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-TDLSSETUPRESPONSE.indication(
                               TDLSResponderAddress,
                               TDLSSetupResponse
                               )
```

Name	Type	Valid range	Description
TDLSResponderAddress	MACAddress	Any valid individual MAC Address	Specifies the MAC address of the TDLS responder STA from which a TDLS Setup Response frame was received.
TDLSSetupResponse	Sequence of octets	As defined in TDLS Setup Response frame	Specifies the proposed service parameters for the TDLS Setup.

6.3.45.5.3 When generated

This primitive is generated by the MLME when a valid TDLS Setup Response frame is received.

6.3.45.5.4 Effect of receipt

On receipt of this primitive, the SME operates according to the procedure in 10.22.

6.3.45.6 MLME-TDLSSETUPCONFIRM.request

6.3.45.6.1 Function

This primitive requests that a TDLS Setup Confirm frame be sent to the TDLS responder STA.

6.3.45.6.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-TDLSSETUPCONFIRM.request(
                                        TDLSResponderAddress,
                                        TDLSSetupConfirm
                                        )
```

Name	Type	Valid range	Description
TDLSResponderAddress	MAC Address	Any valid individual MAC Address	Specifies the MAC address of the TDLS responder STA to which a TDLS Setup Confirm frame is transmitted.
TDLSSetupConfirm	Sequence of octets	As defined in TDLS Setup Confirm frame	Specifies the proposed service parameters for the TDLS Setup.

6.3.45.6.3 When generated

This primitive is generated by the SME to request that a TDLS Setup Confirm frame be sent to the TDLS responder STA.

6.3.45.6.4 Effect of receipt

On receipt of this primitive, the MLME constructs a TDLS Setup Confirm frame. The STA then attempts to transmit this to the TDLS responder STA.

6.3.45.7 MLME-TDLSSETUPCONFIRM.indication

6.3.45.7.1 Function

This primitive indicates that a TDLS Setup Confirm frame was received from the TDLS initiator STA.

6.3.45.7.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-TDLSSETUPCONFIRM.indication(
                                 TDLSInitiatorAddress,
                                 TDLSSetupConfirm
                                 )
```

Name	Type	Valid range	Description
TDLSInitiatorAddress	MACAddress	Any valid individual MAC Address	Specifies the MAC address of the TDLS initiator STA from which a TDLS Setup Confirm frame was received.
TDLSSetupConfirm	Sequence of octets	As defined in TDLS Setup Confirm frame	Specifies the proposed service parameters for the TDLS setup.

6.3.45.7.3 When generated

This primitive is generated by the MLME when a valid TDLS Setup Confirm frame is received.

6.3.45.7.4 Effect of receipt

On receipt of this primitive, the SME operates according to the procedure in 10.22.

6.3.45.8 MLME-TDLSPOTENTIALPEERSTA.request

6.3.45.8.1 Function

This primitive requests information about a potential TDLS peer STA.

6.3.45.8.2 Semantics of the service primitive

The primitive parameter is as follows:

```
MLME-TDLSPOTENTIALPEERSTA.request(
                                 MACAddress
                                 )
```

Name	Type	Valid Range	Description
MACAddress	MAC Address	Any valid individual MAC Address	Specifies the MAC address of the potential TDLS peer STA.

6.3.45.8.3 When generated

This primitive is generated by the SME to request the MLME to provide information about a potential TDLS peer STA.

6.3.45.8.4 Effect of receipt

On receipt of this primitive, the MLME responds with the requested information about the identified STA.

6.3.45.9 MLME-TDLSPOTENTIALPEERSTA.confirm

6.3.45.9.1 Function

This primitive informs the SME about a potential TDLS peer STA.

6.3.45.9.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-TDLSPOTENTIALPEERSTA.confirm(
                                            MACAddress,
                                            RSSI,
                                            VendorSpecificInfo
                                            )
```

Name	Type	Valid range	Description
MACAddress	MAC Address	Any valid individual MAC Address	Specifies the MAC address of the STA for which information is requested.
RSSI	Integer	–1-RSSI Max	Specifies the RSSI from the STA. –1 indicates that the STA is not present.
VendorSpecificInfo	Vendor Specific	Vendor Specific	Specifies vendor-specific information about the STA identified in the MACAddress field.

6.3.45.9.3 When generated

This primitive is generated by the MLME to indicate to the SME that a potential TDLS peer STA has been detected.

6.3.45.9.4 Effect of receipt

On receipt of this primitive, the SME may attempt to set up a TDLS direct link by issuing an MLME-TDLSSETUPREQUEST.request primitive to the MLME.

6.3.46 TDLS direct-link teardown

6.3.46.1 General

The following MLME primitives support the signaling of tunneled direct-link setup. Figure 6-8 depicts the TDLS direct-link teardown process. The figure is only an example of the basic procedure and is not meant to be exhaustive of all possible uses of the protocol.

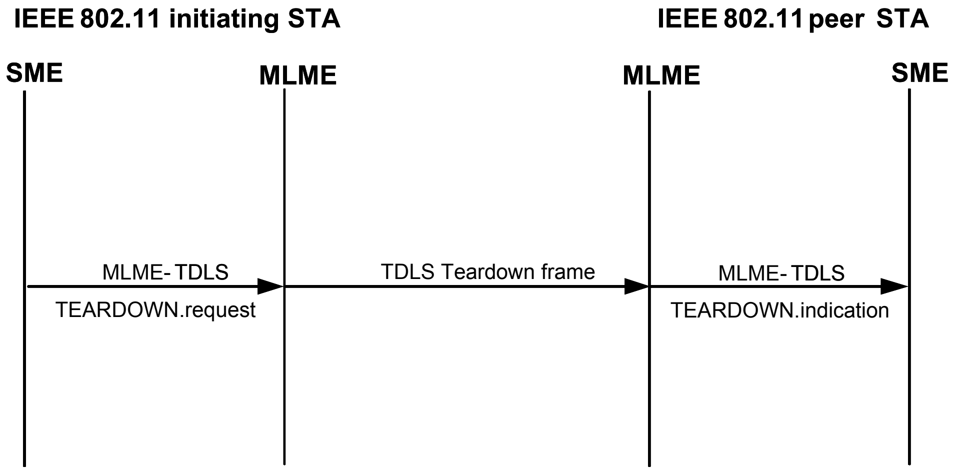


Figure 6-8—TDLS direct-link teardown

6.3.46.2 MLME-TDLSTEARDOWN.request

6.3.46.2.1 Function

This primitive requests that a TDLS Teardown frame be sent to the TDLS peer STA.

6.3.46.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
          MLME-TDLSTEARDOWN.request(
                                        TDLSPeerSTAAddress,
                                        TDLSTeardown
                                        )
```

Name	Type	Valid range	Description
TDLSPeerSTAAddress	MAC Address	Any valid individual MAC Address	Specifies the MAC address of the TDLS peer STA to which a TDLS Teardown frame is transmitted.
TDLSTeardown	Sequence of octets	As defined in TDLS Teardown frame	Specifies the proposed service parameters for the TDLS teardown.

6.3.46.2.3 When generated

This primitive is generated by the SME to request that a TDLS Teardown frame be sent to the TDLS peer STA.

6.3.46.2.4 Effect of receipt

On receipt of this primitive, the MLME constructs a TDLS Teardown frame. The STA then attempts to transmit this frame to the TDLS peer STA.

6.3.46.3 MLME-TDLSTEARDOWN.indication

6.3.46.3.1 Function

This primitive indicates that a TDLS Teardown frame was received from a TDLS peer STA.

6.3.46.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-TDLSTEARDOWN.indication(
                                TDLSPeerSTAAddress,
                                TDLSTeardown
                                )
```

Name	Type	Valid range	Description
TDLSPeerSTAAddress	MACAddress	Any valid individual MAC Address	The MAC address of the TDLS peer STA from which a TDLS Teardown frame was received.
TDLSTeardown	Sequence of octets	As defined in TDLS Teardown frame	Specifies the proposed service parameters for the TDLS teardown.

6.3.46.3.3 When generated

This primitive is generated by the MLME when a valid TDLS Teardown frame is received.

6.3.46.3.4 Effect of receipt

On receipt of this primitive, the SME should operate according to the procedure in 10.22.

6.3.47 TDLS Peer U-APSD

6.3.47.1 General

The following MLME primitives support the signaling of TDLS Peer U-APSD. Figure 6-9 depicts the TDLS peer U-APSD process. The figure is only an example of the basic procedure and is not meant to be exhaustive of all possible uses of the protocol.

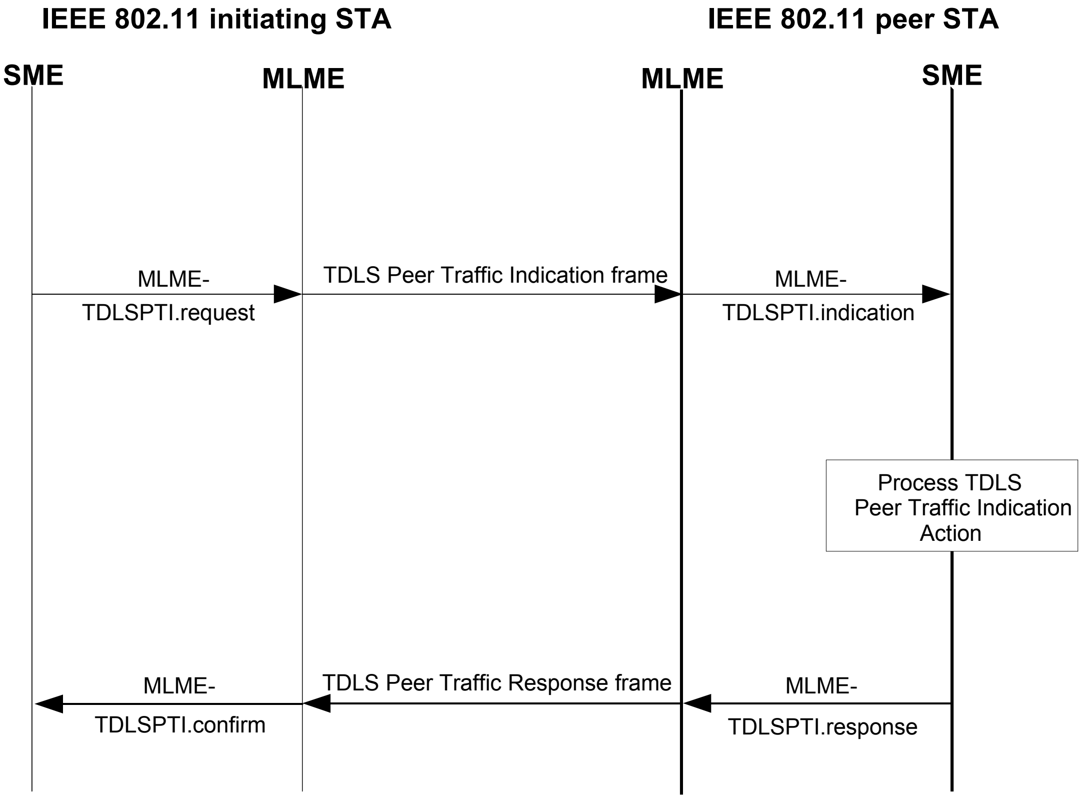


Figure 6-9—TDLS Peer U-APSD

6.3.47.2 MLME-TDLSPTI.request

6.3.47.2.1 Function

This primitive requests that a TDLS Peer Traffic Indication frame be sent to a TDLS peer STA.

6.3.47.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-TDLSPTI.request(
                     TDLSPeerSTAAddress,
                     TDLSPTI
                     )
```

Name	Type	Valid range	Description
TDLSPeerSTAAddress	MAC Address	Any valid individual MAC Address	Specifies the address of the MAC entity with which to perform the TDLS peer U-APSD process.
TDLSPTI	Sequence of octets	As defined in TDLS Peer Traffic Indication frame	Specifies the proposed service parameters for the TDLS Peer U-APSD.

6.3.47.2.3 When generated

This primitive is generated by the SME to request that a TDLS Peer Traffic Indication frame be sent to the TDLS peer STA.

6.3.47.2.4 Effect of receipt

On receipt of this primitive, the MLME constructs a TDLS Peer Traffic Indication frame. The STA then attempts to transmit this to the TDLS peer STA.

6.3.47.3 MLME-TDLSPTI.confirm

6.3.47.3.1 Function

This primitive reports the result of an MLME-TDLSPTI.request primitive to trigger an unscheduled SP from a candidate TDLS peer STA. This primitive is generated after transmitting a Peer Traffic Indication frame when this frame contains a PTI Control field, and after receiving a Peer Traffic Response frame otherwise.

6.3.47.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-TDLSPTI.confirm(
                              TDLSPeerSTAAddress,
                              TDLSPTR
                              )
```

Name	Type	Valid range	Description
TDLSPeerSTAAddress	MAC Address	Any valid individual MAC Address	Specifies the MAC address of the peer MAC entity with which to perform the TDLS Peer U-APSD process.
TDLSPTR	Sequence of octets	As defined in TDLS Peer Traffic Response frame	Specifies the proposed service parameters for the TDLS Peer U-APSD.

6.3.47.3.3 When generated

This primitive is generated by the MLME as a result of an MLME-TDLSPTI.request and indicates the results of the request.

This primitive is generated when the STA successfully receives a TDLS Peer Traffic Response frame from the TDLS peer STA or when an unspecified failure occurs.

6.3.47.3.4 Effect of receipt

On receipt of this primitive, the SME evaluates the results of the MLME-TDLSPTI.request primitive and may use the reported data.

6.3.47.4 MLME-TDLSPTI.indication

6.3.47.4.1 Function

This primitive indicates that a TDLS Peer Traffic Indication frame was received from a TDLS peer STA.

6.3.47.4.2 Semantics of the service primitive

The primitive parameters are as follows:

```
          MLME-TDLSPTI.indication(
                                          TDLSPeerSTAAddress,
                                          TDLSPTI
                                          )
```

Name	Type	Valid range	Description
TDLSPeerSTAAddress	MACAddress	Any valid individual MAC Address	The MAC address of the non-AP STA MAC entity from which a TDLS Peer Traffic Indication frame was received.
TDLSPTI	Sequence of octets	As defined in TDLS Peer Traffic Indication frame	Specifies the proposed service parameters for the TDLS Peer U-APSD.

6.3.47.4.3 When generated

This primitive is generated by the MLME when a valid TDLS Peer Traffic Indication frame is received.

6.3.47.4.4 Effect of receipt

On receipt of this primitive, the SME should operate according to the procedure as specified in 10.2.1.15.

6.3.47.5 MLME-TDLSPTI.response

6.3.47.5.1 Function

This primitive requests that a TDLS Peer Traffic Response frame be sent to the TDLS peer STA.

6.3.47.5.2 Semantics of the service primitive

The primitive parameters are as follows:

```
          MLME-TDLSPTI.response(
                                          PeerSTAAddress,
                                          TDLSPTR
                                          )
```

Name	Type	Valid range	Description
PeerSTAAddress	MAC Address	Any valid individual MAC Address	Specifies the address of the peer MAC entity with which to perform the TDLS Peer U-APSD.
TDLSPTR	Sequence of octets	As defined in TDLS Peer Traffic Response frame	Specifies the proposed service parameters for the TDLS Peer U-APSD.

6.3.47.5.3 When generated

This primitive is generated by the SME to request that a TDLS Peer Traffic Response frame be sent to the TDLS peer STA.

6.3.47.5.4 Effect of receipt

On receipt of this primitive, the MLME constructs a TDLS Peer Traffic Response frame. The STA then attempts to transmit this to the TDLS peer STA.

6.3.48 TDLS channel switching

6.3.48.1 General

The following MLME primitives support the signaling of a TDLS channel switch. Figure 6-10 depicts the TDLS channel switching process. The figure is only an example of the basic procedure and is not meant to be exhaustive of all possible uses of the protocol.

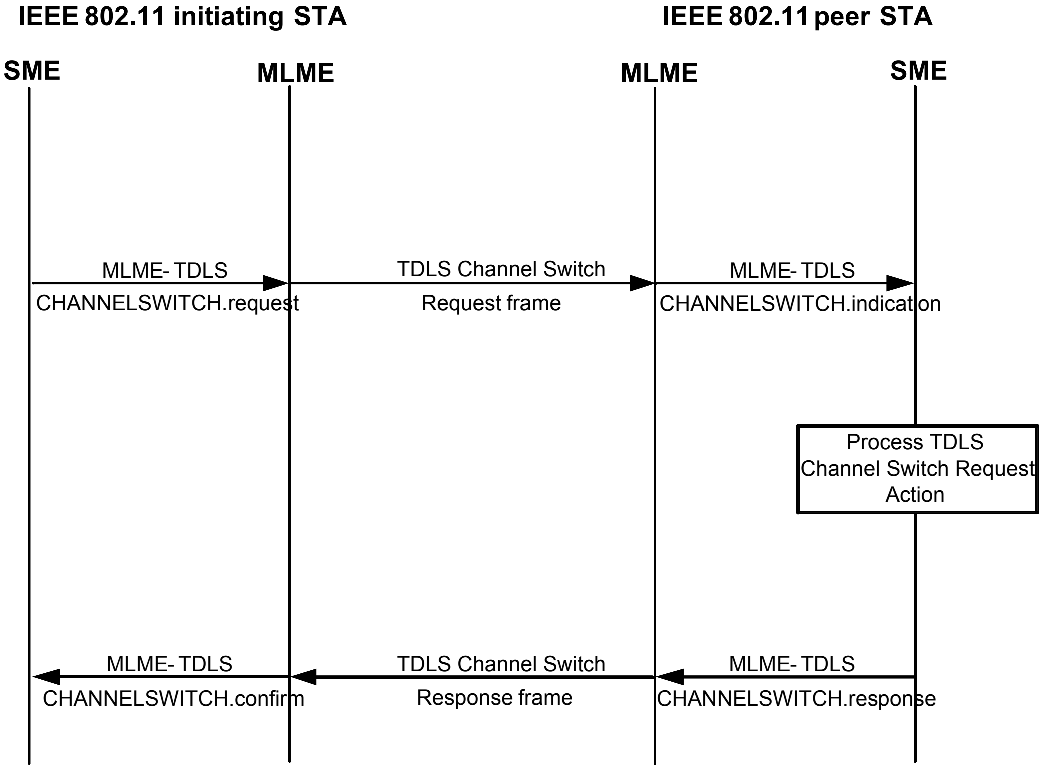

IEEE 802.11 initiating STA
IEEE 802.11 peer STA
SME
MLME
MLME
SME
MLME- TDLS CHANNELSWITCH.request
TDLS Channel Switch Request frame
MLME- TDLS CHANNELSWITCH.indication
Process TDLS Channel Switch Request Action
MLME- TDLS CHANNELSWITCH.confirm
TDLS Channel Switch Response frame
MLME- TDLS CHANNELSWITCH.response

Figure 6-10—TDLS channel switching

6.3.48.2 MLME-TDLSCHANNELSWITCH.request

6.3.48.2.1 Function

This primitive requests that a TDLS Channel Switch Request frame be sent to the TDLS peer STA.

6.3.48.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
          MLME-TDLSCHANNELSWITCH.request(
                                          TDLSPeerSTAAddress,
                                          TDLSChannelSwitchRequest
                                          )
```

Name	Type	Valid range	Description
TDLSPeerSTAAddress	MAC Address	Any valid individual MAC Address	Specifies the address of the TDLS peer MAC entity to which a TDLS Channel Switch Request frame is transmitted.
TDLSChannelSwitchRequest	Sequence of octets	As defined in TDLS Channel Switch Request frame	Specifies the proposed service parameters for the TDLS Channel Switch.

6.3.48.2.3 When generated

This primitive is generated by the SME to request that a TDLS Channel Switch Request frame be sent to the TDLS peer STA.

6.3.48.2.4 Effect of receipt

On receipt of this primitive, the MLME constructs a TDLS Channel Switch Request frame. The STA then attempts to transmit this to the TDLS peer STA.

6.3.48.3 MLME-TDLSCHANNELSWITCH.confirm

6.3.48.3.1 Function

This primitive reports the result of an MLME-TDLSCHANNELSWITCH.request primitive to switch a channel with a TDLS peer STA.

6.3.48.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
          MLME-TDLSCHANNELSWITCH.confirm(
                                          TDLSPeerSTAAddress,
                                          TDLSChannelSwitchResponse
                                          )
```

Name	Type	Valid range	Description
TDLSPeerSTAAddress	MAC Address	Any valid individual MAC Address	Specifies the MAC address of the TDLS peer STA from which a TDLS Channel Switch Response frame was received.
TDLSChannelSwitchResponse	Sequence of octets	As defined in TDLS Channel Switch Response frame	Specifies the proposed service parameters for the TDLS Channel Switch.

6.3.48.3.3 When generated

This primitive is generated by the MLME as a result of an MLME-TDLSCHANNELSWITCH.request and indicates the results of the request.

This primitive is generated when the STA successfully receives a TDLS Channel Switch Response frame from the TDLS peer STA.

6.3.48.3.4 Effect of receipt

On receipt of this primitive, the SME evaluates the results of the MLME-TDLSCHANNEL-SWITCH.request primitive and may use the reported data.

6.3.48.4 MLME-TDLSCHANNELSWITCH.indication

6.3.48.4.1 Function

This primitive indicates that a TDLS Channel Switch request frame was received from a TDLS peer STA.

6.3.48.4.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-TDLSCHANNELSWITCH.indication(
                                            TDLSPeerSTAAddress,
                                            TDLSChannelSwitchRequest
                                            )
```

Name	Type	Valid range	Description
TDLSPeerSTAAddress	MACAddress	Any valid individual MAC Address	Specifies the MAC address of the TDLS peer STA from which a TDLS Channel Switch Request frame was received.
TDLSChannelSwitch Request	Sequence of octets	As defined in TDLS Channel Switch Request frame	Specifies the proposed service parameters for the TDLS Channel Switch.

6.3.48.4.3 When generated

This primitive is generated by the MLME when a valid TDLS Channel Switch Request frame is received.

6.3.48.4.4 Effect of receipt

On receipt of this primitive, the SME should operate according to the procedure in 10.22.

6.3.48.5 MLME-TDLSCHANNELSWITCH.response

6.3.48.5.1 Function

This primitive requests that a TDLS Channel Switch Response frame be sent to the TDLS peer STA.

6.3.48.5.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-TDLSCHANNELSWITCH.response(
                                        TDLSPeerSTAAddress,
                                        TDLSChannelSwitchResponse
                                        )
```

Name	Type	Valid range	Description
TDLSPeerSTAAddress	MAC Address	Any valid individual MAC Address	Specifies the MAC address of the TDLS peer STA to which a TDLS Channel Switch Response frame is transmitted.
TDLSChannelSwitchResponse	Sequence of octets	As defined in TDLS Channel Switch Response frame	Specifies the proposed service parameters for the TDLS Channel Switch.

6.3.48.5.3 When generated

This primitive is generated by the SME to request that a TDLS Channel Switch Response frame be sent to the TDLS peer STA.

6.3.48.5.4 Effect of receipt

On receipt of this primitive, the MLME constructs a TDLS Channel Switch Response frame. The STA then attempts to transmit this frame to the TDLS peer STA.

6.3.49 TDLS Peer PSM

6.3.49.1 General

The following MLME primitives support the management of TDLS Peer PSM. Figure 6-11 depicts the TDLS Peer PSM process. The figure is only an example of the basic procedure and is not meant to be exhaustive of all possible uses of the protocol.

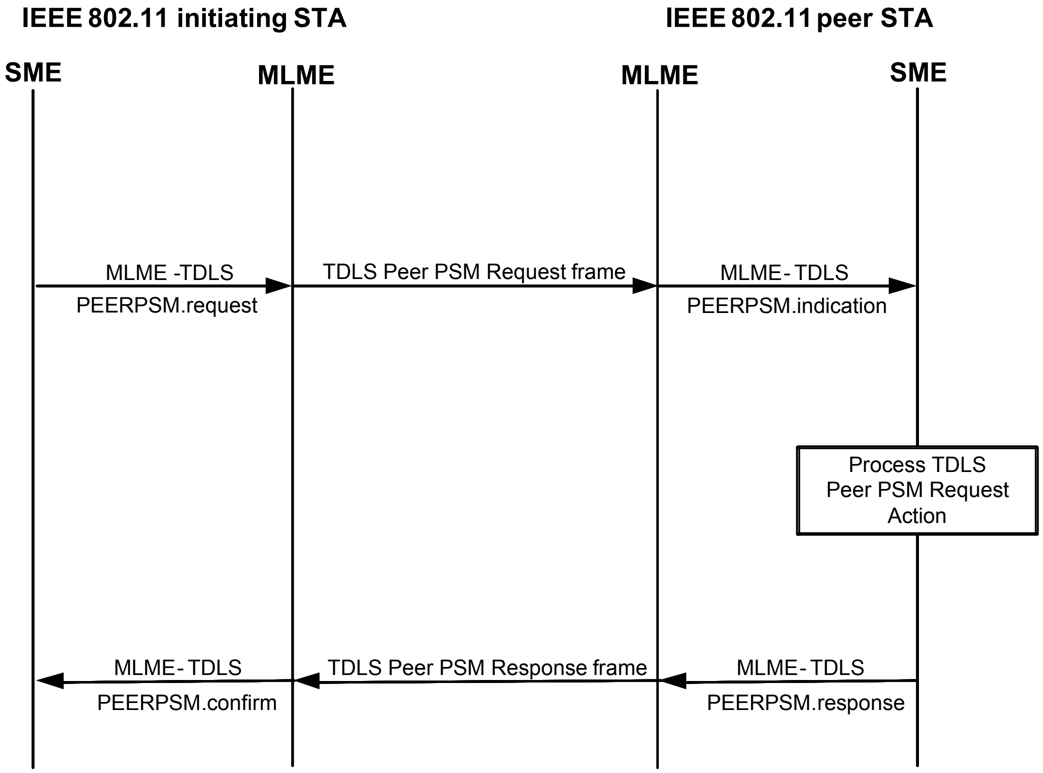


Figure 6-11—TDLS Peer PSM

6.3.49.2 MLME-TDLSPEERPSM.request

6.3.49.2.1 Function

This primitive requests that a TDLS Peer PSM Request frame be sent to the TDLS peer STA.

6.3.49.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-TDLSPEERPSM.request(
                        TDLSPeerSTAAddress,
                        TDLSPeerPSMRequest
                        )
```

Name	Type	Valid range	Description
TDLSPeerSTAAddress	MAC Address	Any valid individual MAC Address	Specifies the MAC address of the TDLS peer STA to which a TDLS Peer PSM Request frame is transmitted.
TDLSPeerPSMRequest	Sequence of octets	As defined in TDLS Peer PSM Request frame	Specifies the proposed service parameters for the TDLS Peer PSM.

6.3.49.2.3 When generated

This primitive is generated by the SME to request that a TDLS Peer PSM Request frame be sent to the TDLS peer STA.

6.3.49.2.4 Effect of receipt

On receipt of this primitive, the MLME constructs a TDLS Peer PSM Request frame. The STA then attempts to transmit this to the TDLS peer STA.

6.3.49.3 MLME-TDLSPEERPSM.confirm

6.3.49.3.1 Function

This primitive reports the result of an MLME-TDLSPEERPSM.request primitive to initiate power save mode based on scheduled service periods with a TDLS peer STA.

6.3.49.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-TDLSPEERPSM.confirm(
                                 TDLSPeerSTAAddress,
                                 TDLSPeerPSMResponse
                                 )
```

Name	Type	Valid range	Description
TDLSPeerSTAAddress	MAC Address	Any valid individual MAC Address	Specifies the MAC address of the TDLS peer STA from which a TDLS Peer PSM Response frame was received.
TDLSPeerPSMResponse	Sequence of octets	As defined in TDLS Peer PSM Response frame	Specifies the proposed service parameters for the TDLS Peer PSM.

6.3.49.3.3 When generated

This primitive is generated by the MLME as a result of an MLME-TDLSPEERPSM.request and indicates the results of the request.

This primitive is generated when the STA successfully receives a TDLS Peer PSM Response frame from the TDLS peer STA or when an unspecified failure occurs.

6.3.49.3.4 Effect of receipt

On receipt of this primitive, the SME evaluates the results of the MLME-TDLSPEERPSM.request primitive and may use the reported data.

6.3.49.4 MLME-TDLSPEERPSM.indication

6.3.49.4.1 Function

This primitive indicates that a TDLS Peer PSM Request frame was received from a TDLS peer STA.

6.3.49.4.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-TDLSPEERPSM.indication(
                                        TDLSPeerSTAAddress,
                                        TDLSPeerPSMRequest
                                        )
```

Name	Type	Valid range	Description
TDLSPeerSTAAddress	MAC Address	Any valid individual MAC Address	Specifies the MAC address of the TDLS peer STA MAC entity from which a TDLS Peer PSM Request frame was received.
TDLSPeerPSMRequest	Sequence of octets	As defined in TDLS Peer PSM Request frame	Specifies the proposed service parameters for the TDLS Peer PSM.

6.3.49.4.3 When generated

This primitive is generated by the MLME when a valid TDLS Peer PSM Request frame is received.

6.3.49.4.4 Effect of receipt

On receipt of this primitive, the SME should operate according to the procedure in 10.2.1.14.

6.3.49.5 MLME-TDLSPEERPSM.response

6.3.49.5.1 Function

This primitive requests that a TDLS Peer PSM Response frame be sent to the TDLS peer STA.

6.3.49.5.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-TDLSPEERPSM.response(
                                      TDLSPeerSTAAddress,
                                      TDLSPeerPSMResponse
                                      )
```

Name	Type	Valid range	Description
TDLSPeerSTAAddress	MAC Address	Any valid individual MAC Address	Specifies the MAC address of the TDLS peer STA to which a TDLS Peer PSM Response frame is transmitted.
TDLSPeerPSMResponse	Sequence of octets	As defined in TDLS Peer PSM Response frame	Specifies the proposed service parameters for the TDLS Peer PSM.

6.3.49.5.3 When generated

This primitive is generated by the SME to request that a TDLS Peer PSM Response frame be sent to the TDLS peer STA.

6.3.49.5.4 Effect of receipt

On receipt of this primitive, the MLME constructs a TDLS Peer PSM Response frame. The STA then attempts to transmit this to the TDLS peer STA.

6.3.50 Event request

6.3.50.1 General

This set of primitives supports the exchange of Event Request and Event Report frames. The informative diagram shown in Figure 6-12 illustrates the Event Request and Event Report process, and is not meant to be exhaustive of all possible protocol uses.

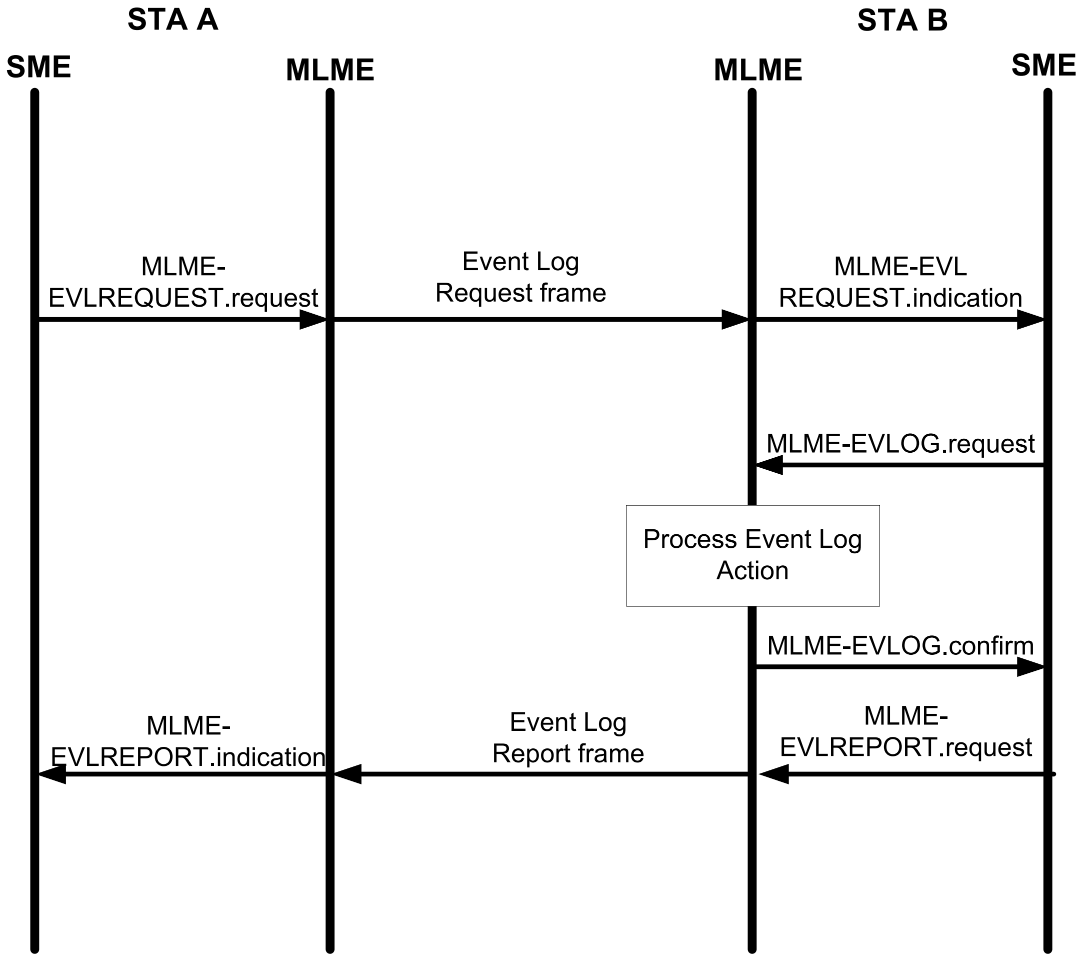


Figure 6-12—Event protocol exchange

6.3.50.2 MLME-EVLREQUEST.request

6.3.50.2.1 Function

This primitive requests the transmission of an event request to a peer entity.

6.3.50.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-EVLREQUEST.request(
                                Peer MAC Address,
                                Dialog Token,
                                Event Request Set,
                                Destination URI
                                )
```

Name	Type	Valid range	Description
Peer MAC Address	MACAddress	Any valid individual MAC Address	The address of the peer MAC entity to which the event request is sent.
Dialog Token	Integer	1–255	The dialog token to identify the event transaction.
Event Request Set	Set of event elements	Set of event elements	A set of event elements describing the requested event.
Destination URI	Destination URI element	Destination URI element	The Destination URI element as defined in 8.4.2.92.

6.3.50.2.3 When generated

This primitive is generated by the SME to request that an Event Request frame be sent to a peer entity to initiate one or more transactions.

6.3.50.2.4 Effect of receipt

On receipt of this primitive, the MLME constructs an Event Request frame containing the set of event elements specified. This frame is then scheduled for transmission.

6.3.50.3 MLME-EVLREQUEST.indication

6.3.50.3.1 Function

This primitive indicates that an Event Request frame requesting an event transaction has been received.

6.3.50.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-EVLREQUEST.indication(
                                Peer MAC Address,
                                Dialog Token,
                                Event Request Set,
                                Destination URI
                                )
```

Name	Type	Valid range	Description
Peer MAC Address	MACAddress	Any valid individual MAC Address	The address of the peer MAC entity from which the event request was received.
Dialog Token	Integer	1–255	The dialog token to identify the event transaction.
Event Request Set	Set of event elements	Set of event elements	A set of event elements describing the requested event.
Destination URI	Destination URI element	Destination URI element	The Destination URI element as defined in 8.4.2.92.

6.3.50.3.3 When generated

This primitive is generated by the MLME when a valid Event Request frame is received.

6.3.50.3.4 Effect of receipt

On receipt of this primitive, the SME either rejects the request or commences the event transaction.

6.3.51 Event report

6.3.51.1 General

This set of primitives supports the signaling of event reports.

6.3.51.2 MLME-EVLREPORT.request

6.3.51.2.1 Function

This primitive supports the signaling of event reports between peer SMEs.

6.3.51.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-EVLREPORT.request(
                                  Peer MAC Address,
                                  Dialog Token,
                                  Event Report Set
                                  )
```

Name	Type	Valid range	Description
Peer MAC Address	MACAddress	Any valid individual MAC Address	The address of the peer MAC entity to which the event report is sent.
Dialog Token	Integer	1–255	The Dialog Token to identify the event transaction.
Event Report Set	Set of event elements	Set of event elements	A set of event elements describing the response to the event request.

6.3.51.2.3 When generated

This primitive is generated by the SME to request that an Event Report frame be sent to a peer entity to report the results of one or more transactions.

6.3.51.2.4 Effect of receipt

On receipt of this primitive, the MLME constructs an Event Report frame containing the set of event elements. This frame is then scheduled for transmission.

6.3.51.3 MLME-EVLREPORT.indication

6.3.51.3.1 Function

This primitive indicates that an Event Report frame has been received from a peer entity.

6.3.51.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-EVLREPORT.indication(
                                    Peer MAC Address,
                                    Dialog Token,
                                    Event Report Set
                                    )
```

Name	Type	Valid range	Description
Peer MAC Address	MACAddress	Any valid individual MAC Address	The address of the peer MAC entity from which the event report was received.
Dialog Token	Integer	1–255	The Dialog Token to identify the event transaction.
Event Report Set	Set of event elements	Set of event elements	A set of event elements describing the response to the event request.

6.3.51.3.3 When generated

This primitive is generated by the MLME when a valid Event Report frame is received.

6.3.51.3.4 Effect of receipt

On receipt of this primitive, the event data can be made available for SME processes.

6.3.52 Event

6.3.52.1 General

This set of primitives supports the requesting and reporting of event data.

6.3.52.2 MLME-EVLOG.request

6.3.52.2.1 Function

This primitive is generated by the SME to request that the MLME identify specific events.

6.3.52.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-EVLOG.request(
```

Dialog Token,
Event Request Set
)

Name	Type	Valid range	Description
Dialog Token	Integer	1–255	The Dialog Token to identify the event transaction.
Event Request Set	Set of event elements	Set of event elements	A set of event elements describing the response to the event request.

6.3.52.2.3 When generated

This primitive is generated by the SME to request that the MLME initiate the specified event.

6.3.52.2.4 Effect of receipt

On receipt of this primitive, the MLME commences the identification of events.

6.3.52.3 MLME-EVLOG.confirm

6.3.52.3.1 Function

This primitive reports the result of an event.

6.3.52.3.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-EVLOG.confirm(
Dialog Token,
Event Report Set
)

Name	Type	Valid range	Description
Dialog Token	Integer	1–255	The dialog token to identify the event transaction.
Event Report Set	Set of event report elements	Set of event report elements	A set of event report elements describing the reported event.

6.3.52.3.3 When generated

This primitive is generated by the MLME to report the results when event identification completes.

6.3.52.3.4 Effect of receipt

On receipt of this primitive, the SME evaluates the result and, if appropriate, stores the events pending communication to the requesting entity or for local use.

6.3.53 Diagnostic request

6.3.53.1 General

This set of primitives supports the initiation of diagnostics between peer SMEs. The informative diagram shown in Figure 6-13 depicts the diagnostic reporting process and is not meant to be exhaustive of all possible protocol uses.

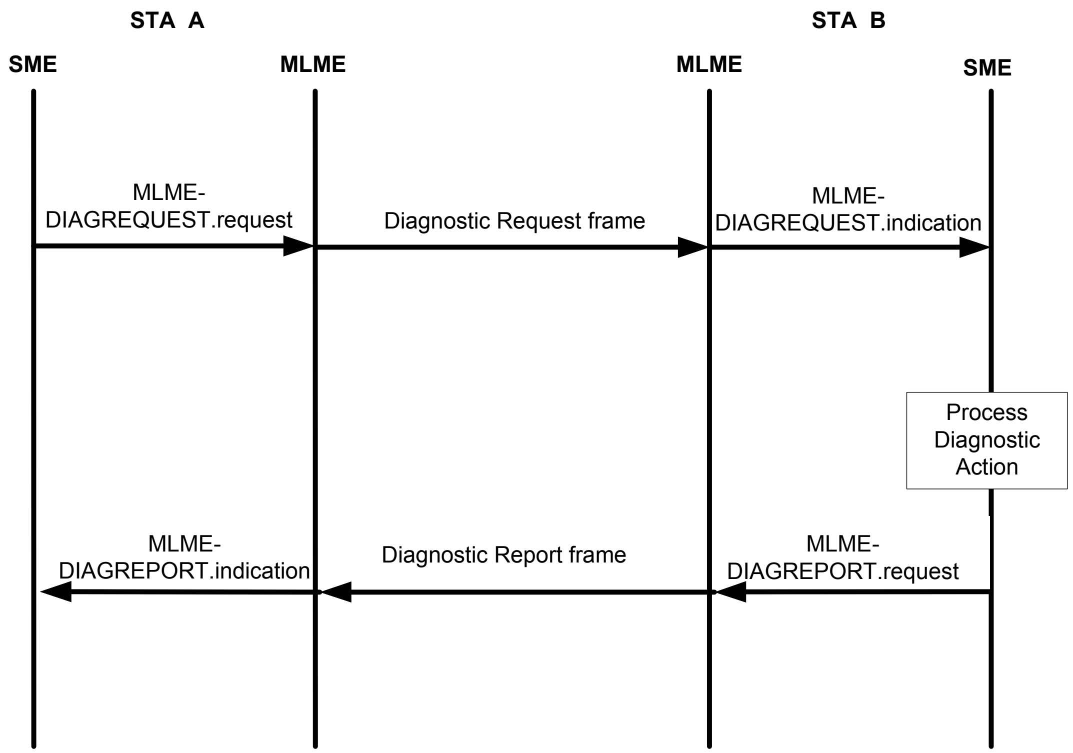


Figure 6-13—Diagnostic protocol exchange

6.3.53.2 MLME-DIAGREQUEST.request

6.3.53.2.1 Function

This primitive requests the transmission of a Diagnostic Request frame to a peer entity.

6.3.53.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-DIAGREQUEST.request(
                         Peer MAC Address,
                         Dialog Token,
                         Diagnostic Request Set,
                         Destination URI
                         )
```

Name	Type	Valid range	Description
Peer MAC Address	MACAddress	Any valid individual MAC Address	The address of the peer MAC entity to which the diagnostic request is sent.
Dialog Token	Integer	1–255	The dialog token to identify the diagnostic transaction.
Diagnostic Request Set	Set of diagnostic elements	Set of diagnostic elements	A set of diagnostic elements describing the requested diagnostics.
Destination URI	Destination URI element	Destination URI element	The Destination URI element as defined in 8.4.2.92.

6.3.53.2.3 When generated

This primitive is generated by the SME to request that a Diagnostic Request frame be sent to a peer entity to initiate one or more diagnostic transactions.

6.3.53.2.4 Effect of receipt

On receipt of this primitive, the MLME constructs a Diagnostic Request frame containing the set of diagnostic elements specified. This frame is then scheduled for transmission.

6.3.53.3 MLME-DIAGREQUEST.indication

6.3.53.3.1 Function

This primitive indicates that a Diagnostic Request frame requesting a Diagnostic transaction has been received.

6.3.53.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-DIAGREQUEST.indication(
                                    Peer MAC Address,
                                    Dialog Token,
                                    Diagnostic Request Set,
                                    Destination URI
                                    )
```

Name	Type	Valid range	Description
Peer MAC Address	MACAddress	Any valid individual MAC Address	The address of the peer MAC entity from which the diagnostic request was received.
Dialog Token	Integer	1–255	The dialog token to identify the diagnostic transaction.
Diagnostic Request Set	Set of diagnostic elements	Set of diagnostic elements	A set of diagnostic elements describing the requested diagnostics.
Destination URI	Destination URI element	Destination URI element	The Destination URI element as defined in 8.4.2.92.

6.3.53.3.3 When generated

This primitive is generated by the MLME when a valid Diagnostic Request frame is received.

6.3.53.3.4 Effect of receipt

On receipt of this primitive, the SME either rejects the request or commences the diagnostic transaction.

6.3.54 Diagnostic report

6.3.54.1 MLME-DIAGREPORT.request

6.3.54.1.1 Function

This primitive supports the signaling of diagnostic reports between peer SMEs.

6.3.54.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-DIAGREPORT.request(
                                  Peer MAC Address,
                                  Dialog Token,
                                  Diagnostic Report Set
                                  )
```

Name	Type	Valid range	Description
Peer MAC Address	MACAddress	Any valid individual MAC Address	The address of the peer MAC entity from which the diagnostic report was received.
Dialog Token	Integer	1–255	The Dialog Token to identify the diagnostic transaction.
Diagnostic Report Set	Set of diagnostic elements	Set of diagnostic elements	A set of diagnostic elements describing the results of the requested diagnostics.

6.3.54.1.3 When generated

This primitive is generated by the SME to request that a Diagnostic Report frame be sent to a peer entity to report the results of one or more diagnostic transactions.

6.3.54.1.4 Effect of receipt

On receipt of this primitive, the MLME constructs a Diagnostic Report frame containing the set of diagnostic elements. This frame is then scheduled for transmission.

6.3.54.2 MLME-DIAGREPORT.indication

6.3.54.2.1 Function

This primitive indicates that a Diagnostic Report frame has been received from a peer entity.

6.3.54.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-DIAGREPORT.indication(
                                  Peer MAC Address,
                                  Dialog Token,
                                  Diagnostic Report Set
                                  )
```

Name	Type	Valid range	Description
Peer MAC Address	MACAddress	Any valid individual MAC Address	The address of the peer MAC entity to which the diagnostic report is sent.
Dialog Token	Integer	1–255	The Dialog Token to identify the diagnostic transaction.
Diagnostic Report Set	Set of diagnostic elements	Set of diagnostic elements	A set of diagnostic elements describing the results of the requested diagnostics.

6.3.54.2.3 When generated

This primitive is generated by the MLME when a valid Diagnostic Report frame is received.

6.3.54.2.4 Effect of receipt

On receipt of this primitive, the diagnostic data can be made available for SME processes.

6.3.55 Location Configuration request

6.3.55.1 General

This set of primitives supports the exchange of location configuration parameter information between peer SMEs. The informative diagram shown in Figure 6-14 depicts the location configuration request and response process, and is not meant to be exhaustive of all possible protocol uses.

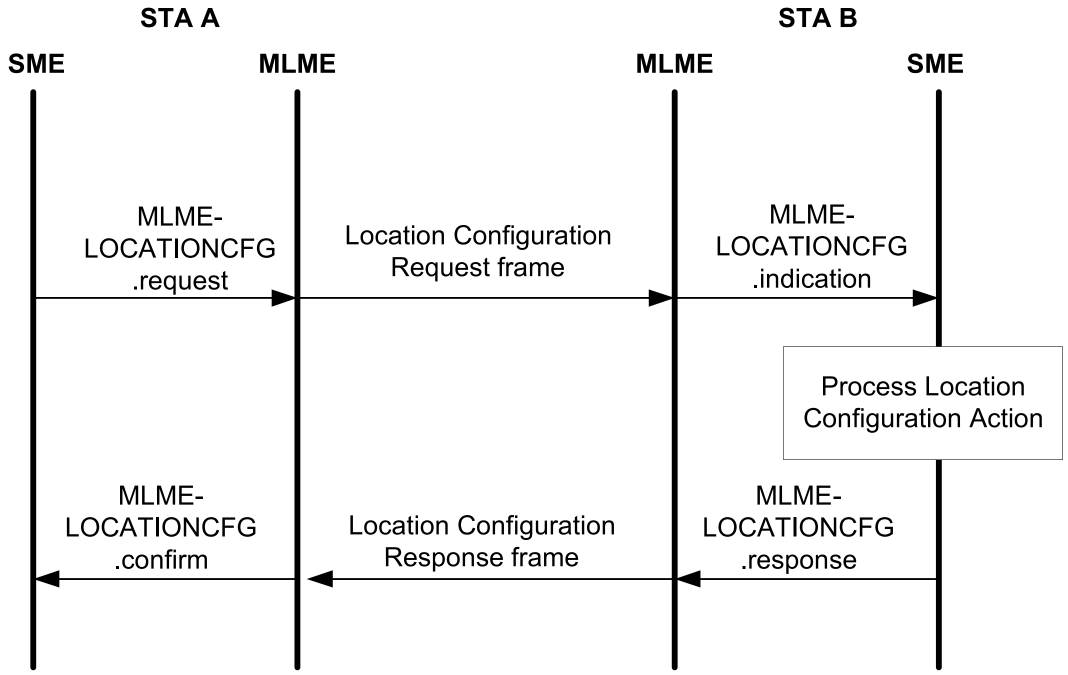

STA A
STA B
SME
MLME
MLME
SME
MLME-LOCATIONCFG.request
Location Configuration Request frame
MLME-LOCATIONCFG.indication
Process Location Configuration Action
MLME-LOCATIONCFG.confirm
Location Configuration Response frame
MLME-LOCATIONCFG.response

Figure 6-14—Location configuration request and response protocol exchange

6.3.55.2 MLME-LOCATIONCFG.request

6.3.55.2.1 Function

This primitive requests the transmission of a Location Configuration Request frame to a peer entity.

6.3.55.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-LOCATIONCFG.request(
                                  Peer MAC Address,
                                  Dialog Token,
                                  Location Parameters
                                  )
```

Name	Type	Valid range	Description
Peer MAC Address	MACAddress	Any valid individual MAC Address	The address of the peer MAC entity to which the location configuration request is sent.
Dialog Token	Integer	1–255	The dialog token to identify the location transaction.
Location Parameters	Location Parameters element	Location Parameters element	A Location Parameters element containing one or more subelements describing the STA location information. See 8.4.2.73.

6.3.55.2.3 When generated

This primitive is generated by the SME to request that a Location Configuration Request frame be sent to a peer entity to convey location configuration information.

6.3.55.2.4 Effect of receipt

On receipt of this primitive, the MLME constructs a Location Configuration Request frame containing the set of Location Parameters elements specified. This frame is then scheduled for transmission.

6.3.55.3 MLME-LOCATIONCFG.confirm

6.3.55.3.1 Function

This primitive reports the result of a location configuration request.

6.3.55.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-LOCATIONCFG.confirm(
                                  Dialog Token,
                                  Peer MAC Address,
                                  Location Parameters
                                  )
```

Name	Type	Valid range	Description
Dialog Token	Integer	1–255	The dialog token to identify the location transaction.
Peer MAC Address	MACAddress	Any valid individual MAC Address	The address of the peer MAC entity to which the location configuration response is sent.
Location Parameters	Location Parameters element	Location Parameters element	A Location Parameters element containing one or more subelements describing the STA location information. See 8.4.2.73.

6.3.55.3.3 When generated

This primitive is generated by the MLME when transmission of the Location Configuration Request frame is acknowledged, when (re)transmission of the Location Configuration Request frame fails, or when a failure reason is unspecified.

6.3.55.3.4 Effect of receipt

No effect of receipt is specified.

6.3.55.4 MLME-LOCATIONCFG.indication

6.3.55.4.1 Function

This primitive indicates that a Location Configuration Request frame has been received requesting a location transaction.

6.3.55.4.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-LOCATIONCFG.indication(
                                    Peer MAC Address,
                                    Dialog Token,
                                    Location Parameters
                                    )
```

Name	Type	Valid range	Description
Peer MAC Address	MACAddress	Any valid individual MAC Address	The address of the peer MAC entity from which the location configuration request was received.
Dialog Token	Integer	1–255	The dialog token to identify the location transaction.
Location Parameters	Location Parameters element	Location Parameters element	A Location Parameters element containing one or more subelements describing the STA location information. See 8.4.2.73.

6.3.55.4.3 When generated

This primitive is generated by the MLME when a valid Location Configuration Request frame is received.

6.3.55.4.4 Effect of receipt

On receipt of this primitive, the SME either rejects the request or commences the location transaction.

6.3.55.5 MLME-LOCATIONCFG.response

6.3.55.5.1 Function

This primitive requests the transmission of location information to a peer entity, in response to a received Location Configuration Request frame.

6.3.55.5.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-LOCATIONCFG.response(
                                    Peer MAC Address,
                                    Dialog Token,
                                    Location Parameters
                                    )
```

Name	Type	Valid range	Description
Peer MAC Address	MACAddress	Any valid individual MAC Address	The address of the peer MAC entity to which the location request is sent.
Dialog Token	Integer	1–255	The dialog token to identify the location transaction.
Location Parameters	Location Parameters element	Location Parameters element	A location parameters element containing one or more subelements describing the STA location information. See 8.4.2.73.

6.3.55.5.3 When generated

This primitive is generated by the SME to request that a Location Configuration Response frame be sent to a peer entity to convey location configuration information.

6.3.55.5.4 Effect of receipt

On receipt of this primitive, the MLME constructs a Location Configuration Response frame containing the set of location parameters elements specified. This frame is then scheduled for transmission.

6.3.56 Location track notification

6.3.56.1 General

This set of primitives supports the location track notification from one SME to one or more receiving SMEs. The informative diagram in Figure 6-15 depicts the location track notification process, is not meant to be exhaustive of all possible protocol uses.

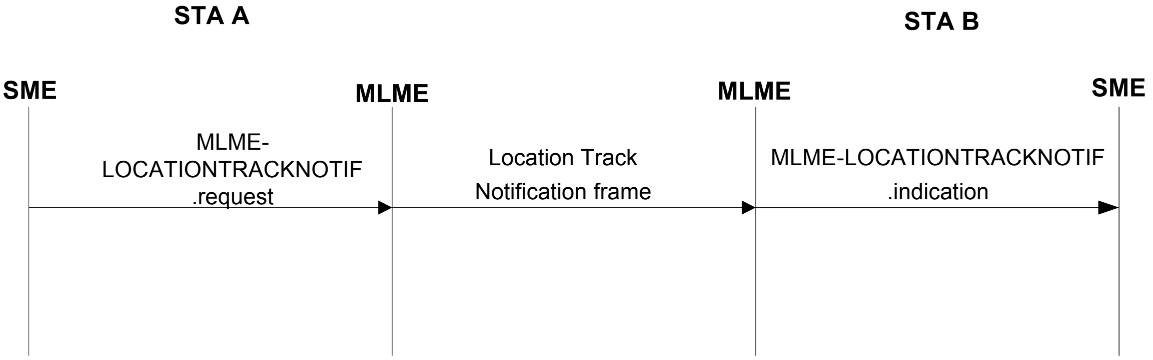


Figure 6-15—Location track notification protocol exchange

6.3.56.2 MLME-LOCATIONTRACKNOTIF.request

6.3.56.2.1 Function

This primitive requests the transmission of Location Configuration Request frame to a peer entity.

6.3.56.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
          MLME-LOCATIONTRACKNOTIF.request(
                                        Peer MAC Address,
                                        Location Parameters,
                                        Measurement Report
                                        )
```

Name	Type	Valid range	Description
Peer MAC Address	MACAddress	Any valid individual or group addressed MAC Address	The address of the peer MAC entity to which the location track notification is sent.
Location Parameters	Location Parameters element	Location Parameters element	A location parameters element containing one or more subelements describing the STA location information. See 8.4.2.73.
Measurement Report	Measurement Report element	Measurement Report element	A Measurement Report element contains the beacon measurement information. See 8.4.2.24.

6.3.56.2.3 When generated

This primitive is generated by the SME to request that a Location Track Notification frame be sent to a peer entity to help convey location information.

6.3.56.2.4 Effect of receipt

On receipt of this primitive, the MLME constructs a Location Track Notification frame containing the set of location parameters elements specified. This frame is then scheduled for transmission.

6.3.56.3 MLME-LOCATIONTRACKNOTIF.indication

6.3.56.3.1 Function

This primitive indicates that a Location Track Notification frame has been received.

6.3.56.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
   MLME-LOCATIONTRACKNOTIF.indication(
                                         Peer MAC Address,
                                         Location Parameters,
                                         Measurement Report
                                         )
```

Name	Type	Valid range	Description
Peer MAC Address	MACAddress	Any valid individual or group addressed MAC Address	The address of the peer MAC entity from which the location track notification was received.
Location Parameters	Location Parameters element	Location Parameters element	A location parameters element containing one or more subelements describing the STA location information. See 8.4.2.73.
Measurement Report	Measurement Report element	Measurement Report element	A Measurement Report element contains the beacon measurement information. See 8.4.2.24.

6.3.56.3.3 When generated

This primitive is generated by the MLME when a valid Location Track Notification frame is received.

6.3.56.3.4 Effect of receipt

On receipt of this primitive, the SME uses the information contained within the notification.

6.3.57 Timing measurement

6.3.57.1 General

The following set of primitives supports exchange of timing measurement information from one SME to another. The informative diagram in Figure 6-16 depicts various points in time that are of interest to the timing measurement procedure.

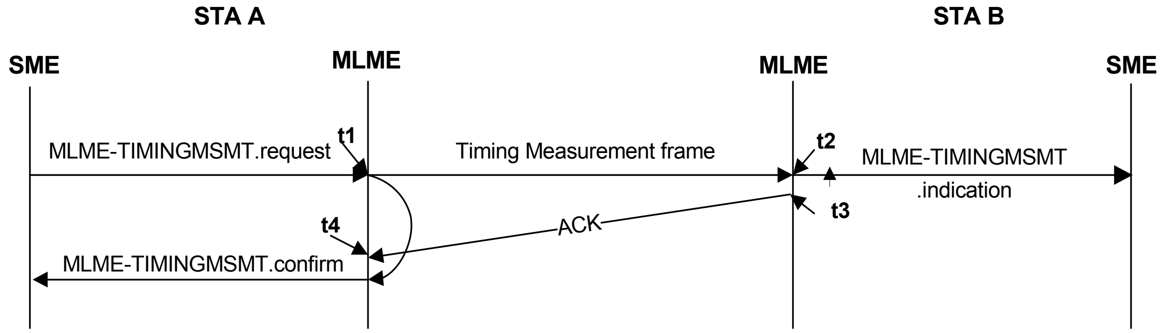


Figure 6-16—Timing measurement primitives and timestamps capture

NOTE 1—In Figure 6-16, t1 and t3 correspond to the point in time at which the start of the preamble for the transmitted frame appears at the transmit antenna port. An implementation may capture a timestamp during the transmit processing earlier or later than the point at which it actually occurs and offset the value to compensate for the time difference.

NOTE 2—In Figure 6-16, t2 and t4 correspond to the point in time at which the start of the preamble for the incoming frame arrives at the receive antenna port. Because time is needed to detect the frame and synchronize with its logical structure, an implementation determines when the start of the preamble for the incoming frame arrived at the receive antenna port by capturing a timestamp some time after it occurred and compensating for the delay by subtracting an offset from the captured value.

6.3.57.2 MLME-TIMINGMSMT.request

6.3.57.2.1 Function

This primitive requests the transmission of Timing Measurement frame to a peer entity.

6.3.57.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-TIMINGMSMT.request(
                                  Peer MAC Address,
                                  Dialog Token,
                                  Follow Up Dialog Token,
                                  t1,
                                  Max t1 Error,
                                  t4,
                                  Max t4 Error,
                                  VendorSpecific
                                  )
```

Name	Type	Valid range	Description
Peer MAC Address	MACAddress	Any valid individual addressed MAC Address	The address of the peer MAC entity to which the Timing Measurement frame is sent.
Dialog Token	Integer	1–255	The dialog token to identify the Timing Measurement transaction.
Follow Up Dialog Token	Integer	0–255	The dialog token of a Timing Measurement frame which the current frame follows. See 10.23.5.
t1	Integer		Set to the value of t1 (see Figure 6-16) expressed in 10 ns units.
Max t1 Error	Integer	0–255	Maximum error in the t1 value expressed in 10 ns units; see 8.5.15.3.
t4	Integer		Set to the value of t4 (see Figure 6-16) expressed in 10 ns units.
Max t4 Error	Integer	0–255	Maximum error in t4 value expressed in 10 ns units.
VendorSpecific	A set of information elements	As defined in 8.4.2.28	Zero or more elements.

6.3.57.2.3 When generated

This primitive is generated by the SME to request that a Timing Measurement frame be sent to a peer entity.

6.3.57.2.4 Effect of receipt

On receipt of this primitive, the MLME constructs a Timing Measurement frame with the specified parameters. This frame is then scheduled for transmission.

6.3.57.3 MLME-TIMINGMSMT.confirm

6.3.57.3.1 Function

This primitive indicates that a Timing Measurement frame has been successfully received by the peer STA to which it was sent.

6.3.57.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-TIMINGMSMT.confirm(
                                Peer MAC Address,
                                Dialog Token,
                                t1,
                                Max t1 Error,
                                t4,
                                Max t4 Error
                                )
```

Name	Type	Valid range	Description
Peer MAC Address	MACAddress	Any valid individual addressed MAC Address	The address of the peer MAC entity to which acknowledges the receipt of the Timing Measurement frame.
Dialog Token	Integer	1–255	The dialog token to identify the Timing Measurement transaction.
t1	32-bit unsigned Integer	0 – (2^{32}–1)	Set to the value of t1 (see Figure 6-16) expressed in 10 ns units.
Max t1 Error	Integer	0–255	Maximum error in the t1 value expressed in 10 ns units.
t4	32-bit unsigned Integer	0 – (2^{32}–1)	Set to the value of t4 (see Figure 6-16) expressed in 10 ns units.
Max t4 Error	Integer	0–255	Maximum error in t4 value expressed in 10 ns units.

6.3.57.3.3 When generated

This primitive is generated by the MLME when an ACK corresponding to the Timing Measurement frame is received from the peer STA.

6.3.57.3.4 Effect of receipt

On receipt of this primitive, the SME uses the information contained within the notification.

6.3.57.4 MLME-TIMINGMSMT.indication

6.3.57.4.1 Function

This primitive indicates that a Timing Measurement frame has been received and the corresponding ACK has been transmitted.

6.3.57.4.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-TIMINGMSMT.indication(
                                      Peer MAC Address,
                                      Dialog Token,
                                      Follow Up Dialog Token,
                                      t1,
                                      Max t1 Error,
                                      t4,
                                      Max t4 Error,
                                      t2,
                                      Max t2 Error,
                                      t3,
                                      Max t3 Error,
```

VendorSpecific
)

Name	Type	Valid range	Description
Peer MAC Address	MACAddress	Any valid individual addressed MAC Address	The address of the peer MAC entity from which the Timing Measurement frame was sent.
Dialog Token	Integer	1–255	The dialog token to identify the Timing Measurement transaction.
Follow Up Dialog Token	Integer	1–255	The dialog token of a Timing Measurement frame which the current frame follows. See 10.23.5.
t1	32-bit unsigned integer	0 – (2^{32}–1)	Set to the value of t1 (see Figure 6-16) expressed in 10 ns units.
Max t1 Error	Integer	0–255	Maximum error in the t1 value expressed in 10 ns units.
t4	32-bit unsigned integer	0 – (2^{32}–1)	Set to the value of t4 (see Figure 6-16) expressed in 10 ns units.
Max t4 Error	Integer	0–255	Maximum error in t4 value expressed in 10 ns units.
t2	32-bit unsigned Integer	0 – (2^{32}–1)	Set to the value of t2 (see Figure 6-16) expressed in 10 ns units.
Max t2 Error	Integer	0–255	Maximum error in t2 value expressed in 10 ns units.
t3	32-bit unsigned integer	0 – (2^{32}–1)	Set to the value of t3 (see Figure 6-16) expressed in 10 ns units.
Max t3 Error	Integer	0–255	Maximum error in t3 value expressed in 10 ns units.
VendorSpecific	A set of information elements	As defined in 8.4.2.28	Zero or more elements.

6.3.57.4.3 When generated

This primitive is generated by the MLME when a valid Timing Measurement frame is received.

6.3.57.4.4 Effect of receipt

On receipt of this primitive, the SME uses the information contained within the notification.

6.3.58 BSS Transition Management

6.3.58.1 BSS Transition Management procedure

The informative diagram shown in Figure 6-17 depicts the BSS Transition Management procedure and is not meant to be exhaustive of all possible protocol uses.

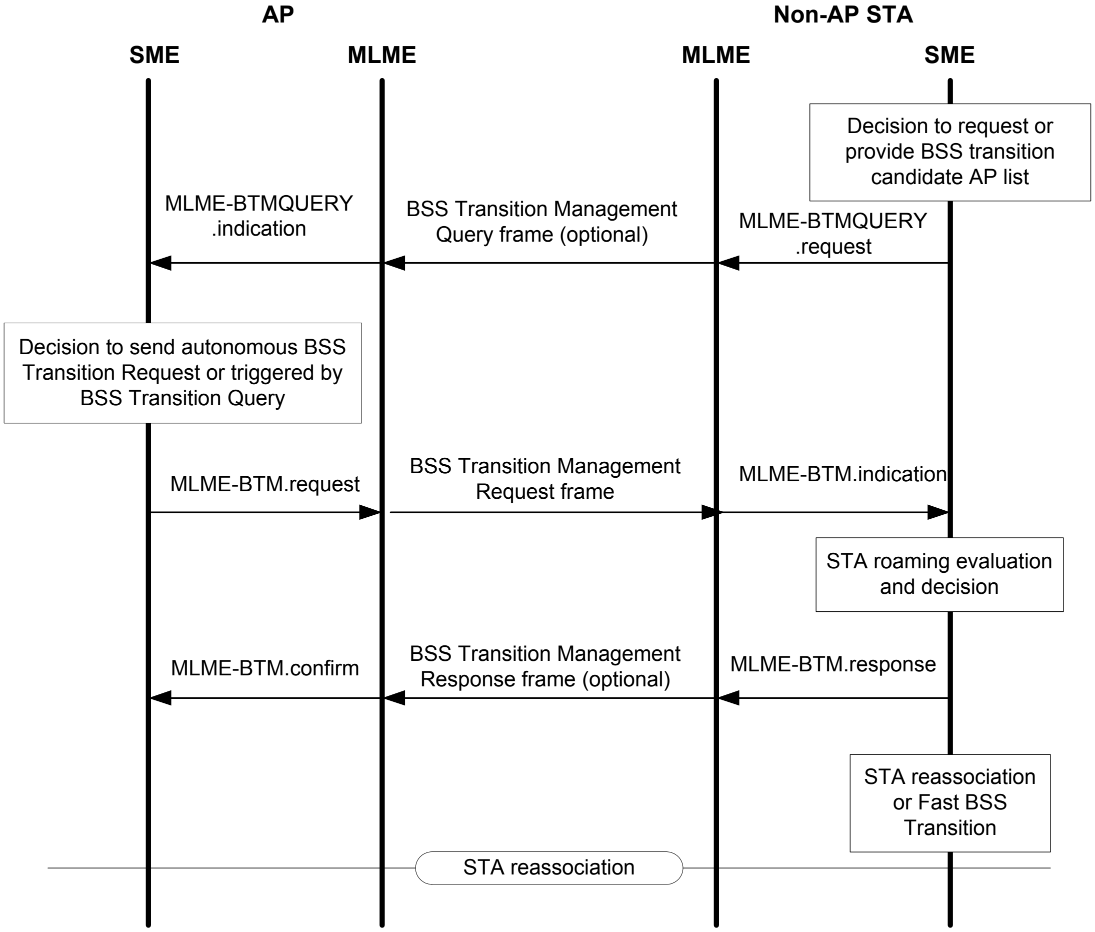

AP
Non-AP STA
SME
MLME
MLME
SME
Decision to request or provide BSS transition candidate AP list
MLME-BTMQUERY .indication
BSS Transition Management Query frame (optional)
MLME-BTMQUERY .request
Decision to send autonomous BSS Transition Request or triggered by BSS Transition Query
MLME-BTM.request
BSS Transition Management Request frame
MLME-BTM.indication
STA roaming evaluation and decision
MLME-BTM.confirm
BSS Transition Management Response frame (optional)
MLME-BTM.response
STA reassociation or Fast BSS Transition
STA reassociation

Figure 6-17—BSS Transition Management request—accepted

6.3.58.2 MLME-BTMQUERY.request

This set of primitives supports the signaling of BSS Transition Management Query frames between non-AP STAs and an AP.

6.3.58.2.1 Function

This primitive requests transmission of a BSS Transition Management Query frame to the AP with which the STA is associated.

6.3.58.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-BTMQUERY.request(
                              Peer MAC Address,
                              DialogToken,
                              BSSTransitionQueryReason,
                              BSSTransitionCandidateListEntries
                              )
```

Name	Type	Valid range	Description
Peer MAC Address	MACAddress	Any valid individual MAC Address	The address of the peer MAC entity to which the BSS Transition Management Query frame is sent.
DialogToken	Integer	1–255	The Dialog Token to identify this BSS Transition Management transaction.
BSSTransitionQuery Reason	Integer	0–255	As defined in 8.5.14.8.
BSSTransition CandidateListEntries	Set of Neighbor Report Elements	Set of Neighbor Report Elements as defined in the Neighbor Report Element in 8.4.2.39	Contains the description of candidate BSS transition APs and their capabilities as described in 8.4.2.39.

6.3.58.2.3 When generated

This primitive is generated by the SME to request that a BSS Transition Management Query frame be sent to the AP with which the STA is associated to initiate a BSS Transition Management procedure.

6.3.58.2.4 Effect of receipt

On receipt of this primitive, the MLME constructs a BSS Transition Management Query management frame of action type. The STA then attempts to transmit the frame to the AP with which it is associated.

6.3.58.3 MLME-BTMQUERY.indication

6.3.58.3.1 Function

This primitive indicates that a BSS Transition Management Query frame was received from a non-AP STA.

6.3.58.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-BTMQUERY.indication(
                              Peer MAC Address,
                              DialogToken,
                              BSSTransitionQueryReason,
                              BSSTransitionCandidateListEntries
                              )
```

Name	Type	Valid range	Description
Peer MAC Address	MACAddress	Any valid individual MAC Address	The address of the non-AP STA MAC entity from which a BSS Transition Management Query frame was received.
DialogToken	Integer	1–255	The Dialog Token to identify the BSS Transition Management transaction received in the BSS Transition Management Query frame.
BSSTransitionQuery Reason	Integer	0–255	The BSS Transition Query Reason Code in the BSS Transition Management Query frame that was received.
BSSTransition CandidateListEntries	Set of Neighbor Report Elements	Set of Neighbor Report Elements as defined in the Neighbor Report Element in 8.4.2.39	Contains the description of candidate BSS transition APs and their capabilities as described in 8.4.2.39.

6.3.58.3.3 When generated

This primitive is generated by the MLME when a valid BSS Transition Management Query frame is received.

6.3.58.3.4 Effect of receipt

On receipt of this primitive, the SME shall operate according to the procedure in 10.23.6.

6.3.58.4 MLME-BTM.request

6.3.58.4.1 Function

This primitive requests transmission of a BSS Transition Management Request frame to a non-AP STA.

6.3.58.4.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-BTM.request(
                        Peer MAC Address
                        DialogToken,
                        RequestMode,
                        DisassociationTimer,
                        ValidityInterval,
                        BSSTerminationDuration,
                        SessionInformationURL,
                        BSSTransitionCandidateListEntries
                        )
```

Name	Type	Valid range	Description
Peer MAC Address	MACAddress	Any valid individual MAC Address	The address of the peer MAC entity to which the BSS Transition Management Request frame is sent.
DialogToken	Integer	1–255	The Dialog Token to identify the BSS Transition Management transaction. Set to 0 for an autonomous BSS Transition Management Request frame.

Name	Type	Valid range	Description
RequestMode	Integer	As specified in 8.5.14.10	Contains RequestMode for the BSS Transition Management Request frame.
DisassociationTimer	Integer	0 – 65 535	Specifies the number of TBTTs until the AP disassociates the non-AP STA. A value of 0 indicates time of disassociation has not yet been determined and a value of 1 indicates disassociation shall occur before the next TBTT.
ValidityInterval	Integer	1–255	Specifies the number of beacon transmission times (TBTTs) until this recommendation of this BSS transition candidate list is no longer valid.
BSSTerminationDuration	BSS Termination Duration subelement	BSS Termination Duration subelement	Contains the BSS Termination Duration subelement (see 8.4.2.39) for the current BSS and is present only when the BSS Termination Included field is 1 in the Request mode field.
SessionInformationURL	URL	n/a	Optionally contains a URL formatted per IETF RFC 3986 where additional information pertaining to the user's accounting session may be found.
BSSTransition CandidateListEntries	Set of Neighbor Report Elements	Set of Neighbor Report Elements as defined in the Neighbor Report Element in 8.4.2.39	Contains the description of candidate BSS transition APs and their capabilities as described in 8.4.2.39.

6.3.58.4.3 When generated

This primitive is generated by the SME to request that a BSS Transition Management Request frame be sent to an associated non-AP STA. This request is sent either following the reception of a MLME-BTMQUERY.indication or may be sent autonomously.

6.3.58.4.4 Effect of receipt

On receipt of this primitive, the MLME constructs a BSS Transition Management Request management frame of action type. The STA then attempts to transmit this frame to the indicated non-AP STA.

6.3.58.5 MLME-BTM.indication

6.3.58.5.1 Function

This primitive indicates that a BSS Transition Management Request frame was received from the AP with which the STA is associated.

6.3.58.5.2 Semantics of the service primitive

The primitive parameters are as follows:

```
    MLME-BTM.indication(
                        PeerMACAddress,
                        DialogToken,
                        RequestMode,
                        DisassociationTimer,
                        ValidityInterval,
                        BSSTerminationDuration,
                        SessionInformationURL,
```

BSSTransitionCandidateListEntries
)

Name	Type	Valid range	Description
PeerMACAddress	MACAddress	Any valid individual MAC Address	The address of the MAC entity from which a BSS Transition Management Request frame was received.
DialogToken	Integer	1–255	The Dialog Token to identity the BSS Transition Management transaction as received in the BSS Transition Management Request frame.
RequestMode	Integer	As specified in 8.5.14.10	Contains the RequestMode for the BSS Transition Management Request frame.
Disassociation Timer	Integer	0 – 65 535	Specifies the number of TBTTs until the AP disassociates the non-AP STA. A value of 0 indicates time of disassociation has not been determined yet and a value of 1 indicates disassociation shall occur before the next TBTT.
ValidityInterval	Integer	1–255	Specifies the number of beacon transmission times (TBTTs) until this recommendation of this BSS transition candidate list is no longer valid.
BSSTerminationDuration	BSS Termination Duration subelement	BSS Termination Duration subelement	Contains the BSS Termination Duration subelement (see 8.4.2.39) for the current BSS and is present only when the BSS Termination Included field is 1 in the Request mode field.
SessionInformationURL	URL	n/a	Optionally contains a URL formatted per IETF RFC 3986 where additional information pertaining to the user’s accounting session may be found.
BSSTransition CandidateListEntries	Set of Neighbor Report Elements	Set of Neighbor Report Elements as defined in the Neighbor Report Element in 8.4.2.39	Contains the description of candidate BSS transition APs and their capabilities as described in 8.4.2.39.

6.3.58.5.3 When generated

This primitive is generated by the MLME when a valid BSS Transition Management Request frame is received. This primitive is also generated when a timeout or failure occurs.

6.3.58.5.4 Effect of receipt

On receipt of this primitive the SME shall operate according to the procedure in 10.23.6.

6.3.58.6 MLME-BTM.response

6.3.58.6.1 Function

This primitive requests transmission of a BSS Transition Management Response frame to the AP with which the STA is associated.

6.3.58.6.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-BTM.response(
Peer MAC Address

DialogToken,
StatusCode,
BSSTerminationDelay,
TargetBSSID,
BSSTransitionCandidateListEntries
)

Name	Type	Valid range	Description
Peer MAC Address	MACAddress	Any valid individual MAC Address	The address of the peer MAC entity to which the BSS Transition Management Query frame is sent.
DialogToken	Integer	1–255	The Dialog Token to identify the BSS Transition Management transaction.
StatusCode	Integer	0–255	As defined in 8.5.14.10.
BSSTerminationDelay	Integer	0–255	As defined in 8.5.14.10.
TargetBSSID	MACAddress	Any valid individual MAC Address	The BSSID of the BSS that the STA decided to transition to. Field shall be null if STA decided not to transition.
BSSTransition CandidateListEntries	Set of Neighbor Report Elements	Set of Neighbor Report Elements as defined in the Neighbor Report Element in 8.4.2.39	Contains the description of candidate BSS transition APs and their capabilities as described in 8.4.2.39.

6.3.58.6.3 When generated

This primitive is generated by the SME to request that a BSS Transition Management Response frame be sent to the AP with which the STA is associated.

6.3.58.6.4 Effect of receipt

On receipt of this primitive, the MLME constructs a BSS Transition Management Response management frame of action type. The non-AP STA then attempts to transmit this to the AP with which it is associated.

6.3.58.7 MLME-BTM.confirm

6.3.58.7.1 Function

This primitive reports the results of a BSS Transition Management attempt with a specified peer MAC entity that is within an AP.

6.3.58.7.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-BTM.confirm(
Peer MAC Address,
DialogToken,
BSSTransitionQueryReason,
StatusCode,
BSSTerminationDelay,
TargetBSSID,
BSSTransitionCandidateListEntries
)

Name	Type	Valid range	Description
Peer MAC Address	MAC Address	Any valid individual MAC Address	The address of the non-AP STA MAC entity from which a BSS Transition Management Response frame was received.
DialogToken	Integer	1–255	The Dialog Token to identify the BSS transition management transaction as received in the BSS Transition Management Response frame.
StatusCode	Integer	0–255	As defined in 8.5.14.10.
BSSTerminationDelay	Integer	0–255	As defined in 8.5.14.10.
TargetBSSID	MACAddress	Any valid individual MAC Address	The BSSID of the BSS that the STA indicated to transition to as received in the BSS Transition Management Response frame.
BSSTransition CandidateListEntries	Set of Neighbor Report Elements	Set of Neighbor Report Elements as defined in the Neighbor Report Element in 8.4.2.39	Contains the BSS Transition Candidate List Entries in the received BSS Transition Management Response frame.

6.3.58.7.3 When generated

This primitive is generated by the MLME when transmission of the BSS Transition Management Request frame is acknowledged, when (re)transmission of the BSS Transition Management Request frame fails, or when a failure reason is unspecified.

6.3.58.7.4 Effect of receipt

On receipt of this primitive, the SME shall operate according to the procedure in 10.23.6.

6.3.59 FMS setup

6.3.59.1 General

The following MLME primitives support the signaling of FMS Setup. The informative diagram shown in Figure 6-18 depicts the FMS setup process and is not meant to be exhaustive of all possible protocol uses.

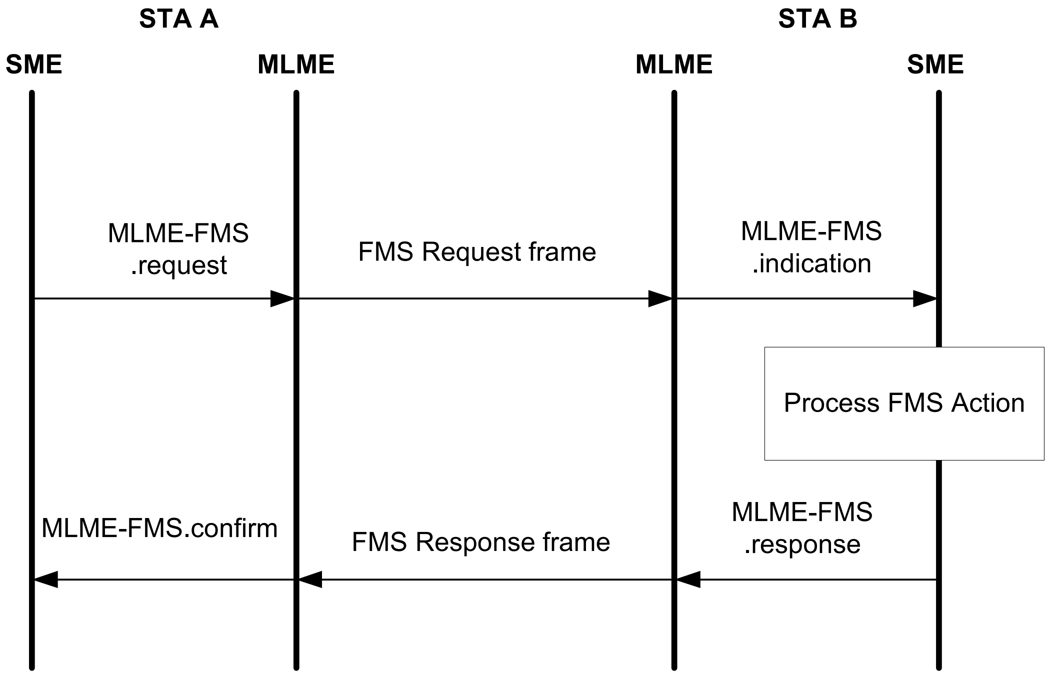


Figure 6-18—FMS setup protocol exchange

6.3.59.2 MLME-FMS.request

6.3.59.2.1 Function

This primitive requests that an FMS Request frame be sent to the AP with which the non-AP STA is associated.

6.3.59.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
    MLME-FMS.request(
                        PeerSTAAddress,
                        DialogToken,
                        FMSRequest
                        )
```

Name	Type	Valid range	Description
PeerSTAAddress	MAC Address	Any valid individual MAC Address	Specifies the address of the peer MAC entity with which to perform the FMS process.
Dialog Token	Integer	1–255	The dialog token to identify the FMS transaction.

Name	Type	Valid range	Description
FMSRequest	FMS Request element	As defined in 8.4.2.78	Specifies the proposed service parameters for the FMS.

6.3.59.2.3 When generated

This primitive is generated by the SME to request that an FMS Request frame be sent to the AP with which the STA is associated.

6.3.59.2.4 Effect of receipt

On receipt of this primitive, the MLME constructs an FMS Request action management frame. The STA then attempts to transmit this to the AP with which it is associated.

6.3.59.3 MLME-FMS.confirm

6.3.59.3.1 Function

This primitive reports the result of an FMS procedure.

6.3.59.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME- FMS.confirm(
                              Peer MAC Address,
                              Dialog Token,
                              FMSResponse
                              )
```

Name	Type	Valid range	Description
Peer MAC Address	MACAddress	Any valid individual MAC Address	The address of the peer MAC entity to which the location response is sent.
Dialog Token	Integer	0–255	The dialog token to identify the FMS transaction.
FMSResponse	FMS Response element	As defined in 8.4.2.79	Specifies service parameters for the FMS.

6.3.59.3.3 When generated

This primitive is generated by the MLME as a result of an MLME-FMS.request and indicates the results of the request. This primitive is generated when a timeout or failure occurs or when the STA receives an FMS Response frame from the AP.

6.3.59.3.4 Effect of receipt

On receipt of this primitive, the SME should operate according to the procedure in 10.2.1.16.

6.3.59.4 MLME-FMS.indication

6.3.59.4.1 Function

This primitive indicates that an FMS Request frame was received from a non-AP STA.

6.3.59.4.2 Semantics of the service primitive

The primitive parameters are as follows:

```
          MLME- FMS.indication(
                                  PeerSTAAddress,
                                  DialogToken,
                                  FMSRequest
                                  )
```

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC Address	The address of the non-AP STA MAC entity from which an FMS Request frame was received.
Dialog Token	Integer	1–255	The dialog token to identify the FMS transaction.
FMSRequest	FMS Request element	As defined in 8.4.2.78	Specifies the proposed service parameters for the FMS.

6.3.59.4.3 When generated

This primitive is generated by the MLME when a valid FMS Request frame is received.

6.3.59.4.4 Effect of receipt

On receipt of this primitive, the SME should operate according to the procedure in 10.2.1.16.

6.3.59.5 MLME-FMS.response

6.3.59.5.1 Function

This primitive is either generated in response to a received FMS Request frame or autonomously by the AP and requests the transmission of an FMS Response frame.

6.3.59.5.2 Semantics of the service primitive

The primitive parameters are as follows:

```
          MLME-FMS.response(

                                  PeerSTAAddress,
                                  FMSResponse
                                  )
```

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC Address	The address of the non-AP STA MAC entity from which an FMS Request frame was received.
Dialog Token	Integer	0–255	The dialog token to identify the FMS transaction.
FMSResponse	FMS Response element	As defined in 8.4.2.79	Specifies service parameters for the FMS.

6.3.59.5.3 When generated

This primitive is generated by the SME to request that an FMS Response frame be sent to a peer entity to convey FMS information.

6.3.59.5.4 Effect of receipt

On receipt of this primitive, the MLME constructs an FMS Response frame. The STA then attempts to transmit this to the non-AP STA indicated by the PeerSTAAddress parameter.

6.3.60 Collocated Interference request

6.3.60.1 General

This set of primitives supports the exchange of collocated interference information between peer SMEs. The informative diagram shown in Figure 6-19 depicts the Collocated Interference request and response process, and is not meant to be exhaustive of all possible protocol uses.

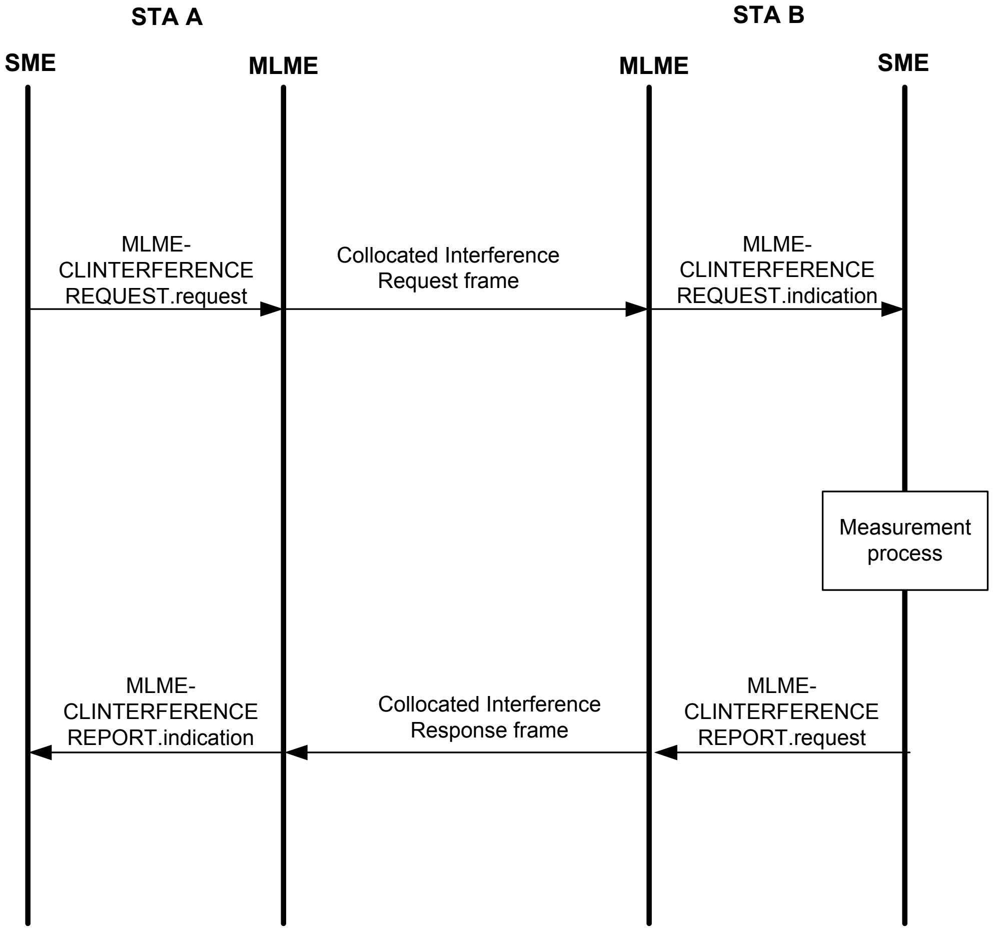

STA A
STA B
SME
MLME
MLME
SME
MLME-CLINTERFERENCE REQUEST.request
Collocated Interference Request frame
MLME-CLINTERFERENCE REQUEST.indication
Measurement process
MLME-CLINTERFERENCE REPORT.indication
Collocated Interference Response frame
MLME-CLINTERFERENCE REPORT.request

Figure 6-19—Collocated interference protocol exchange

6.3.60.2 MLME-CLINTERFERENCEREQUEST.request

6.3.60.2.1 Function

This primitive requests the transmission of Collocated Interference Request frame to a peer entity.

6.3.60.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-CLINTERFERENCEREQUEST.request(
                                    Peer MAC Address,
                                    Dialog Token,
                                    Request Info
                                    )
```

Name	Type	Valid range	Description
Peer MAC Address	MACAddress	Any valid individual MAC Address, or the group MAC address	The address of the peer MAC entity to which the collocated interference request is sent.
Dialog Token	Integer	1–255	The dialog token to identify the collocated interference transaction.
Request Info	Integer	As defined in the Collocated Interference Request frame, see 8.5.14.13	Specifies the requested information.

6.3.60.2.3 When generated

This primitive is generated by the SME to request that a Collocated Interference Request frame be sent to a peer entity.

6.3.60.2.4 Effect of receipt

On receipt of this primitive, the MLME constructs a Collocated Interference Request frame. This frame is then scheduled for transmission.

6.3.60.3 MLME-CLINTERFERENCEREQUEST.indication

6.3.60.3.1 Function

This primitive indicates that a Collocated Interference Request frame has been received requesting a Collocated Interference report.

6.3.60.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-CLINTERFERENCEREQUEST.indication(
                                    Peer MAC Address,
                                    Dialog Token,
                                    Request Info
                                    )
```

Name	Type	Valid range	Description
Peer MAC Address	MACAddress	Any valid individual MAC Address or the group MAC address	The address of the peer MAC entity from which the Collocated Interference request was received.
Dialog Token	Integer	1–255	The dialog token to identify the collocated interference transaction.
Request Info	Integer	As defined in the Collocated Interference Request frame, see 8.5.14.13	Specifies the requested information.

6.3.60.3.3 When generated

This primitive is generated by the MLME when a valid Collocated Interference Request frame is received.

6.3.60.3.4 Effect of receipt

On receipt of this primitive, the SME either rejects the request or commences the Collocated Interference reporting procedure as described in 10.23.8.

6.3.61 Collocated Interference report

6.3.61.1 General

This set of primitives supports the exchange of collocated interference information between peer SMEs.

6.3.61.2 MLME-CLINTERFERENCEREPORT.request

6.3.61.2.1 Function

This primitive requests the transmission of Collocated Interference Report to a peer entity, in response to a received Collocated Interference Request frame.

6.3.61.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-CLINTERFERENCEREPORT.request(
                                        Peer MAC Address,
                                        Dialog Token,
                                        Collocated Interference Report
                                        )
```

Name	Type	Valid range	Description
Peer MAC Address	MACAddress	Any valid individual MAC Address	The address of the peer MAC entity to which the Collocated Interference Report is sent.
Dialog Token	Integer	1–255	The dialog token to identify the collocated interference transaction.
Collocated Interference Report	Collocated Interference Report elements	As defined in 8.4.2.87	Specifies the collocated interference characteristics.

6.3.61.2.3 When generated

This primitive is generated by the SME to request that a Collocated Interference Report frame be sent to a peer entity to convey collocated interference information.

6.3.61.2.4 Effect of receipt

On receipt of this primitive, the MLME constructs a Collocated Interference Report frame. This frame is then scheduled for transmission.

6.3.61.3 MLME-CLINTERFERENCEREPORT.indication

6.3.61.3.1 Function

This primitive indicates that a Collocated Interference Report frame has been received.

6.3.61.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-CLINTERFERENCEREPORT.indication(
                                         Peer MAC Address,
                                         Dialog Token,
                                         Collocated Interference Report
                                         )
```

Name	Type	Valid range	Description
Peer MAC Address	MACAddress	Any valid individual MAC Address	The address of the peer MAC entity from which the location response was received.
Dialog Token	Integer	1–255	The dialog token to identify the location transaction.
Collocated Interference Report	Collocated Interference Report elements	As defined in 8.4.2.87	Specifies the collocated interference characteristics.

6.3.61.3.3 When generated

This primitive is generated by the MLME when a valid Collocated Interference Report frame is received.

6.3.62 TFS Setup

6.3.62.1 General

This set of primitives supports the exchange of the signaling of TFS Setup between peer SMEs. The informative diagram shown in Figure 6-20 depicts the TFS request and response process, and is not meant to be exhaustive of all possible protocol uses.

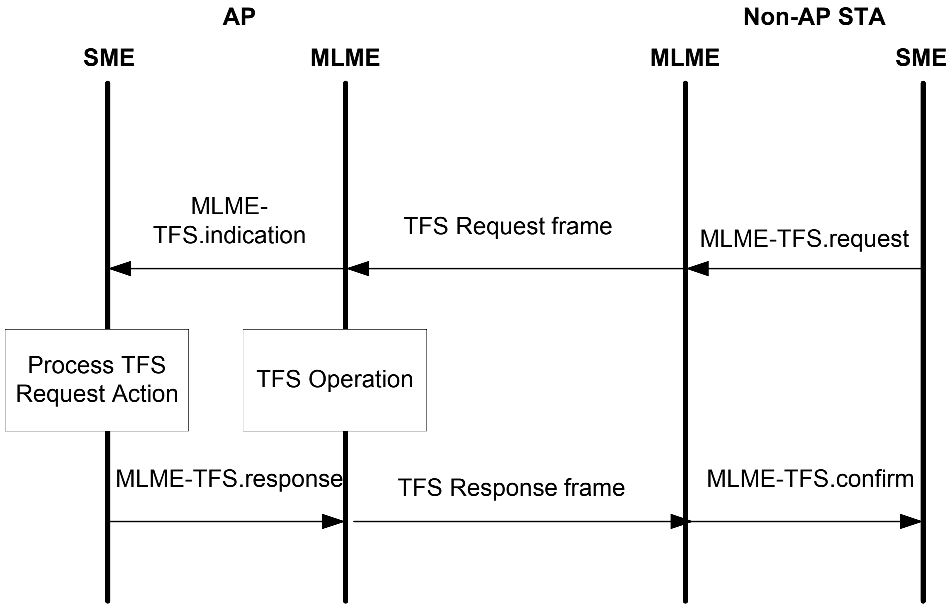


Figure 6-20—TFS request and response exchange

6.3.62.2 MLME-TFS.request

6.3.62.2.1 Function

This primitive requests that a TFS Request frame be sent to the AP with which the STA is associated.

6.3.62.2.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-TFS.request(
PeerSTAAddress,
DialogToken,
TFSRequest,
VendorSpecific
)

Name	Type	Valid range	Description
PeerSTAAddress	MAC Address	Any valid individual MAC Address	Specifies the address of the peer MAC entity for TFS.
DialogToken	Integer	0–255	The Dialog Token to identify the TFS Request and Response transaction.

Name	Type	Valid range	Description
TFSRequest	A set of TFS Request elements	As defined in the TFS Request element	Specifies the proposed service parameters for the TFS. One or more TFS Request elements.
VendorSpecific	A set of information elements	As defined in 8.4.2.28	Zero or more elements.

6.3.62.2.3 When generated

This primitive is generated by the SME to request that a TFS Request frame be sent to the AP with which the STA is associated.

6.3.62.2.4 Effect of receipt

On receipt of this primitive, the MLME constructs a TFS Request action management frame. The STA then attempts to transmit this to the AP with which it is associated.

6.3.62.3 MLME-TFS.confirm

6.3.62.3.1 Function

This primitive reports the result of a TFS procedure.

6.3.62.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-TFS.confirm(
                        PeerSTAAddress,
                        DialogToken,
                        TFSResponse,
                        VendorSpecific
                        )
```

Name	Type	Valid range	Description
PeerSTAAddress	MAC Address	Any valid individual MAC Address	The address of the peer MAC entity from which the TFS Response frame is received.
DialogToken	Integer	0–255	The Dialog Token to identify the TFS Request and Response transaction.
TFSResponse	A set of TFS Response elements	As defined in the TFS Response element	Specifies service parameters for the TFS. One or more TFS Response elements.
VendorSpecific	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.62.3.3 When generated

This primitive is generated by the MLME when transmission of the TFS Request frame is acknowledged, when (re)transmission of the TFS Request frame fails, when a failure reason is unspecified, or when the STA receives a TFS Response frame from the AP.

This primitive is also generated when the MLME-TFS.request contains invalid parameters and when a timeout or failure occurs.

6.3.62.3.4 Effect of receipt

On receipt of this primitive, the SME evaluates the result and may use the reported data. The SME should operate according to the procedure in 10.23.11.

6.3.62.4 MLME-TFS.indication

6.3.62.4.1 Function

This primitive indicates that a TFS Request frame was received from a non-AP STA.

6.3.62.4.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-TFS.indication(
                                PeerSTAAddress,
                                DialogToken,
                                TFSRequest,
                                VendorSpecific
                                )
```

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC Address	The address of the non-AP STA MAC entity from which a TFS Request frame was received.
DialogToken	Integer	0–255	The Dialog Token to identify the TFS Request and Response transaction.
TFSRequest	A set of TFS Request elements	As defined in the TFS Request element	Specifies the proposed service parameters for the TFS. One or more TFS Request elements.
VendorSpecific	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.62.4.3 When generated

This primitive is generated by the MLME when a valid TFS Request action management frame is received.

6.3.62.4.4 Effect of receipt

On receipt of this primitive the SME should operate according to the procedure in 10.23.11.

6.3.62.5 MLME-TFS.response

6.3.62.5.1 Function

This primitive is generated in response to an MLME-TFS.indication requesting a TFS Response frame be sent to a non-AP STA.

6.3.62.5.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-TFS.response(
                                PeerSTAAddress,
                                DialogToken,
                                TFSResponse,
                                VendorSpecific
                                )
```

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC Address	The address of the non-AP STA MAC entity from which a TFS Request frame was received.
DialogToken	Integer	0–255	The Dialog Token to identify the TFS Request and Response transaction.
TFSResponse	A set of TFS Response elements	As defined in 8.4.2.83	Specifies service parameters for the TFS. One or more TFS Response elements.
VendorSpecific	A set of elements.	As defined in 8.4.2.28	Zero or more elements.

6.3.62.5.3 When generated

This primitive is generated by the SME in response to an MLME-TFS.indication requesting a TFS Response be sent to a non-AP STA.

6.3.62.5.4 Effect of receipt

On receipt of this primitive, the MLME constructs a TFS Response action management frame. The STA then attempts to transmit this to the non-AP STA indicated by the PeerSTAAddress parameter.

6.3.63 Sleep Mode request

6.3.63.1 General

This set of primitives supports the exchange of sleep mode parameter information between peer SMEs. The informative diagram shown in Figure 6-21 depicts the sleep mode request and response process, and is not meant to be exhaustive of all possible protocol uses.

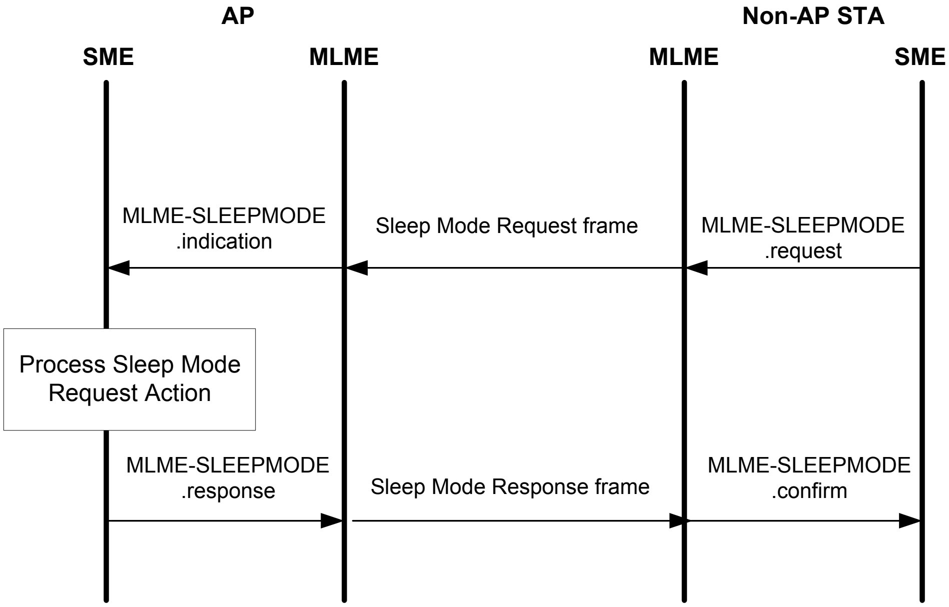


Figure 6-21—Sleep mode request and response exchange

6.3.63.2 MLME-SLEEPMODE.request

6.3.63.2.1 Function

This primitive requests that a Sleep Mode Request frame be sent to the AP with which the STA is associated.

6.3.63.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-SLEEPMODE.request(
                                          PeerSTAAddress,
                                          DialogToken,
                                          SleepMode,
                                          TFSRequest,
                                          VendorSpecificInfo
                                          )
```

Name	Type	Valid range	Description
PeerSTAAddress	MAC Address	Any valid individual MAC Address	The address of the peer MAC entity to which the Sleep Mode Request frame is to be sent.
DialogToken	Integer	0–255	The Dialog Token to identify the Sleep Mode Request and Response transaction.
SleepMode	As defined in the Sleep Mode element	As defined in the Sleep Mode element	Specifies the proposed sleep mode service parameters for the Sleep Mode Request frame.
TFSRequest	A set of TFS Request elements	As defined in 8.4.2.82	Specifies the proposed TFS service parameters for the Sleep Mode Request frame.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.63.2.3 When generated

This primitive is generated by the SME to request that a Sleep Mode Request frame be sent to the AP with which the STA is associated.

6.3.63.2.4 Effect of receipt

On receipt of this primitive, the MLME constructs a Sleep Mode Request frame. The STA then attempts to transmit this to the AP with which it is associated.

6.3.63.3 MLME-SLEEPMODE.indication

6.3.63.3.1 Function

This primitive indicates that a Sleep Mode Request frame was received from a non-AP STA.

6.3.63.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-SLEEPMODE.indication(
                          PeerSTAAddress,
                          DialogToken,
                          SleepMode,
                          TFSRequest,
                          VendorSpecificInfo
                          )
```

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC Address	The address of the peer MAC entity from which the Sleep Mode request frame is received.
DialogToken	Integer	0–255	The Dialog Token to identify the Sleep Mode Request and Response transaction.
SleepMode	As defined in the Sleep Mode element	As defined in the Sleep Mode element	Specifies the proposed sleep mode service parameters for the Sleep Mode Request frame.

Name	Type	Valid range	Description
TFSRequest	A set of TFS Request elements	As defined in 8.4.2.82	Specifies the proposed TFS service parameters for the Sleep Mode Request frame.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.63.3.3 When generated

This primitive is generated by the MLME when a valid Sleep Mode Request frame is received.

6.3.63.3.4 Effect of receipt

On receipt of this primitive, the SME should operate according to the procedure in 10.2.1.18.

6.3.63.4 MLME-SLEEPMODE.response

6.3.63.4.1 Function

This primitive requests the transmission of Sleep Mode information to a peer entity, in response to a received Sleep Mode Request frame.

6.3.63.4.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-SLEEPMODE.response(
                                  PeerSTAAddress,
                                  DialogToken,
                                  SleepMode,
                                  TFSResponse,
                                  VendorSpecificInfo
                                  )
```

Name	Type	Valid range	Description
PeerSTAAddress	MAC Address	Any valid individual MAC Address	The address of the peer MAC entity to which the Sleep Mode Response frame is to be sent.
DialogToken	Integer	0–255	The Dialog Token to identify the Sleep Mode Request and Response transaction.
SleepMode	As defined in the Sleep Mode element	As defined in the Sleep Mode element	Specifies the proposed sleep mode service parameters for the Sleep Mode Response frame.
TFSResponse	A set of TFS Request elements	As defined in 8.4.2.83	Specifies the proposed TFS service parameters for the Sleep Mode Response frame.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.63.4.3 When generated

This primitive is generated by the SME to request that a Sleep Mode Response frame be sent to a peer entity to convey Sleep Mode information.

6.3.63.4.4 Effect of receipt

On receipt of this primitive, the MLME constructs a Sleep Mode Response frame containing the Sleep Mode elements specified. This frame is then scheduled for transmission.

6.3.63.5 MLME-SLEEPMODE.confirm

6.3.63.5.1 Function

This primitive reports the result of a request to send a Sleep Mode Response frame.

6.3.63.5.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-SLEEPMODE.confirm(
                                PeerSTAAddress,
                                DialogToken,
                                SleepMode,
                                TFSResponse,
                                VendorSpecificInfo
                                )
```

Name	Type	Valid range	Description
PeerSTAAddress	MAC Address	Any valid individual MAC Address	The address of the peer MAC entity from which the Sleep Mode Response frame is received.
DialogToken	Integer	0–255	The Dialog Token to identify the Sleep Mode Request and Response transaction.
SleepMode	As defined in the Sleep Mode element	As defined in the Sleep Mode element	Specifies the proposed sleep mode service parameters for the Sleep Mode Response frame.
TFSResponse	A set of TFS Request elements	As defined in the TFS Response element	Specifies the proposed TFS service parameters for the Sleep Mode Response frame.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.63.5.3 When generated

This primitive is generated by the MLME when transmission of the Sleep Mode Request frame is acknowledged, when (re)transmission of the Sleep Mode Request frame fails, when the Sleep Mode Request frame contains invalid parameters, or when a failure reason is unspecified.

6.3.63.5.4 Effect of receipt

No effect of receipt is specified.

6.3.64 TIM broadcast setup

6.3.64.1 General

The following MLME primitives support the signaling of TIM Broadcast Setup. The informative diagram shown in Figure 6-22 depicts the TIM Broadcast setup process and is not meant to be exhaustive of all possible protocol uses.

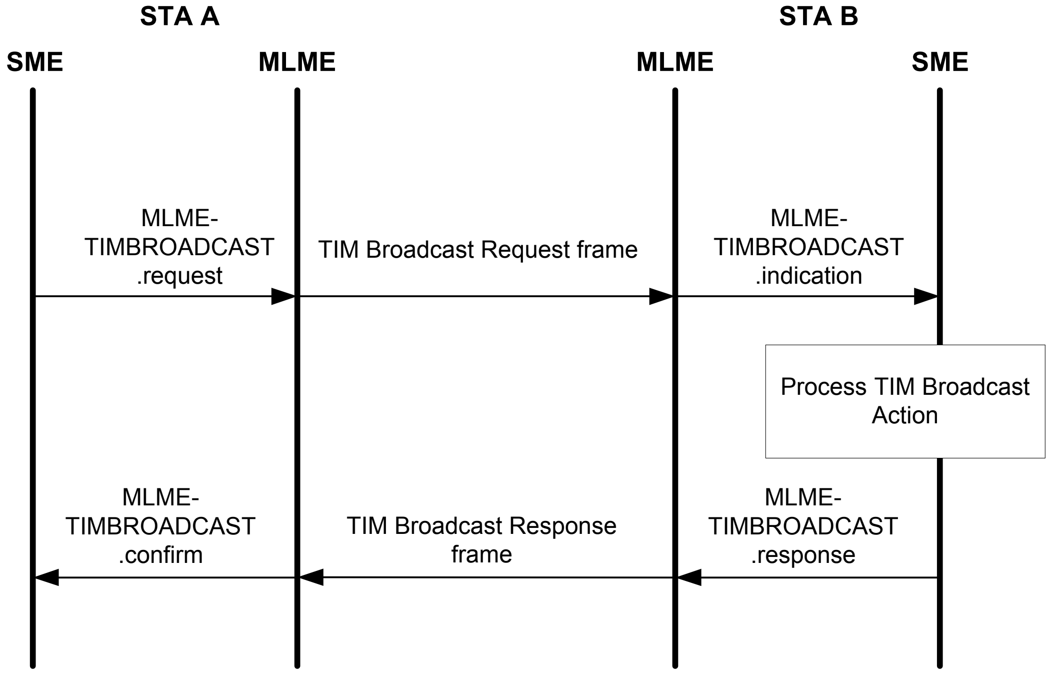

STA A
STA B
SME
MLME
MLME
SME
MLME-TIMBROADCAST.request
TIM Broadcast Request frame
MLME-TIMBROADCAST.indication
Process TIM Broadcast Action
MLME-TIMBROADCAST.confirm
TIM Broadcast Response frame
MLME-TIMBROADCAST.response

Figure 6-22—TIM broadcast setup protocol exchange

6.3.64.2 MLME-TIMBROADCAST.request

6.3.64.2.1 Function

This primitive requests that a TIM Broadcast Request frame be sent to the AP with which the non-AP STA is associated.

6.3.64.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-TIMBROADCAST.request(
                                    PeerSTAAddress,
                                    Dialog Token,
                                    TIMBroadcastRequest
                                    )
```

Name	Type	Valid range	Description
PeerSTAAddress	MAC Address	Any valid individual MAC Address	Specifies the address of the peer MAC entity with which to perform the TIM Broadcast process.
Dialog Token	Integer	1–255	The Dialog Token to identify the TIM Broadcast request and response transaction.
TIMBroadcastRequest	As defined in TIM Broadcast Request element	As defined in 8.4.2.85	Specifies the proposed service parameters for the TIM Broadcast.

6.3.64.2.3 When generated

This primitive is generated by the SME to request that a TIM Broadcast Request frame be sent to the AP with which the STA is associated.

6.3.64.2.4 Effect of receipt

On receipt of this primitive, the MLME constructs a TIM Broadcast Request action management frame. The STA then attempts to transmit this to the AP with which it is associated.

6.3.64.3 MLME-TIMBROADCAST.confirm

6.3.64.3.1 Function

This primitive reports the result of a TIM Broadcast procedure.

6.3.64.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-TIMBROADCAST.confirm(
                                    PeerSTAAddress,
                                    Dialog Token,
                                    TIMBroadcastResponse
                                    )
```

Name	Type	Valid range	Description
PeerSTAAddress	MAC Address	Any valid individual MAC Address	Specifies the address of the peer MAC entity with which to perform the TIM Broadcast process.
Dialog Token	Integer	0–255	The Dialog Token to identify the TIM Broadcast request and response transaction.
TIMBroadcast-Response	As defined in TIM Broadcast Response element	As defined in 8.4.2.86	Specifies service parameters for the TIM Broadcast.

6.3.64.3.3 When generated

This primitive is generated by the MLME when transmission of the TIM Broadcast Request frame is acknowledged, when (re)transmission of the TIM Broadcast Request frame fails, when the TIM Broadcast Request frame contains invalid parameters, or when a failure reason is unspecified.

6.3.64.3.4 Effect of receipt

On receipt of this primitive, the SME evaluates the results in the TIMBroadcastResponse element and may use the reported data.

6.3.64.4 MLME-TIMBROADCAST.indication

6.3.64.4.1 Function

This primitive indicates that a TIM Broadcast Request frame was received from a non-AP STA.

6.3.64.4.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-TIMBROADCAST.indication(
                                    PeerSTAAddress,
                                    Dialog Token,
                                    TIMBroadcastRequest
                                    )
```

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC Address	The address of the non-AP STA MAC entity from which a TIM Broadcast Request frame was received.
Dialog Token	Integer	1–255	The Dialog Token to identify the TIM Broadcast request and response transaction.
TIMBroadcast-Request	As defined in TIM Broadcast Request element	As defined in 8.4.2.85	Specifies the proposed service parameters for the TIM Broadcast.

6.3.64.4.3 When generated

This primitive is generated by the MLME when a valid TIM Broadcast Request frame is received.

6.3.64.4.4 Effect of receipt

On receipt of this primitive, the SME should operate according to the procedure in 10.2.1.17.

6.3.64.5 MLME-TIMBROADCAST.response

6.3.64.5.1 Function

This primitive is generated in response to an MLME-TIMBROADCAST.indication requesting a TIM Broadcast Response frame be sent to a non-AP STA.

6.3.64.5.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-TIMBROADCAST.response(
                                PeerSTAAddress,
                                Dialog Token,
                                TIMBroadcastResponse
                                )
```

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC Address	The address of the non-AP STA MAC entity from which a TIM Broadcast Request frame was received.
Dialog Token	Integer	0–255	The Dialog Token to identify the TIM Broadcast request and response transaction.
TIM Broadcast Response	As defined in TIM Broadcast Response element	As defined in 8.4.2.86	Specifies service parameters for the TIM Broadcast.

6.3.64.5.3 When generated

This primitive is generated by the SME in response to an MLME-TIMBROADCAST.indication requesting a TIM Broadcast Response be sent to a non-AP STA.

6.3.64.5.4 Effect of receipt

On receipt of this primitive, the MLME constructs a TIM Broadcast Response frame. The STA then attempts to transmit this to the non-AP STA indicated by the PeerSTAAddress parameter.

6.3.65 QoS Traffic Capability Update

6.3.65.1 MLME-QOSTRAFFICCAPUPDATE.request

6.3.65.1.1 Function

This primitive requests that a QoS Traffic Capability Update frame be sent to the AP with which the STA is associated. The informative diagram shown in Figure 6-23 depicts the QoS Traffic Capability Update process and is not meant to be exhaustive of all possible protocol uses.

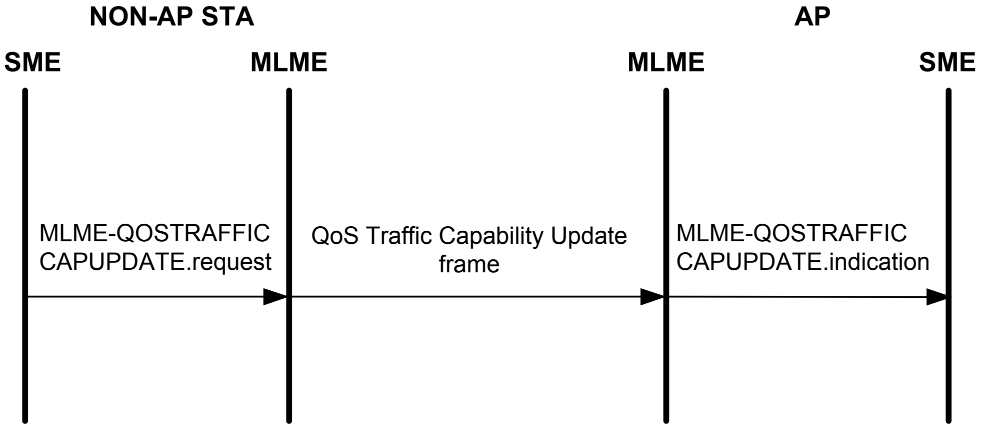

NON-AP STA
AP
SME
MLME
MLME
SME
MLME-QOSTRAFFIC CAPUPDATE.request
QoS Traffic Capability Update frame
MLME-QOSTRAFFIC CAPUPDATE.indication

Figure 6-23—QoS traffic capability update protocol exchange

6.3.65.1.2 Semantics of the service primitive

The primitive parameter is as follows:

```
MLME-QOSTRAFFICCAPUPDATE.request(
                                        QoSTrafficCapability
                                        )
```

Name	Type	Valid range	Description
QoS Traffic Capability	Bit field as defined in 8.5.14.22	As defined in 8.5.14.22	Specifies the QoS Traffic Capability flags of the non-AP STA. This parameter is present if dot11WirelessManagementImplemented is true and dot11MgmtOptionQoSTrafficCapabilityActivated is true, and is not present otherwise.

6.3.65.1.3 When generated

This primitive is generated by the SME to request that a QoS Traffic Capability Update frame be sent to the AP with which the STA is associated.

6.3.65.1.4 Effect of receipt

On receipt of this primitive, the MLME constructs a QoS Traffic Capability Update action management frame. The STA then attempts to transmit this to the AP with which it is associated.

6.3.65.2 MLME-QOSTRAFFICCAPUPDATE.indication

6.3.65.2.1 Function

This primitive indicates that a QoS Traffic Capability Update frame was received from a non-AP STA.

6.3.65.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-QOSTRAFFICCAPUPDATE.indication(
                                        PeerSTAAddress,
                                        QoS Traffic Capability
                                        )
```

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC Address	The address of the non-AP STA MAC entity from which a QoS Traffic Capability Update frame was received.
QoS Traffic Capability	Bit field as defined in 8.5.14.22	As defined in 8.5.14.22	Specifies the QoS Traffic Capability flags of the non-AP STA. This parameter is present if dot11WirelessManagementImplemented is true and dot11MgmtOptionQoSTrafficCapabilityActivated is true, and is not present otherwise.

6.3.65.2.3 When generated

This primitive is generated by the MLME when a valid QoS Traffic Capability Update action management frame is received.

6.3.65.2.4 Effect of receipt

On receipt of this primitive the SME should operate according to the procedure in 10.23.9.

6.3.66 Channel Usage request

6.3.66.1 General

The following MLME primitives support the signaling of Channel Usage request. Figure 6-24 depicts the Channel Usage request process. The figure illustrates the basic protocol and is only an example and therefore is not meant to be exhaustive of all possible protocol uses.

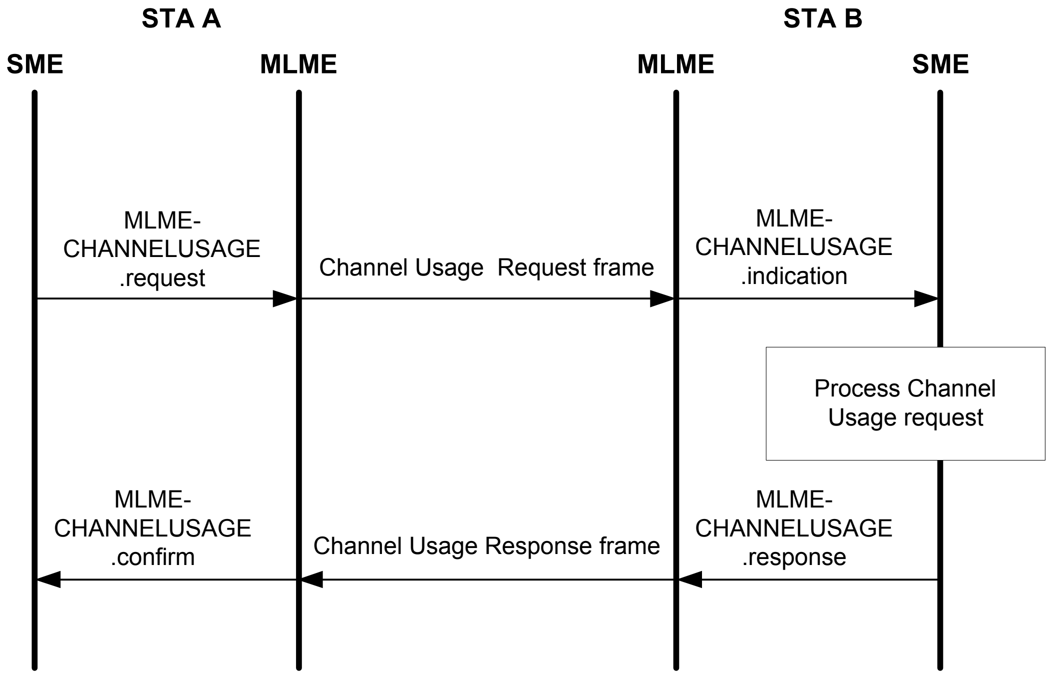

STA A
STA B
SME
MLME
MLME
SME
MLME-CHANNELUSAGE.request
Channel Usage Request frame
MLME-CHANNELUSAGE.indication
Process Channel Usage request
MLME-CHANNELUSAGE.confirm
Channel Usage Response frame
MLME-CHANNELUSAGE.response

Figure 6-24—Channel usage request protocol exchange

6.3.66.2 MLME-CHANNELUSAGE.request

6.3.66.2.1 Function

This primitive requests the transmission of a Channel Usage Request frame be sent to an AP.

6.3.66.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-CHANNELUSAGE.request(
                                        PeerSTAAddress,
                                        Dialog Token,
                                        ChannelUsage,
                                        SSID,
                                        SupportedOperatingClasses,
                                        VendorSpecificInfo
                                        )
```

Name	Type	Valid range	Description
PeerSTAAddress	MAC Address	Any valid individual MAC Address	Specifies the address of the peer MAC entity with which to perform the Channel Usage process.
Dialog Token	Integer	1–255	The Dialog Token to identify the Channel Usage request and response transaction.
ChannelUsage	A set of Channel Usage element	As defined in 8.4.2.88	Specifies request types for the Channel Usage request.
SSID	Octet string	0–32 octets	Specifies the desired SSID or the wildcard SSID.
SupportedOperating Classes	As defined in Supported Operating Classes element	As defined in 8.4.2.56	Specifies the Supported Operating Classes information for the Channel Usage Request.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.66.2.3 When generated

This primitive is generated by the SME to request that a Channel Usage Request frame be sent to the BSS which is advertising the SSID passed down in this primitive.

6.3.66.2.4 Effect of receipt

On receipt of this primitive, the MLME constructs a Channel Usage Request action management frame. The STA then attempts to transmit this to the BSS which is advertising the SSID included in this primitive request. When wild card SSID is passed down, the MLME-CHANNELREQUEST.request shall be transmitted to all BSSs in the current scan list as determined by the most recent MLME-SCAN.request.

6.3.66.3 MLME-CHANNELUSAGE.confirm

6.3.66.3.1 Function

This primitive reports the result of a Channel Usage procedure.

6.3.66.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-CHANNELUSAGE.confirm(
                                        PeerSTAAddress,
                                        Dialog Token,
                                        ChannelUsage,
                                        SSID,
                                        Country String,
                                        Power Constraint,
                                        VendorSpecificInfo
                                        )
```

Name	Type	Valid range	Description
PeerSTAAddress	MAC Address	Any valid individual MAC Address	Specifies the address of the peer MAC entity with which to perform the Channel Usage process.
Dialog Token	Integer	1–255	The Dialog Token to identify the Channel Usage request and response transaction.
ChannelUsage	A set of Channel Usage element	As defined in 8.4.2.88	Specifies parameters for the Channel Usage.
SSID	Octet string	0–32 octets	Specifies the SSID or the wildcard SSID used in the request.
Country String	Octet string	3 octets	Specifies Country strings.
Power Constraint	An element	As defined in 8.4.2.16	Zero or one Power Constraint element.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.66.3.3 When generated

This primitive is generated by the MLME as a result of an MLME-CHANNELUSAGE.request and indicates the results of the request.

This primitive is generated when the MLME-CHANNELUSAGE.request contains invalid parameters, when a timeout or failure occurs, or when the STA receives a Channel Usage Response frame from the AP.

6.3.66.3.4 Effect of receipt

On receipt of this primitive the SME should operate according to the procedure in 10.23.14.

6.3.66.4 MLME-CHANNELUSAGE.indication

6.3.66.4.1 Function

This primitive indicates that a Channel Usage Request frame was received from a non-AP STA.

6.3.66.4.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-CHANNELUSAGE.indication(
                                        PeerSTAAddress,
                                        Dialog Token,
                                        ChannelUsage,
                                        SSID,
                                        SupportedOperatingClasses,
                                        VendorSpecificInfo
                                        )
```

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC Address	The address of the non-AP STA MAC entity from which a Channel Usage Request frame was received.
Dialog Token	Integer	0–255	The Dialog Token to identify the Channel Usage request and response transaction.

Name	Type	Valid range	Description
ChannelUsage	A set of Channel Usage element	As defined in 8.4.2.88	Specifies request types for the Channel Usage request.
SSID	Octet string	0–32 octets	Specifies the desired SSID or the wildcard SSID.
SupportedOperating Classes	As defined in Supported Operating Classes element	As defined in 8.4.2.56	Specifies the Supported Operating Classes information for the Channel Usage Request.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.66.4.3 When generated

This primitive is generated by the MLME when a valid Channel Usage Request frame is received.

6.3.66.4.4 Effect of receipt

On receipt of this primitive the SME should operate according to the procedure in 10.23.14.

6.3.66.5 MLME-CHANNELUSAGE.response

6.3.66.5.1 Function

This primitive is generated in response to an MLME-CHANNELUSAGE.indication requesting a Channel Usage Response frame be sent to a non-AP STA.

6.3.66.5.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-CHANNELUSAGE.response(
                                        PeerSTAAddress,
                                        Dialog Token,
                                        ChannelUsage,
                                        SSID,
                                        Country String,
                                        Power Constraint,
                                        VendorSpecificInfo
                                        )
```

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC Address	The address of the non-AP STA MAC entity from which a Channel Usage Request frame was received.
Dialog Token	Integer	0–255	The Dialog Token to identify the Channel Usage request and response transaction.
ChannelUsage	A set of Channel Usage elements	As defined in 8.4.2.88	Specifies parameters for the Channel Usage.
SSID	Octet string	0–32 octets	Specifies the desired SSID or the wildcard SSID.
Country String	Octet string	3 octets	Specifies Country strings.
Power Constraint	An element	As defined in 8.4.2.16	Zero or one Power Constraint element.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.66.5.3 When generated

This primitive is generated by the SME in response to an MLME-CHANNELUSAGE.indication requesting a Channel Usage Response be sent to a non-AP STA.

6.3.66.5.4 Effect of receipt

On receipt of this primitive, the MLME constructs a Channel Usage Response frame. The STA then attempts to transmit this to the non-AP STA indicated by the PeerSTAAddress parameter.

6.3.67 DMS request and response procedure

6.3.67.1 General

The following MLME primitives support the signaling of DMS request and response procedure. The informative diagram shown in Figure 6-25 depicts the DMS request and response process and is not meant to be exhaustive of all possible protocol uses.

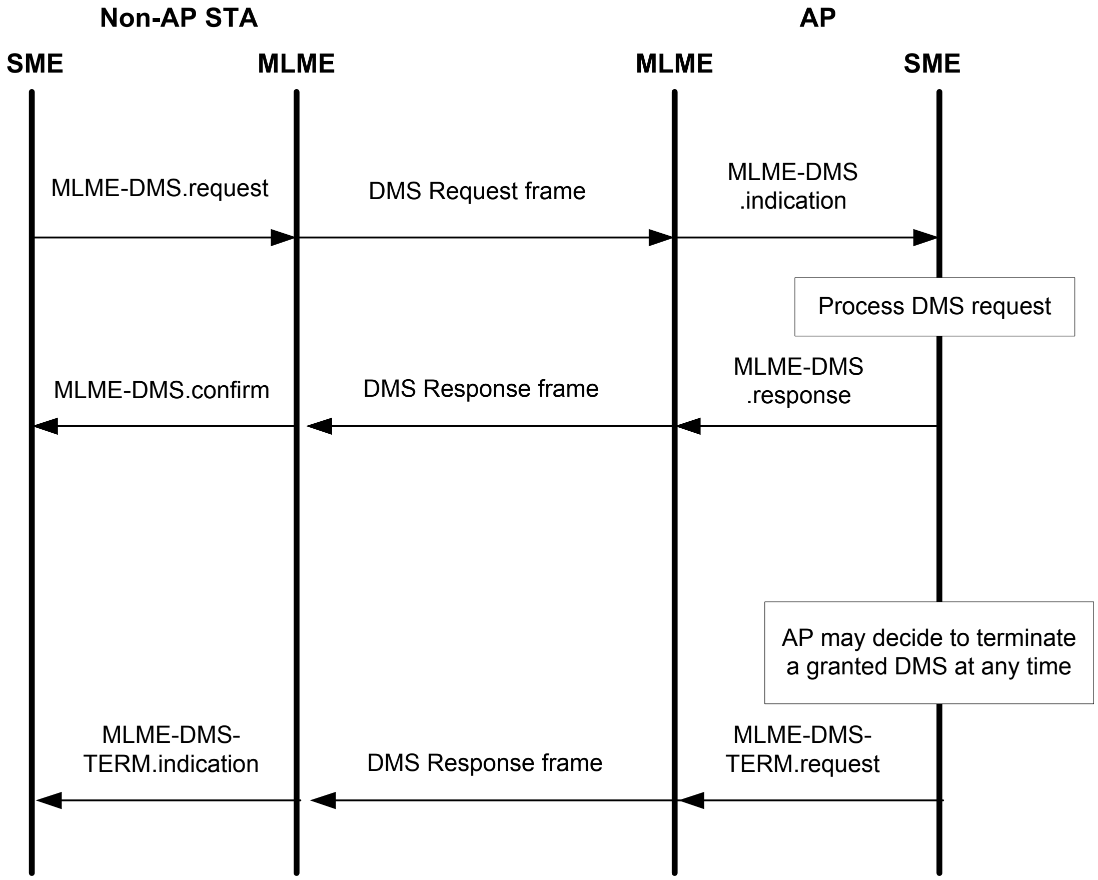


Figure 6-25—DMS setup protocol exchange

6.3.67.2 MLME-DMS.request

6.3.67.2.1 Function

This primitive requests the transmission of a DMS Request frame be sent to an AP.

6.3.67.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-DMS.request(
                          PeerSTAAddress,
                          Dialog Token,
                          DMSRequest
                          )
```

Name	Type	Valid range	Description
PeerSTAAddress	MAC Address	Any valid individual MAC Address	Specifies the address of the peer MAC entity with which to perform the DMS process.
Dialog Token	Integer	1–255	The Dialog Token to identify the DMS request and response transaction.
DMSRequest	DMS Request element	As defined in 8.4.2.90	Specifies group addressed frames for the requested DMS stream.

6.3.67.2.3 When generated

This primitive is generated by the SME to request that a DMS Request frame be sent to the AP with which the STA is associated.

6.3.67.2.4 Effect of receipt

On receipt of this primitive, the MLME constructs a DMS Request action management frame. The STA then attempts to transmit this to the AP with which the STA is associated.

6.3.67.3 MLME-DMS.confirm

6.3.67.3.1 Function

This primitive reports the result of a DMS procedure.

6.3.67.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-DMS.confirm(
                                PeerSTAAddress,
                                Dialog Token,
                                DMSResponse
                                )
```

Name	Type	Valid range	Description
PeerSTAAddress	MAC Address	Any valid individual MAC Address	Specifies the address of the peer MAC entity with which to perform the DMS process.
Dialog Token	Integer	1–255	The Dialog Token to identify the DMS request and response transaction.
DMSResponse	DMS Response element	As defined in 8.4.2.91	Specifies the status returned by the AP responding to the STA's requested DMS stream.

6.3.67.3.3 When generated

This primitive is generated by the MLME as a result of an MLME-DMS.request and indicates the results of the request.

This primitive is generated when the MLME-DMS.request contains invalid parameters, when a timeout or failure occurs, or when the STA receives a DMS Response frame from the AP.

6.3.67.3.4 Effect of receipt

On receipt of this primitive the SME should operate according to the procedure in 10.23.15.

6.3.67.4 MLME-DMS.indication

6.3.67.4.1 Function

This primitive indicates that a DMS Request frame was received from a non-AP STA.

6.3.67.4.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-DMS.indication(
                                PeerSTAAddress,
                                Dialog Token,
                                DMSRequest
                                )
```

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC Address	The address of the non-AP STA MAC entity from which a DMS Request frame was received.
Dialog Token	Integer	1–255	The Dialog Token to identify the DMS request and response transaction.
DMSRequest	DMS Request element	As defined in 8.4.2.90	Specifies group addressed frames for the requested DMS stream.

6.3.67.4.3 When generated

This primitive is generated by the MLME when a valid DMS Request frame is received.

6.3.67.4.4 Effect of receipt

On receipt of this primitive the SME should operate according to the procedure in 10.23.15.

6.3.67.5 MLME-DMS.response

6.3.67.5.1 Function

This primitive is generated in response to an MLME-DMS.indication requesting a DMS Response frame be sent to a non-AP STA.

6.3.67.5.2 Semantics of the service primitive

The primitive parameters are as follows:

```
          MLME-DMS.response(
                              PeerSTAAddress,
                              Dialog Token,
                              DMSResponse
                              )
```

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC Address	The address of the non-AP STA MAC entity from which a DMS Request frame was received.
Dialog Token	Integer	1–255	The Dialog Token to identify the DMS request and response transaction.
DMSResponse	DMS Response element	As defined in 8.4.2.91	Specifies the status returned by the AP responding to the STA’s requested DMS stream.

6.3.67.5.3 When generated

This primitive is generated by the SME in response to an MLME-DMS.indication requesting a DMS Response be sent to a non-AP STA.

6.3.67.5.4 Effect of receipt

On receipt of this primitive, the MLME constructs a DMS Response frame. The STA then attempts to transmit this to the non-AP STA indicated by the PeerSTAAddress parameter.

6.3.67.6 MLME-DMS-TERM.request

6.3.67.6.1 Function

This primitive requests the transmission of a DMS Response frame to non-AP STAs to terminate a granted DMS.

6.3.67.6.2 Semantics of the service primitive

The primitive parameters are as follows:

```
          MLME-DMS-TERM.request(
```

PeerSTAAddress,
Dialog Token,
DMSResponse
)

Name	Type	Valid range	Description
PeerSTAAddress	MACAddress	Any valid individual MAC Address	The address of the non-AP STA MAC entity from which a DMS Request frame was received.
Dialog Token	Integer	0	Set to 0 for an autonomous DMS Response frame.
DMSResponse	DMS Response element	As defined in 8.4.2.91	Specifies the requested DMS stream that is cancelled by the AP.

6.3.67.6.3 When generated

This primitive is generated by the SME to terminate DMS.

6.3.67.6.4 Effect of receipt

On receipt of this primitive, the MLME constructs a DMS Response frame. The STA then attempts to transmit this to the non-AP STA indicated by the PeerSTAAddress parameter.

6.3.67.7 MLME-DMS-TERM.indication

6.3.67.7.1 Function

This primitive is generated by the MLME when a valid unsolicited DMS Response frame is received.

6.3.67.7.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-DMS-TERM.indication(
PeerSTAAddress,
Dialog Token,
DMSResponse
)

Name	Type	Valid range	Description
PeerSTAAddress	MAC Address	Any valid individual MAC Address	Specifies the address of the peer MAC entity with which to perform the DMS process.
Dialog Token	Integer	0	Set to 0 for an autonomous DMS Response frame.
DMSResponse	DMS Response element	As defined in 8.4.2.91	Specifies the requested DMS stream that is cancelled by the AP.

6.3.67.7.3 When generated

This primitive is generated when the STA receives an unsolicited DMS Response frame from the AP.

6.3.67.7.4 Effect of receipt

On receipt of this primitive the SME should operate according to the procedure in 10.23.15.

6.3.68 Timing Measurement Request

6.3.68.1 General

The following set of primitives supports triggering a Timing Measurement procedure or stopping an ongoing Timing Measurement procedure.

6.3.68.2 MLME-TIMINGMSMTRQ.request

6.3.68.2.1 Function

This primitive requests the transmission of a Timing Measurement Request frame to a peer entity.

6.3.68.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-TIMINGMSMTRQ.request(
                                      Peer MAC Address,
                                      Trigger
                                      )
```

Name	Type	Valid range	Description
PeerSTAAddress	MAC Address	Any valid individual MAC Address	The address of the peer MAC entity to which the Timing Measurement Request frame is sent.
Trigger	Integer	0–1	The trigger to identify the action.

6.3.68.2.3 When generated

This primitive is generated by the SME to request that a Timing Measurement Request frame be sent to a peer entity.

6.3.68.2.4 Effect of receipt

On receipt of this primitive, the MLME constructs a Timing Measurement Request frame with the specified parameters. This frame is then scheduled for transmission.

6.3.68.3 MLME-TIMINGMSMTRQ.indication

6.3.68.3.1 Function

This primitive indicates that a Timing Measurement Request frame has been received and the corresponding ACK has been transmitted.

6.3.68.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-TIMINGMSMTRQ.indication(
                                 Peer MAC Address,
                                 Trigger
                                 )
```

Name	Type	Valid range	Description
PeerSTAAddress	MAC Address	Any valid individual MAC Address	The address of the peer MAC entity to which the Timing Measurement Request frame is sent.
Trigger	Integer	0–1	The trigger to identify the action.

6.3.68.3.3 When generated

This primitive is generated by the MLME when a valid Timing Measurement Request frame is received.

6.3.68.3.4 1 Effect of receipt

On receipt of this primitive, the SME uses the information contained within the notification.

6.3.69 WNM-Notification request

6.3.69.1 General

This set of primitives supports the exchange of WNM-Notification Request and Response frames between peer SMEs.

6.3.69.2 MLME-WNMNOTIFICATIONREQUEST.request

6.3.69.2.1 Function

This primitive requests the transmission of a WNM-Notification Request frame to a peer entity.

6.3.69.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-WNMNOTIFICATIONREQUEST.request(
                                              Peer MAC Address,
                                              Dialog Token,
                                              Type,
                                              Optional Subelements
                                              )
```

Name	Type	Valid range	Description
PeerSTAAddress	MAC Address	Any valid individual MAC Address	The address of the peer MAC entity to which the WNM-Notification Request frame is sent.
Dialog Token	Integer	1–255	The dialog token to identify the WNM-Notification transaction.
Type	Integer	1–255	The type of WNM-Notification.
Optional Subelements	Set subelements	As defined in 8.5.14.28	A set of subelements describing the notification.

6.3.69.2.3 When generated

This primitive is generated by the SME to request that a WNM-Notification Request frame be sent to a peer entity to initiate a WNM-Notification.

6.3.69.2.4 Effect of receipt

On receipt of this primitive, the MLME constructs a WNM-Notification Request frame containing the elements specified. This frame is then scheduled for transmission.

6.3.69.3 MLME-WNMNOTIFICATIONREQUEST indication

6.3.69.3.1 Function

This primitive indicates that a WNM-Notification Request frame has been received.

6.3.69.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
   MLME-WNMNOTIFICATIONREQUEST.indication(
                                          Peer MAC Address,
                                          Dialog Token,
                                          Type,
                                          Optional Subelements
                                          )
```

Name	Type	Valid range	Description
PeerSTAAddress	MAC Address	Any valid individual MAC Address	The address of the peer MAC entity from which the WNM-Notification request was received.
Dialog Token	Integer	1–255	The dialog token to identify the WNM-Notification transaction.
Type	Integer	1–255	The type of WNM-Notification Request.
Optional Subelements	Set subelements	As defined in 8.5.14.28	A set of subelements describing the notification.

6.3.69.3.3 When generated

This primitive is generated by the MLME when a valid WNM-Notification Request frame is received.

6.3.69.3.4 Effect of receipt

On receipt of this primitive, the SME replies to the request.

6.3.70 WNM-Notification response

6.3.70.1 MLME-WNMNOTIFICATIONRESPONSE.request

6.3.70.1.1 Function

This primitive supports the signaling of the WNM-Notification Response frame between peer SMEs.

6.3.70.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```
   MLME-WNMNOTIFICATIONRESPONSE.request(
```

Peer MAC Address,
Dialog Token,
Response Status,
Optional Subelements
)

Name	Type	Valid range	Description
PeerSTAAddress	MAC Address	Any valid individual MAC Address	The address of the peer MAC entity to which the WNM-Notification Response frame is sent.
Dialog Token	Integer	1–255	The dialog token to identify the WNM-Notification transaction.
Response Status	Integer	1–255	The response status of the WNM-Notification.
Optional Subelements	Set subelements	As defined in 8.5.14.29	A set of subelements describing the results of the WNM-Notification.

6.3.70.1.3 When generated

This primitive is generated by the SME to request that a WNM-Notification Response frame be sent to a peer entity to report the results of the WNM-Notification.

6.3.70.1.4 Effect of receipt

On receipt of this primitive, the MLME constructs a WNM-Notification Response frame containing the indicated fields. This frame is then scheduled for transmission.

6.3.70.2 MLME-WNMNOTIFICATIONRESPONSE.indication

6.3.70.2.1 Function

This primitive indicates that a WNM-Notification Response frame has been received from a peer entity.

6.3.70.2.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-WNMNOTIFICATIONRESPONSE.indication(
Peer MAC Address,
Dialog Token,
Response Status,
Optional Subelements
)

Name	Type	Valid range	Description
PeerSTAAddress	MAC Address	Any valid individual MAC Address	The address of the peer MAC entity from which the WNM-Notification Response frame was received.
Dialog Token	Integer	1–255	The dialog token to identify the WNM-Notification transaction.
Response Status	Integer	1–255	The response status of the WNM-Notification.
Optional Subelements	Set subelements	As defined in 8.5.14.29	A set of subelements describing the results of the WNM-Notification.

6.3.70.2.3 When generated

This primitive is generated by the MLME when a valid WNM-Notification Response frame is received.

6.3.70.2.4 Effect of receipt

On receipt of this primitive, the WNM-Notification can be made available for SME processes.

6.3.71 Network discovery and selection support

6.3.71.1 General

This set of primitives supports the process of GAS.

6.3.71.2 MLME-GAS.request

6.3.71.2.1 Function

This primitive requests the information of a specific advertisement service from another STA and requests the STA to provide GAS.

6.3.71.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-GAS.request(
                        PeerSTAAddress,
                        DialogToken,
                        AdvertisementProtocolID,
                        Query,
                        QueryFailureTimeout,
                        Protected
                        )
```

Name	Type	Valid range	Description
PeerSTAAddress	MacAddress	Any valid individual MacAddress	Specifies the address of the peer MAC entity to which query is transmitted.
DialogToken	Integer	0–255	The dialog token to identify the GAS transaction.
AdvertisementProtocolID	Integer or Sequence of Integers	As defined in Table 8-175	This contains an Advertisement Protocol ID (see 8.4.2.95), which may be IEEE 802.11 assigned or vendor specified.
Query	String	N/A	Query string formatted using protocol identified in AdvertisementProtocolID. E.g., if the AdvertisementProtocolID value is 1, then Query is formatted as defined in IEEE Std 802.21-2008.
QueryFailureTimeout	Integer	> 1	The time limit, in units of Beacon intervals, after which the GAS Query procedure is terminated.
Protected	Boolean	true, false	Specifies whether the request is sent using a Robust Management frame. If true, the request is sent using a Protected Dual of Public Action frame. Otherwise, the request is sent using a Public Action frame.

6.3.71.2.3 When generated

This primitive is generated by the SME at a STA to request a specific Advertisement Service from another STA.

6.3.71.2.4 Effect of receipt

The STA operates according to the procedures defined in 10.24.3.

6.3.71.3 MLME-GAS.confirm

6.3.71.3.1 Function

This primitive reports the status code and Query Response from an Advertisement Server to the requesting STA.

6.3.71.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-GAS.confirm(
                          PeerSTAAddress,
                          DialogToken,
                          ResultCode,
                          ResponseInfo,
                          Protected
                          )
```

Name	Type	Valid range	Description
PeerSTAAddress	MacAddress	Any valid individual MacAddress	Specifies the address of the peer MAC entity to which query is transmitted.
DialogToken	Integer	0–255	The dialog token to identify the GAS transaction.
ResultCode	Enumeration	SUCCESS, NO_OUTSTANDING_GAS_REQUEST, GAS_ADVERTISEMENT_PROTOCOL_NOT_SUPPORTED, GAS_QUERY_RESPONSE_OUTSTANDING, GAS_QUERY_RESPONSE_TOO_LARGE, SERVER_UNREACHABLE, GAS_QUERY_TIMEOUT, GAS_RESPONSE_NOT_RECEIVED_FROM_SERVER	Indicates the result response to the GAS request from the peer MAC entity.

Name	Type	Valid range	Description
ResponseInfo	String	N/A	Query Response string formatted using protocol identified in AdvertisementProtocolID. E.g., if the AdvertisementProtocolID value is 1, then Query is formatted as defined in IEEE Std 802.21-2008.
Protected	Boolean	true, false	Specifies whether the response was received in a Robust Management frame. If true, the response was received using a Protected Dual of Public Action frame. Otherwise, the response was received using a Public Action frame.

The mapping of Status Code received in the GAS Response frame is mapped to the corresponding Result Code in Table 8-37.

6.3.71.3.3 When generated

This primitive is generated by the MLME as a response to the MLME-GAS.request primitive indicating the result of that request.

The primitive is generated when the requesting STA receives a query response in a (Protected) GAS Initial Response frame or one or more (Protected) GAS Comeback Response frames.

6.3.71.3.4 Effect of receipt

The STA operates according to the procedures defined in 10.24.3.

6.3.71.4 MLME-GAS.indication

6.3.71.4.1 Function

This primitive reports to the STA’s SME about the GAS Request.

6.3.71.4.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-GAS.indication(
                                PeerSTAAddress,
                                DialogToken,
                                AdvertisementProtocolID,
                                Query,
                                Protected
                                )
```

Name	Type	Valid range	Description
PeerSTAAddress	MacAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity from which the query message was received.
DialogToken	Integer	0–255	The dialog token to identify the GAS transaction.
AdvertisementProtocolID	Integer or Sequence of Integers	As defined in Table 8-175	This contains an Advertisement Protocol ID (see 8.4.2.95), which may be IEEE 802.11 assigned or vendor specified.
Query	String	N/A	Query string formatted using protocol identified in AdvertisementProtocolID. E.g., if the AdvertisementProtocolID value is 1, then Query is formatted as defined in IEEE Std 802.21-2008.
Protected	Boolean	true, false	Specifies whether the request was received in a Robust Management frame. If true, the request was received in a Protected Dual of Public Action frame. Otherwise, the request was received in a Public Action frame.

6.3.71.4.3 When generated

This primitive is generated by the MLME as a result of receipt of a GAS request from STA.

6.3.71.4.4 Effect of receipt

The SME is notified of the request from the STA.

The SME operates according to the procedures defined in 10.24.3.

The SME generates an MLME-GAS.response primitive within a dot11GASResponseTimeout.

6.3.71.5 MLME-GAS.response

6.3.71.5.1 Function

This primitive responds to the request for an advertisement service by a specified STA MAC entity.

6.3.71.5.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-GAS.response(
                                 PeerSTAAddress,
                                 DialogToken,
                                 ResultCode,
                                 ResponseInfo,
                                 Protected
                                 )
```

Name	Type	Valid range	Description
PeerSTAAddress	MacAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity to which query response information is transmitted.

Name	Type	Valid range	Description
DialogToken	Integer	0–255	The dialog token to identify the GAS transaction.
ResultCode	Enumeration	SUCCESS, NO_OUTSTANDING_GAS_REQUEST, GAS_ADVERTISEMENT_PROTOCOL_NOT_SUPPORTED, GAS_QUERY_RESPONSE_OUTSTANDING, GAS_QUERY_RESPONSE_TOO_LARGE, SERVER_UNREACHABLE, GAS_QUERY_TIMEOUT, GAS_RESPONSE_NOT_RECEIVED_FROM_SERVER	Indicates the result response to the GAS-request from the peer MAC entity. See Table 8-37.
ResponseInfo	String	N/A	Query Response string formatted using protocol identified in AdvertisementProtocolID. E.g., if the AdvertisementProtocolID value is 1, then Query is formatted as defined in IEEE Std 802.21-2008.
Protected	Boolean	true, false	Specifies whether the response is sent using a Robust Management frame. If true, the response is sent using a Protected Dual of Public Action frame. Otherwise, the response is sent using a Public Action frame.

6.3.71.5.3 When generated

This primitive is generated by the MLME at a STA as a result of an MLME-GAS.indication primitive.

6.3.71.5.4 Effect of receipt

This primitive causes the MAC entity at the STA to send a (Protected) GAS Initial Response frame to the requesting STA and optionally one or more (Protected) GAS Comeback Response frames.

6.3.72 QoS Map Set element management

6.3.72.1 General

The QoS Map Set element is provided to non-AP STAs in (Re)Association Response frames. However, if the SME of an AP detects a change of the QoS Map information while one or more non-AP STAs are associated to the BSS, then the AP may transmit an unsolicited QoS Map Set element to associated STAs. The AP's SME invokes the MLME-QoSMap.request primitive to cause individually addressed frames containing a QoS Map Set element to be transmitted to associated STAs. The AP's SME invokes the MLME-QoSMap.request primitive to transmit individually addressed frames containing a QoS Map Set element to associated STAs. When a non-AP STA receives such unsolicited QoS Map information, its MLME generates a MLME-QoSMap.indication primitive to the STA's SME. In turn, the SME should take appropriate action, e.g., initiate an ADDTS or DELTS if admission control changes are necessary.

6.3.72.2 MLME-QoSMap.request

6.3.72.2.1 Function

This primitive is used by an AP to transmit an unsolicited QoS Map Set to a specified non-AP STA MAC entity. The specified non-AP STA MAC address is an individual MAC address.

6.3.72.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
        MLME-QoSMap.request(
                              Non-APSTAAddress,
                              QoSMapSet
                              )
```

Name	Type	Valid range	Description
Non-APSTAAddress	MacAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity from which query message is received.
QoSMapSet	As defined in frame format	As defined in 8.4.2.97	Specifies the QoS Map Set the non-AP STA should use.

6.3.72.2.3 When generated

This primitive is generated by the MLME at the AP as a result of any change in the AP QoS Map configurations.

6.3.72.2.4 Effect of receipt

This primitive causes the MAC entity at the AP to send a QoS MAP Set element in a QoS MAP Configure frame to the non-AP STA.

6.3.72.3 MLME-QoSMap.indication

6.3.72.3.1 Function

This primitive reports the QoS mapping information sent from the AP to the non-AP STA.

6.3.72.3.2 Semantics of the service primitive

The primitive parameter is as follows:

```
        MLME-QoSMap.indication(
                                  QoSMapSet
                                  )
```

Name	Type	Valid range	Description
QoSMapSet	As defined in frame format	As defined in 8.4.2.97	Specifies the QoS Map Set to be used by the non-AP STA.

6.3.72.3.3 When generated

This primitive is generated when the non-AP STA receives a QoS Map Set element in an unsolicited QoS Map Configure frame from the AP.

The SME of the non-AP STA should use the information to decide proper actions. For example, an ADDTS or DELTS procedure should be activated if the QoS Map information indicates a change in the admission control.

6.3.72.3.4 Effect of receipt

The non-AP STA operates according to the procedures defined in 10.24.9.

6.3.73 Mesh peering management

6.3.73.1 Introduction

The following primitives facilitate the mesh peering management protocol and authenticated mesh peering exchange protocol.

6.3.73.2 MLME-MESHPEERINGMANAGEMENT.request

6.3.73.2.1 Function

This primitive requests that the MAC entity establish, confirm, or close a mesh peering with the specified peer MAC entity by sending a Mesh Peering Management frame to the peer MAC entity. The mesh peering management procedures are specified in 13.3.

6.3.73.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MESHPEERINGMANAGEMENT.request(
                                        PeerMACAddress,
                                        MeshPeeringMgmtFrameContent
                                        )
```

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Valid individual MAC address	Specifies the address of the peer MAC entity to which the Mesh Peering Management frame is to be sent.
MeshPeeringMgmtFrameContent	Sequence of octets	As defined in 8.5.16.2, 8.5.16.3, or 8.5.16.4	The contents of the Action field of the Mesh Peering Open, Mesh Peering Confirm, or Mesh Peering Close frame to send to the peer MAC entity.

6.3.73.2.3 When generated

This primitive is generated by the SME to request that a Mesh Peering Management frame be sent to the specified mesh STA.

6.3.73.2.4 Effect of receipt

On receipt of this primitive, the MLME constructs a Mesh Peering Management frame containing the information specified. The frame is scheduled for transmission.

6.3.73.3 MLME-MESHPEERINGMANAGEMENT.confirm

6.3.73.3.1 Function

This primitive reports the results of a request to send a Mesh Peering Management frame.

6.3.73.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MESHPEERINGMANAGEMENT.confirm(
                                              PeerMACAddress,
                                              MeshPeeringMgmtFrameContent
                                              )
```

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Valid individual MAC address	Specifies the address of the peer MAC entity to which the Mesh Peering Management frame was sent.
MeshPeeringMgmtFrameContent	Sequence of octets	As defined in 8.5.16.2, 8.5.16.3, or 8.5.16.4	The contents of the Action field of the Mesh Peering Open, Mesh Peering Confirm, or Mesh Peering Close frame received from the peer MAC entity.

6.3.73.3.3 When generated

This primitive is generated as a result of an MLME-MESHPEERINGMANAGEMENT.request with a specified MAC peer.

6.3.73.3.4 Effect of receipt

The SME is notified of the results of the mesh peering management protocol request.

6.3.73.4 MLME-MESHPEERINGMANAGEMENT.indication

6.3.73.4.1 Function

This primitive indicates to the SME that the MLME has received a Mesh Peering Management frame from a peer MAC entity.

6.3.73.4.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MESHPEERINGMANAGEMENT.indication(
                                              PeerMACAddress,
                                              MeshPeeringMgmtFrameContent
                                              )
```

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Valid individual MAC address	Specifies the address of the peer MAC entity from which the Mesh Peering Management frame was received.
MeshPeeringMgmtFrameContent	Sequence of octets	As defined in 8.5.16.2, 8.5.16.3, or 8.5.16.4	The contents of the Action field of the Mesh Peering Open, Mesh Peering Confirm, or Mesh Peering Close frame received from the peer MAC entity.

6.3.73.4.3 When generated

This primitive is generated by the MLME as a result of the receipt of a Mesh Peering Management frame from a peer MAC entity.

6.3.73.4.4 Effect of receipt

The SME is notified of the reception of a Mesh Peering Management frame and is provided the contents of the frame.

6.3.73.5 MLME-MESHPEERINGMANAGEMENT.response

6.3.73.5.1 Function

This primitive is used to send a response to a Mesh Peering Management frame to the specified peer MAC entity.

6.3.73.5.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MESHPEERINGMANAGEMENT.response(
                                        PeerMACAddress,
                                        ResultCode,
                                        MeshPeeringMgmtFrameContent
                                        )
```

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Valid individual MAC address	Specifies the address of the peer MAC entity to which the Mesh Peering Management frame is to be sent.
ResultCode	Enumeration	SUCCESS, INVALID_PARAMETERS, UNSPECIFIED_FAILURE	Reports the result response to the Mesh Peering Management frame from the peer MAC entity.
MeshPeeringMgmtFrameContent	Sequence of octets	As defined in 8.5.16.2, 8.5.16.3, or 8.5.16.4	The contents of the Action field of the Mesh Peering Open, Mesh Peering Confirm, or Mesh Peering Close frame to send to the peer MAC entity.

6.3.73.5.3 When generated

This primitive is generated by the SME as a response to an MLME-MESHPEERINGMANAGEMENT.indication primitive.

6.3.73.5.4 Effect of receipt

This primitive indicates scheduling for transmission of a Mesh Peering management frame containing the indicated response.

6.3.74 Mesh power management

6.3.74.1 Introduction

The following primitives describe how a mesh entity changes its mesh power mode for a mesh peering.

6.3.74.2 MLME-MESHPOWERMGT.request

6.3.74.2.1 Function

This primitive requests a change in the mesh STAs mesh power mode for the mesh peering.

6.3.74.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
   MLME-MESHPOWERMGT.request(
                                        PeerMACAddress,
                                        Mesh Power Mode
                                        )
```

Name	Type	Valid Range	Description
PeerMACAddress	MAC Address	Valid individual MAC address	Specifies the address of the peer MAC entity to which the mesh power mode is changed.
Mesh Power Mode	Enumeration	ACTIVE_MODE, LIGHT_SLEEP_MODE, DEEP_SLEEP_MODE	Specifies the mesh power mode that the local mesh STA is using for the mesh peering.

6.3.74.2.3 When generated

The primitive is generated when the mesh entity wishes to change its mesh power mode for a mesh peering.

6.3.74.2.4 Effect of receipt

This primitive initiates the local mesh STA’s mesh power mode change for the mesh peering. The MLME subsequently issues an MLME-MESHPOWERMGT.confirm that reflects the results.

6.3.74.3 MLME-MESHPOWERMGT.confirm

6.3.74.3.1 Function

This primitive reports the result of a mesh power mode change attempt.

6.3.74.3.2 Semantics of the service primitive

The primitive parameter is as follows:

```
   MLME-MESHPOWERMGT.confirm(
                                        PeerMACAddress
                                        )
```

Name	Type	Valid Range	Description
PeerMAC Address	MAC Address	Valid individual MAC address	Specifies the address of the peer MAC entity to which the mesh power mode is changed.

6.3.74.3.3 When generated

This primitive is generated as a result of an MLME-MESHPOWERMGT.request.

6.3.74.3.4 Effect of receipt

The SME is notified of the results of the mesh power mode change for a mesh peering procedure.

6.3.75 Mesh neighbor offset synchronization

6.3.75.1 Introduction

This mechanism manages the neighbor offset synchronization method with the specified neighbor STA.

6.3.75.2 MLME-MESHNEIGHBOROFFSETSYNCSTART.request

6.3.75.2.1 Function

This primitive requests to start the neighbor offset synchronization method with the specified neighbor STA.

6.3.75.2.2 Semantics of the service primitive

The primitive parameter is as follows:

```
MLME-MESHNEIGHBOROFFSETSYNCSTART.request(
                                         PeerMACAddress
                                         )
```

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Any valid individual MAC address	Specifies the address of the peer MAC entity with which to start the neighbor offset synchronization method.

6.3.75.2.3 When generated

This primitive is generated by the SME to start the neighbor offset synchronization method with the specified neighbor STA.

6.3.75.2.4 Effect of receipt

On receipt of this primitive, the MLME commences the neighbor offset synchronization method and the calculation of the TSF timer offset value. The MLME subsequently issues an MLME-MESHNEIGHBOROFFSETSYNCSTART.confirm that reflects the results of this request.

6.3.75.3 MLME-MESHNEIGHBOROFFSETSYNCSTART.confirm

6.3.75.3.1 Function

This primitive reports the results of a mesh neighbor offset synchronization request.

6.3.75.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MESHNEIGHBOROFFSETSYNCSTART.confirm(
                                          PeerMACAddress,
                                          TSFOffsetValue
                                          )
```

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Valid individual MAC address	Specifies the address of the peer MAC entity to which the neighbor offset synchronization is requested.
TSFOffsetValue	Integer	-2^{63} to $(2^{63}-1)$	Indicates the TSF offset value with the specified neighbor STA, expressed in twos complement in µs.

6.3.75.3.3 When generated

This primitive is generated by the MLME as a result of an MLME-MESHNEIGHBOROFFSETSYNCSTART.request and report the TSF offset value.

6.3.75.3.4 Effect of receipt

The SME is notified of the results of the mesh neighbor offset synchronization request.

6.3.75.4 MLME-MESHNEIGHBOROFFSETCALCULATE.request

6.3.75.4.1 Function

This primitive requests a calculation result of the TSF timer offset value for the specified neighbor STA.

6.3.75.4.2 Semantics of the service primitive

The primitive parameter is as follows:

```
MLME-MESHNEIGHBOROFFSETCALCULATE.request(
                                          PeerMACAddress
                                          )
```

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Any valid individual MAC address	Specifies the address of the peer MAC entity with which to report the TSF offset value.

6.3.75.4.3 When generated

This primitive is generated by the SME to order a calculation of the TSF timer offset value with the specified neighbor STA.

6.3.75.4.4 Effect of receipt

On receipt of this primitive, the MLME receives a Beacon or Probe Response frame and calculates the TSF timer offset value from the received frame. The MLME tries to receive a Beacon frame immediately after the issue of MLME-MESHNEIGHBOROFFSETCALCULATE.request even if the mesh STA does not listen to the Beacon frame from the specified neighbor STA regularly (i.e., in deep sleep mode toward the specified neighbor STA). The MLME subsequently issues an MLME-MESHNEIGHBOROFFSETCALCULATE.confirm that reflects the results of this request.

6.3.75.5 MLME-MESHNEIGHBOROFFSETCALCULATE.confirm

6.3.75.5.1 Function

This primitive reports the results of a mesh neighbor offset calculation request.

6.3.75.5.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MESHNEIGHBOROFFSETCALCULATE.confirm(
                                          PeerMACAddress,
                                          TSFOffsetValue
                                          )
```

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Valid individual MAC address	Specifies the address of the peer MAC entity to which the Neighbor Offset Measure is requested.
TSFOffsetValue	Integer	-2^{63} to $(2^{63}-1)$	Indicates the TSF offset value with the specified neighbor STA, expressed in twos complement in µs.

6.3.75.5.3 When generated

This primitive is generated by the MLME as a result of an MLME-MESHNEIGHBOROFFSETCALCULATE.request to report a TSF offset value.

6.3.75.5.4 Effect of receipt

The SME is notified of the results of the mesh neighbor offset calculation request.

6.3.75.6 MLME-MESHNEIGHBOROFFSETSYNCSTOP.request

6.3.75.6.1 Function

This primitive requests to stop the neighbor offset synchronization method with the specified neighbor STA.

6.3.75.6.2 Semantics of the service primitive

The primitive parameter is as follows:

```
MLME-MESHNEIGHBOROFFSETSYNCSTOP.request(
                                          PeerMACAddress
                                          )
```

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Any valid individual MAC address	Specifies the address of the peer MAC entity with which to stop the neighbor offset synchronization method.

6.3.75.6.3 When generated

This primitive is generated by the SME to stop the maintenance of the neighbor offset synchronization method with the specified neighbor STA.

6.3.75.6.4 Effect of receipt

On receipt of this primitive, the MLME stops the neighbor offset synchronization method with the specified peer. The MLME subsequently issues an MLME-MESHNEIGHBOROFFSETSYNCSTOP.confirm that reflects the results of this request.

6.3.75.7 MLME-MESHNEIGHBOROFFSETSYNCSTOP.confirm

6.3.75.7.1 Function

This primitive reports the results of a neighbor offset synchronization method stop request.

6.3.75.7.2 Semantics of the service primitive

The primitive parameter is as follows:

```
   MLME-MESHNEIGHBOROFFSETSYNCSTOP.confirm(
                                       PeerMACAddress
                                       )
```

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Valid individual MAC address	Specifies the address of the peer MAC entity to which the Neighbor Offset Stop is requested.

6.3.75.7.3 When generated

This primitive is generated by the MLME as a result of an MLME-MESHNEIGHBOROFFSETSYNCSTOP.request.

6.3.75.7.4 Effect of receipt

The SME is notified of the results of the mesh neighbor offset synchronization stop request.

6.3.76 Mesh TBTT adjustment

6.3.76.1 Introduction

The following primitives describe how a mesh STA requests a TBTT adjustment from a neighboring peer mesh STA.

6.3.76.2 MLME-MESHTBTTADJUSTMENT.request

6.3.76.2.1 Function

This primitive requests transmission of a TBTT Adjustment Request frame.

6.3.76.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MESHTBTTADJUSTMENT.request(
                                PeerMACAddress,
                                BeaconTiming,
                                VendorSpecificInfo
                                )
```

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Any valid individual MAC address	Specifies the address of the peer MAC entity to which the TBTT Adjustment Request is sent.
BeaconTiming	A set of Beacon Timing elements	As defined in 8.4.2.107	A set of Beacon Timing elements of the mesh STA.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.76.2.3 When generated

This primitive is generated by the SME to request that a TBTT Adjustment Request frame be sent to a peer entity to request the adjustment of the peer entity’s TBTT.

6.3.76.2.4 Effect of receipt

On receipt of this primitive, the MLME constructs a TBTT Adjustment Request frame containing the Beacon Timing elements. This frame is then scheduled for transmission. The MLME subsequently issues an MLME-MESHTBTTADJUSTMENT.confirm that reflects the result of this request.

6.3.76.3 MLME-MESHTBTTADJUSTMENT.confirm

6.3.76.3.1 Function

This primitive reports the result of a mesh TBTT adjustment request.

6.3.76.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MESHTBTTADJUSTMENT.confirm(
                                PeerMACAddress,
                                ResultCode,
                                BeaconTiming,
                                VendorSpecificInfo
                                )
```

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Any valid individual MAC address	Specifies the address of the peer MAC entity from which the TBTT Adjustment Response is received.

Name	Type	Valid range	Description
ResultCode	Enumeration	SUCCESS, REFUSED_REASON_UNSPECIFIED, CANNOT_FIND_ALTERNATIVE_TBTT	Indicates the result of the TBTT adjustment request.
BeaconTiming	A set of Beacon Timing elements	As defined in 8.4.2.107	A set of Beacon Timing elements of the responding mesh STA. Present only when such an element was present in the TBTT Adjustment Response frame.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.76.3.3 When generated

This primitive is generated by the MLME as a result of an MLME-MESHTBTTADJUSTMENT.request primitive to indicate the result of that request.

6.3.76.3.4 Effect of receipt

The SME is notified of the result of the mesh TBTT adjustment request.

6.3.76.4 MLME-MESHTBTTADJUSTMENT.indication

6.3.76.4.1 Function

This primitive indicates that a specific peer MAC entity is requesting adjustment of the TBTT.

6.3.76.4.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MESHTBTTADJUSTMENT.indication(
                                   PeerMACAddress,
                                   BeaconTiming,
                                   VendorSpecificInfo
                                   )
```

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Any valid individual MAC address	Specifies the address of the peer MAC entity from which the TBTT Adjustment request was received.
BeaconTiming	A set of Beacon Timing elements	As defined in 8.4.2.107	A set of Beacon Timing elements of the requesting mesh STA.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.76.4.3 When generated

This primitive is generated by the MLME as a result of the receipt of a TBTT Adjustment Request frame from the specified peer MAC entity.

6.3.76.4.4 Effect of receipt

The SME is notified of the receipt of the TBTT adjustment request by the specified peer MAC entity. The mesh STA that received this primitive subsequently processes the TBTT scanning and adjustment procedure described in 13.13.4.4.3, and responds with the MLME-MESHTBTTADJUSTMENT.response.

6.3.76.5 MLME-MESHTBTTADJUSTMENT.response

6.3.76.5.1 Function

This primitive is used to send a response to the specified peer MAC entity that requested a TBTT adjustment from the mesh STA that issued this primitive.

6.3.76.5.2 Semantics of the service primitive

The primitive parameters are as follows:

```
  MLME-MESHTBTTADJUSTMENT.response(
                                              PeerMACAddress,
                                              Status Code,
                                              BeaconTiming,
                                              VendorSpecificInfo
                                              )
```

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Any valid individual MAC address	Specifies the address of the peer MAC entity to which the TBTT Adjustment Response is sent.
Status Code	As defined in frame format	SUCCESS, REFUSED_REASON_UNSPECIFIED, CANNOT_FIND_ALTERNATIVE_TBTT	Indicates the result response to the TBTT adjustment request from the peer mesh STA.
BeaconTiming	A set of Beacon Timing elements	As defined in 8.4.2.107	A set of Beacon Timing elements of the mesh STA. Present only when the STA could not find an alternative TBTT.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.76.5.3 When generated

This primitive is generated by the SME of a mesh STA as response to an MLME-MESHTBTTADJUSTMENT.indication primitive.

6.3.76.5.4 Effect of receipt

This primitive initiates the transmission of a TBTT Adjustment Response frame to the peer MAC entity that requested the TBTT adjustment.

On receipt of this primitive, the MLME constructs a TBTT Adjustment Response frame. This frame is then scheduled for transmission.

6.3.77 MCCA management interface

6.3.77.1 Introduction

The following primitives describe how a mesh entity manages its MCCA operation.

6.3.77.2 MLME-ACTIVATEMCCA.request

6.3.77.2.1 Function

This primitive requests that the MAC entity activate MCCA.

6.3.77.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-ACTIVATEMCCA.request(
                                    MCCAScanDuration,
                                    MAFLimit,
                                    MCCAAdvertPeriodMax,
                                    MCCAMaxTrackStates,
                                    MCCACWmin,
                                    MCCACWmax,
                                    MCCAAIFSN
                                    )
```

Name	Type	Valid range	Description
MCCAScanDuration	Integer	0 – 65 535	Specifies the duration in TUs that the mesh STA shall not initiate or accept MCCA Setup Request frames.
MAFLimit	Integer	0–255	Specifies the maximum MCCA access fraction allowed at the mesh STA. This number is always a multiple of (1/255) of the DTIM Interval.
MCCAAdvertPeriodMax	Integer	0–255	Specifies the maximum interval that a mesh STA with dot11MCCAActivated equal to true waits for an MCCAOP advertisement. It is expressed in number of DTIM intervals.
MCCAMaxTrackStates	Integer	dot11MCCAMinTrackStates – 65 535	Specifies the total number of MCCAOP reservations that the MAC entity is able to track.
MCCACWmin	Integer	0–15	Specifies the value of the minimum size of the contention window that the MAC entity uses for channel access during an MCCAOP.
MCCACWmax	Integer	0–63	Specifies the value of the maximum size of the contention that the MAC entity uses for channel access during an MCCAOP.
MCCAAIFSN	Integer	0–15	Specifies the value of the AIFSN that the MAC entity uses for channel access during an MCCAOP.

6.3.77.2.3 When generated

This primitive is generated by the SME to start the use of MCCA.

6.3.77.2.4 Effect of receipt

This primitive sets dot11MCCAActivated to true and initializes the MCCA parameters.

6.3.77.3 MLME-MCCASETUP.request

6.3.77.3.1 Function

This primitive requests that the MAC entity set up an MCCAOP reservation.

6.3.77.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
   MLME-MCCASETUP.request(
                                    MCCAOPDuration,
                                    MCCAOPPeriodicity,
                                    MCCAOPOffset,
                                    MCCAOPResponder,
                                    VendorSpecificInfo
                                    )
```

Name	Type	Valid range	Description
MCCAOPDuration	Integer	0 – 65 535	Specifies the MCCAOP Duration of the needed MCCAOPs as described in 8.4.2.108.2.
MCCAOPPeriodicity	Integer	0–255	Specifies the MCCAOP Periodicity of the needed MCCAOPs as described in 8.4.2.108.2.
MCCAOPOffset	Integer	0 – 16 777 215	Specifies the MCCAOP offset of the needed MCCAOPs as described in 8.4.2.108.2.
MCCAOPResponder	MAC address	Any valid individual or group MAC address	Specifies the MAC address of the intended MCCAOP responder.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.77.3.3 When generated

This primitive is generated by the SME to start an MCCAOP setup procedure.

6.3.77.3.4 Effect of receipt

This primitive causes the transmission of an MCCA Setup Request frame to the MCCAOP responder provided that the conditions for the transmission are met. The MLME subsequently issues an MLME-MCCASETUP.confirm primitive that reflects the results.

6.3.77.4 MLME-MCCASETUP.confirm

6.3.77.4.1 Function

This primitive is generated by the MLME to report the result of an MLME-MCCASETUP.request primitive, which was issued in order to establish an MCCAOP reservation with the peer MAC entity specified in MCCAOPResponder.

6.3.77.4.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MCCASETUP.confirm(
                                   MCCAOPParameters,
                                   MCCAOPID,
                                   MCCAOPResponder,
                                   ResultCode,
                                   VendorSpecificInfo
                                   )
```

Name	Type	Valid range	Description
MCCAOPParameters	MCCAOP Reservation	See 8.4.2.108.2	The MCCAOP reservation parameters.
MCCAOPID	Integer	0–254	MCCAOP reservation ID of the MCCAOP reservation.
MCCAOPResponder	MAC address	Any valid individual or group MAC address	Specifies the MAC address of the intended MCCAOP responder.
ResultCode	Enumeration	SUCCESS, INVALID_PARAMETERS, MCCAOP_RESERVATION_CONFLICT, MAF_LIMIT_EXCEEDED, MCCA_TRACK_LIMIT_EXCEEDED, MCCA_SETUP_TIMEOUT	Indicates the result of the MLME-MCCASETUP.request.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.77.4.3 When generated

This primitive is generated by the MLME as a result of an MLME-MCCASETUP.request to establish an MCCAOP reservation with the peer mesh STA identified in MCCAOPResponder or upon receipt of an MCCA Setup Reply frame from the peer mesh STA identified in MCCAOPResponder.

6.3.77.4.4 Effect of receipt

The SME is notified of the results of the MCCAOP setup procedure.

6.3.77.5 MLME-MCCASETUP.indication

6.3.77.5.1 Function

This primitive indicates the receipt of an MCCA Setup Request frame from the peer MAC entity specified in MCCAOPOwner.

6.3.77.5.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MCCASETUP.indication(
                                   MCCAOPParameters,
                                   MCCAOPID,
                                   MCCAOPOwner,
                                   VendorSpecificInfo
                                   )
```

Name	Type	Valid range	Description
MCCAOPParameters	MCCAOP Reservation	See 8.4.2.108.2	The MCCAOP reservation parameters.
MCCAOPID	Integer	0–254	MCCAOP reservation ID of the MCCAOP reservation.
MCCAOPOwner	MAC address	Any valid individual MAC address	Specifies the MAC address of the MCCAOP owner.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.77.5.3 When generated

This primitive is generated by the MLME as result of the receipt of an MCCA Setup Request frame from the peer MAC entity specified in MCCAOPOwner.

6.3.77.5.4 Effect of receipt

The SME is notified of the request to establish an MCCAOP reservation with the peer MAC entity specified in MCCAOPOwner.

6.3.77.6 MLME-MCCASETUP.response

6.3.77.6.1 Function

This primitive is used to send a response to the peer MAC entity specified in MCCAOPOwner that requested the set up of the MCCAOP reservation.

6.3.77.6.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MCCASETUP.response(
                                    MCCAOPParameters,
                                    MCCAOPID,
                                    MCCAOPOwner,
                                    ResultCode,
                                    VendorSpecificInfo
                                    )
```

Name	Type	Valid range	Description
MCCAOPParameters	MCCAOP Reservation	See 8.4.2.108.2	The MCCAOP reservation parameters.
MCCAOPID	Integer	0–254	MCCAOP reservation ID of the MCCAOP reservation.
MCCAOPOwner	MAC address	Any valid individual MAC address	Specifies the MAC address of the MCCAOP owner.
ResultCode	Enumeration	SUCCESS, INVALID_PARAMETERS, MCCAOP_RESERVATION_CONFLICT, MAF_LIMIT_EXCEEDED, MCCA_TRACK_LIMIT_EXCEEDED	Indicates the result of the MLME-MCCASETUP.request.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.77.6.3 When generated

This primitive is generated by the SME of a STA as a response to an MLME-MCCASETUP.indication procedure.

6.3.77.6.4 Effect of receipt

This primitive initiates transmission of a response to the peer MAC entity specified in the MCCAOPOwner that requested the set up of an MCCAOP reservation.

6.3.77.7 MLME-MCCAADVERTISEMENT.request

6.3.77.7.1 Function

This primitive requests that the MAC entity request an MCCAOP advertisement from the specified peer MAC entity by sending an MCCA Advertisement Request frame to the peer MAC entity.

6.3.77.7.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MCCAADVERTISEMENT.request(
                                    PeerMACAddress,
                                    VendorSpecificInfo
                                    )
```

Name	Type	Valid range	Description
PeerMACAddress	MAC address	Any valid individual MAC address	Specifies the MAC address of the peer MAC that will send the Advertisement.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.77.7.3 When generated

This primitive is generated by the SME to request an MCCAOP advertisement from the specified peer MAC entity.

6.3.77.7.4 Effect of receipt

This primitive causes the transmission of an MCCA Advertisement Request frame to the specified peer MAC entity. The MLME subsequently issues an MLME-MCCAADVERTISEMENT.confirm primitive that reflects the results.

6.3.77.8 MLME-MCCAADVERTISEMENT.confirm

6.3.77.8.1 Function

This primitive reports the result of an MLME-MCCAADVERTISEMENT.request.

6.3.77.8.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MCCAADVERTISEMENT.confirm(
```

MCCAOPAdvertisement,
PeerMACAddress,
ResultCode,
VendorSpecificInfo
)

Name	Type	Valid range	Description
MCCAOPAdvertisement	MCCAOP Advertisement	See 8.4.2.111	One or more MCCAOP Advertisement elements.
PeerMACAddress	MAC address	Any valid individual MAC address	Specifies the MAC address of the transmitter of the MCCAOP advertisement.
ResultCode	Enumeration	SUCCESS, REFUSED	Indicates the result of the MLME-MCCAADVERTISEMENT.request.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.77.8.3 When generated

This primitive is generated by the MLME as a result of an MLME-MCCAADVERTISEMENT.request.

6.3.77.8.4 Effect of receipt

The SME is notified of the results of the MCCA Advertisement Request frame.

6.3.77.9 MLME-MCCAADVERTISEMENT.indication

6.3.77.9.1 Function

This primitive reports that an MCCA Advertisement Request frame has been received from the specified peer MAC entity.

6.3.77.9.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-MCCAADVERTISEMENT.indication(
PeerMACAddress,
VendorSpecificInfo
)

Name	Type	Valid range	Description
PeerMACAddress	MAC address	Any valid individual MAC address	Specifies the MAC address of the transmitter of the MCCA Advertisement Request frame.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.77.9.3 When generated

This primitive is generated by the MLME upon receipt of an MCCA Advertisement Request frame from the specified peer MAC entity.

6.3.77.9.4 Effect of receipt

The SME is notified of the request to advertise its MCCAOP reservations.

6.3.77.10 MLME-MCCAADVERTISEMENT.response

6.3.77.10.1 Function

This primitive requests that the MAC entity respond to the MCCAOP advertisement request from the specified peer MAC entity by sending an MCCA Advertisement frame to the peer MAC entity.

6.3.77.10.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MCCAADVERTISEMENT.response(
                                        MCCAOPAdvertisement,
                                        PeerMACAddress,
                                        ResultCode,
                                        VendorSpecificInfo
                                        )
```

Name	Type	Valid range	Description
MCCAOPAdvertisement	MCCAOP Advertisement	See 8.4.2.111	One or more MCCAOP Advertisement elements.
PeerMACAddress	MAC address	Any valid individual MAC address	Specifies the MAC address of the transmitter of the MCCA Advertisement frame.
ResultCode	Enumeration	SUCCESS, REFUSED	Indicates the result of the MLME-MCCAADVERTISEMENT.response.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.77.10.3 When generated

This primitive is generated by the SME of a STA as a response to an MLME-MCCAADVERTISEMENT.indication procedure.

6.3.77.10.4 Effect of receipt

This primitive initiates transmission of a response to the specified peer MAC entity that requested advertisement of the MCCAOP reservations.

6.3.77.11 MLME-MCCATEARDOWN.request

6.3.77.11.1 Function

This primitive requests that the MAC entity tear down an MCCAOP reservation.

6.3.77.11.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MCCATEARDOWN.request(
```

MCCAOPID,
PeerMACAddress
)

Name	Type	Valid range	Description
MCCAOPID	Integer	0–255	Specifies the MCCAOP reservation ID of the MCCAOP reservation to be torn down.
PeerMACAddress	MAC address	Any valid individual MAC address	Specifies the MAC address of the peer MAC for the MCCAOP reservation.

6.3.77.11.3 When generated

This primitive is generated by the SME to start an MCCAOP teardown procedure.

6.3.77.11.4 Effect of receipt

This primitive causes the teardown MCCAOP reservation identified by means of the MCCAOP reservation ID in MCCAOPID, and the transmission of an MCCA Teardown frame to the peer MAC entity in PeerMACAddress.

6.3.77.12 MLME-MCCATEARDOWN.indication

6.3.77.12.1 Function

This primitive reports that an MCCA Teardown frame has been received from the specified peer MAC entity.

6.3.77.12.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-MCCATEARDOWN.indication(
MCCAOPID,
PeerMACAddress
)

Name	Type	Valid range	Description
MCCAOPID	Integer	0–255	MCCAOP reservation ID of the MCCAOP reservation to be torn down.
PeerMACAddress	MAC address	Any valid individual MAC address	Specifies the MAC address of the peer MAC for the MCCAOP reservation.

6.3.77.12.3 When generated

This primitive is generated by the MLME as result of receipt of a MCCA Teardown frame.

6.3.77.12.4 Effect of receipt

The SME is notified of the request to start an MCCAOP teardown procedure.

6.3.78 MBSS congestion control

6.3.78.1 Introduction

The following primitives describe how a mesh STA manages its congestion control operation.

6.3.78.2 MLME-MBSSCONGESTIONCONTROL.request

6.3.78.2.1 Function

This primitive requests that the MAC entity notify the peer MAC entity on the congestion level or requests to traffic generation by transmitting a Congestion Control Notification frame to the specified peer MAC entity.

6.3.78.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME- MBSSCONGESTIONCONTROL.request(
                    PeerMACAddress,
                    CongestionNotification,
                    VendorSpecificInfo
                    )
```

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Any valid individual MAC address	Specifies the address of the peer MAC entity to which the Congestion Control Notification frame is sent.
CongestionNotification	A set of Congestion Notification elements	As defined in 8.4.2.103	Congestion notification information generated by the mesh STA.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.78.2.3 When generated

The SME generates this primitive to request that the MAC notify its peer MAC about the current congestion level.

6.3.78.2.4 Effect of receipt

On receipt of this primitive, the MLME constructs a Congestion Control Notification frame. This frame is then scheduled for transmission.

6.3.78.3 MLME-MBSSCONGESTIONCONTROL.indication

6.3.78.3.1 Function

This primitive indicates that a congestion notification has been received.

6.3.78.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME- MBSSCONGESTIONCONTROL.indication(
                                        PeerMACAddress,
                                        CongestionNotification,
                                        VendorSpecificInfo
                                        )
```

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Any valid individual MAC address	Specifies the address of the peer MAC entity from which the Congestion Control Notification frame was received.
CongestionNotification	A set of Congestion Notification elements	As defined in 8.4.2.103	Congestion notification information contained in the received Congestion Control Notification frame.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.78.3.3 When generated

This primitive is generated by the MLME as a result of the receipt of a Congestion Control Notification frame from a specific peer MAC entity.

6.3.78.3.4 Effect of receipt

The SME is notified of the results of the receipt of the congestion control notification from the specified peer MAC entity. The mesh STA that received this primitive subsequently activates the local rate control as described in 13.12.

6.3.79 MBSS proxy update

6.3.79.1 Introduction

The following primitives describe how a mesh STA reports the proxy update information to another mesh STA in the MBSS.

6.3.79.2 MLME-MBSSPROXYUPDATE.request

6.3.79.2.1 Function

This primitive requests that the MAC entity inform a destination mesh STA about its proxy information by transmitting a Proxy Update frame to the specified peer MAC entity.

6.3.79.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME- MBSSPROXYUPDATE.request(
                                PeerMACAddress,
                                ProxyUpdate,
                                VendorSpecificInfo
                                )
```

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Any valid individual MAC address	Specifies the address of the peer MAC entity to which the Proxy Update frame is sent.
ProxyUpdate	A set of PXU elements	As defined in 8.4.2.118	A set of proxy information available at the mesh STA.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.79.2.3 When generated

This primitive is generated by the SME to request that a Proxy Update frame be sent to the specified peer MAC entity.

6.3.79.2.4 Effect of receipt

On receipt of this primitive, the MLME constructs a Proxy Update frame containing the PXU element. This frame is then scheduled for transmission. The MLME subsequently issues an MLME-MBSSPROXYUPDATE.confirm that reflects the results of this request.

6.3.79.3 MLME-MBSSPROXYUPDATE.confirm

6.3.79.3.1 Function

This primitive reports the results of a proxy update request.

6.3.79.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME- MBSSPROXYUPDATE.confirm(
                              PeerMACAddress,
                              ProxyUpdateConfirmation,
                              VendorSpecificInfo
                              )
```

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Any valid individual MAC address	Specifies the address of the peer MAC entity from which the Proxy Update Confirmation frame is received.
ProxyUpdateConfirmation	A set of PXUC elements	As defined in 8.4.2.119	A set of proxy update confirmation information from the peer MAC entity to which the Proxy Update frame was sent.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.79.3.3 When generated

This primitive is generated by the MLME as a result of an MLME-MBSSPROXYUPDATE.request primitive.

6.3.79.3.4 Effect of receipt

The SME is notified of the results of the MBSS proxy update request.

6.3.79.4 MLME-MBSSPROXYUPDATE.indication

6.3.79.4.1 Function

This primitive indicates that an update of the proxy information has been received from a specific peer MAC entity.

6.3.79.4.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME- MBSSPROXYUPDATE.indication(
                              PeerMACAddress,
                              ProxyUpdate,
                              VendorSpecificInfo
                              )
```

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Any valid individual MAC address	Specifies the address of the peer MAC entity from which the update of the proxy information was received.
ProxyUpdate	A set of PXU elements	As defined in 8.4.2.118	A set of proxy information received from the peer mesh STA.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.79.4.3 When generated

This primitive is generated by the MLME as a result of the receipt of a Proxy Update frame from a specific peer MAC entity.

6.3.79.4.4 Effect of receipt

The SME is notified of the results of the receipt of the proxy update request by the specified peer MAC entity. The mesh STA that received this primitive subsequently updates the proxy information as described in 13.11.4.3.

6.3.79.5 MLME-MBSSPROXYUPDATE.response

6.3.79.5.1 Function

This primitive is used to send a response to a specific peer MAC entity that sent an update of the proxy information to the mesh STA that issued this primitive.

6.3.79.5.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME- MBSSPROXYUPDATE.response(
                              PeerMACAddress,
                              ProxyUpdateConfirmation,
```

VendorSpecificInfo
)

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Any valid individual MAC address	Specifies the address of the peer MAC entity to which the Proxy Update Confirmation frame is sent.
ProxyUpdateConfirmation	A set of PXUC elements	As defined in 8.4.2.119	A set of proxy update confirmation information to be sent to the peer MAC entity.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.79.5.3 When generated

This primitive is generated by the SME of a STA as a response to an MLME-MBSSPROXYUPDATE.indication primitive.

6.3.79.5.4 Effect of receipt

This primitive initiates transmission of a response to the specific peer MAC entity that sent an update of the proxy information.

On receipt of this primitive, the MLME constructs a Proxy Update Confirmation frame. The frame contains one or more PXUC elements. This frame is then scheduled for transmission.

6.3.80 MBSS mesh gate announcement

6.3.80.1 Introduction

The following primitives describe how a mesh STA announces mesh gate reachability.

6.3.80.2 MLME-MBSSGATEANNOUNCEMENT.request

6.3.80.2.1 Function

This primitive requests that the MAC entity update the mesh gate information by transmitting a Gate Announcement frame to the specified MAC entity.

6.3.80.2.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME- MBSSGATEANNOUNCEMENT.request(
PeerMACAddress,
GateAnnouncement,
VendorSpecificInfo
)

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Any valid group MAC address	Specifies the address of the MAC entity to which the Gate announcement frame is sent.
GateAnnouncement	GANN element	As defined in 8.4.2.113	A set of gate announcement information to be sent through a Gate Announcement frame.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.80.2.3 When generated

This primitive is generated by the SME to request that a Gate Announcement frame be sent to the specified MAC entity.

6.3.80.2.4 Effect of receipt

On receipt of this primitive, the MLME constructs a Gate Announcement frame containing the GANN element. This frame is then scheduled for transmission following the interval specified by dot11MeshGateAnnouncementInterval.

6.3.80.3 MLME-MBSSGATEANNOUNCEMENT.indication

6.3.80.3.1 Function

This primitive indicates that a mesh gate announcement has been received from the specific peer MAC entity.

6.3.80.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME- MBSSGATEANNOUNCEMENT.indication(
                                    PeerMACAddress,
                                    GateAnnouncement,
                                    VendorSpecificInfo
                                    )
```

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Any valid individual MAC address	Specifies the address of the peer MAC entity from which the gate announcement was received.
GateAnnouncement	GANN element	As defined in 8.4.2.113	A set of gate announcement information contained in the received Gate Announcement frame.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.80.3.3 When generated

This primitive is generated by the MLME as a result of the receipt of a Gate Announcement frame from a specific peer MAC entity.

6.3.80.3.4 Effect of receipt

The SME is notified of the reachability to a mesh gate in the mesh BSS. The mesh STA received this primitive subsequently triggers MBSSGATEANNOUNCEMENT.request as described in 13.11.2.

6.3.81 Mesh link metric

6.3.81.1 Introduction

Subclause 6.3.81 describes the management procedures associated with mesh link metric reporting.

6.3.81.2 MLME-MESHLINKMETRICREAD.request

6.3.81.2.1 Function

This primitive requests to read a link metric value between the local MAC entity and a specific peer MAC entity.

6.3.81.2.2 Semantics of the service primitive

The primitive parameter is as follows:

```
MLME-MESHLINKMETRICREAD.request(
                              PeerMACAddress
                              )
```

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Any valid individual MAC address	Specifies the address of the peer MAC entity for which the link metric value is read

6.3.81.2.3 When generated

This primitive is generated by the SME to read the link metric value for the mesh link to the specified peer MAC entity.

6.3.81.2.4 Effect of receipt

On receipt of this primitive, the MLME reports the link metric value. The MLME subsequently issues an MLME-MESHLINKMETRICREAD.confirm that reflects the results of this request.

6.3.81.3 MLME-MESHLINKMETRICREAD.confirm

6.3.81.3.1 Function

This primitive reports the results of a link metric read request.

6.3.81.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME- MESHLINKMETRICREAD.confirm(
                              LinkMetricValue,
                              VendorSpecificInfo
                              )
```

Name	Type	Valid range	Description
LinkMetricValue	Mesh Link Metric Report element	As defined in 8.4.2.102	The link metric value for the mesh link to the specified peer MAC entity.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.81.3.3 When generated

This primitive is generated by the MLME as a result of an MLME-MESHLINKMETRICREAD.request primitive to request a link metric value.

6.3.81.3.4 Effect of receipt

The SME is notified of the results of the link metric read request.

6.3.81.4 MLME-MESHLINKMETRICREPORT.request

6.3.81.4.1 Function

This primitive requests that the MAC entity either transmit a link metric to or request a link metric from the specified peer MAC entity.

6.3.81.4.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MESHLINKMETRICREPORT.request(
                                  PeerMACAddress,
                                  LinkMetricRequestFlag,
                                  MeshLinkMetricReport,
                                  VendorSpecificInfo
                                  )
```

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Any valid individual MAC address	Specifies the address of the peer MAC entity to which the Mesh Link Metric Report is sent.
LinkMetricRequestFlag	Enumeration	REPORT_ONLY, REPORT_AND_REQUEST	Indicates whether the mesh STA requests a link metric report from the peer MAC entity.
MeshLinkMetricReport	Mesh Link Metric Report element	As defined in 8.4.2.102	A metric value computed for the corresponding link.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.81.4.3 When generated

This primitive is generated by the SME to request that a Mesh Link Metric Report frame be sent to a peer MAC entity in order to report a link metric value and to request a mesh link metric report from the peer MAC entity if LinkMetricRequestFlag is equal to REPORT_AND_REQUEST.

6.3.81.4.4 Effect of receipt

On receipt of this primitive, the MLME constructs a Mesh Link Metric Report frame.The Request subfield in the Flags field of the Mesh Link Metric Report element is set depending on the parameter given by the LinkMetricRequestFlag. If LinkMetricRequestFlag is equal to REPORT_ONLY, the Request subfield is set to 0. If LinkMetricRequestFlag is equal to REPORT_AND_REQUEST, the Request subfield is set to 1. This frame is then scheduled for transmission.

6.3.81.5 MLME-MESHLINKMETRICREPORT.indication

6.3.81.5.1 Function

This primitive indicates that a Mesh Link Metric Report frame has been received from a peer MAC entity. This Mesh Link Metric Request Report can be in response to an earlier MLME-MESHLINKMETRICREPORT.request primitive with LinkMetricRequestFlag equal to REPORT_AND_REQUEST.

6.3.81.5.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME- MESHLINKMETRICREPORT.indication(
                                      PeerMACAddress,
                                      LinkMetricRequestFlag,
                                      MeshLinkMetricReport,
                                      VendorSpecificInfo
                                      )
```

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Any valid individual MAC address	Specifies the address of the peer MAC entity from which the Mesh Link Metric Report frame was received.
LinkMetricRequestFlag	Enumeration	REPORT_ONLY, REPORT_AND_ REQUEST	Indicates whether the peer MAC entity requests a link metric report.
MeshLinkMetricReport	Mesh Link Metric Report element	As defined in 8.4.2.102	A metric value reported from the specified peer MAC entity.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.81.5.3 When generated

This primitive is generated by the MLME as a result of the receipt of a Mesh Link Metric Report frame from a specific peer MAC entity.

6.3.81.5.4 Effect of receipt

The SME is notified of the receipt of the link metric report from the specified peer MAC entity. When LinkMetricRequestFlag is equal to REPORT_AND_REQUEST, the mesh STA responds with a Mesh Link Metric Report frame.

6.3.82 HWMP mesh path selection

6.3.82.1 Introduction

The following primitives describe how a mesh STA establishes and maintains a mesh path to a specified peer MAC entity.

6.3.82.2 MLME-HWMPMESHPATHSELECTION.request

6.3.82.2.1 Function

This primitive requests that the MAC entity establish or maintain a mesh path to the specified peer MAC entity by transmitting an HWMP Mesh Path Selection frame to the specified peer MAC entity.

6.3.82.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-HWMPMESHPATHSELECTION.request(
                                PeerMACAddress,
                                RootAnnouncement,
                                PathRequest,
                                PathReply,
                                PathError,
                                VendorSpecificInfo
                                )
```

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Any valid individual or group MAC address	Specifies the address of the peer MAC entity to which the HWMP Mesh Path Selection frame is sent.
RootAnnouncement	RANN element	As defined in 8.4.2.114	A set of RANN elements generated by the mesh STA. Present only if the mesh STA is configured as a root mesh STA using the proactive RANN mechanism [dot11MeshHWMProotMode = rann (4)], and as described in 13.10.12.
PathRequest	PREQ element	As defined in 8.4.2.115	A set of PREQ elements generated by the mesh STA. Present as described in 13.10.9.
PathReply	PREP element	As defined in 8.4.2.116	A set of PREP elements generated by the mesh STA. Present as described in 13.10.10.
PathError	PERR element	As defined in 8.4.2.117	A set of PERR elements generated by the mesh STA. Present as described in 13.10.11.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.82.2.3 When generated

This primitive is generated by the SME to request that an HWMP Mesh Path Selection frame be sent to a specified peer entity.

6.3.82.2.4 Effect of receipt

On receipt of this primitive, the MLME constructs an HWMP Mesh Path Selection frame. This frame is then scheduled for transmission.

6.3.82.3 MLME-HWMPMESHPATHSELECTION.indication

6.3.82.3.1 Function

This primitive indicates that an HWMP Mesh Path Selection frame has been received from the specified peer MAC entity.

6.3.82.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-HWMPMESHPATHSELECTION.indication(
                                    PeerMACAddress,
                                    RootAnnouncement,
                                    PathRequest,
                                    PathReply,
                                    PathError,
                                    VendorSpecificInfo
                                    )
```

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Any valid individual MAC address	Specifies the address of the peer MAC entity from which the HWMP Mesh Path Selection frame was received.
RootAnnouncement	RANN element	As defined in 8.4.2.114	A set of RANN elements contained in the received frame. Present only when such an element was present in the received frame.
PathRequest	PREQ element	As defined in 8.4.2.115	A set of PREQ elements contained in the received frame. Present only when such an element was present in the received frame.
PathReply	PREP element	As defined in 8.4.2.116	A set of PREP elements contained in the received frame. Present only when such an element was present in the received frame.
PathError	PERR element	As defined in 8.4.2.117	A set of PERR elements contained in the received frame. Present only when such an element was present in the received frame.
VendorSpecificInfo	A set of elements	As defined in 8.4.2.28	Zero or more elements.

6.3.82.3.3 When generated

This primitive is generated by the MLME as a result of the receipt of an HWMP Mesh Path Selection frame from a specific peer MAC entity.

6.3.82.3.4 Effect of receipt

The SME is notified of the results of the receipt of the HWMP Mesh Path Selection from the specified peer MAC entity. The mesh STA received this primitive subsequently activates path selection procedures described in 13.10.

6.4 MAC state generic convergence function (MSGCF)

6.4.1 Overview of the convergence function

The MSGCF and its interaction with other management entities is defined in 6.4. The MSGCF correlates information exchanged between the MAC management entities regarding the state of an IEEE 802.11 interface and converges this information into events and status for consumption by higher layer protocols. Non-AP STAs when dot11MSGCFActivated is set to true shall support the MSGCF procedures in this clause; APs do not support the MSGCF.

This clause defines interactions between the MSGCF and MLME and PLME through the MLME_SAP and PLME_SAP respectively, as well as with the SME via the MSGCF-SME_SAP. The detailed manner in which the SAPs are implemented is not specified within this standard.

The MSGCF operates at the level of an IEEE 802.11 ESS, and generates events based on the state of the link between a non-AP STA and an ESS. A non-AP STA that transitions between two APs in the same ESS can operate transparently to the LLC sublayer, and does not change state in the state machine defined within this clause.

6.4.2 Overview of convergence function state machine

The convergence function maintains information on the state of the ESS, using the state machine shown in Figure 6-26. Because Figure 6-26 is defined in terms of ESS connectivity, it is not affected by changes in association provided that the transition was an intra-ESS transition.

6.4.3 Convergence function state list

6.4.3.1 ESS_CONNECTED

In the ESS_CONNECTED state, a non-AP STA has completed all layer 2 setup activities and is able to send Class 3 frames to peer LLC entities. A non-AP STA remains in this state as long as it is possible to send Class 3 frames through any AP within an ESS. A non-AP STA does not leave this state upon successful intra-ESS transitions.

6.4.3.2 ESS_DISCONNECTED

In the ESS_DISCONNECTED state, a non-AP STA is unable to send Class 3 frames to peer LLC entities. Higher layer network protocols are unavailable. In this state, a non-AP STA may use GAS to perform network discovery and selection.

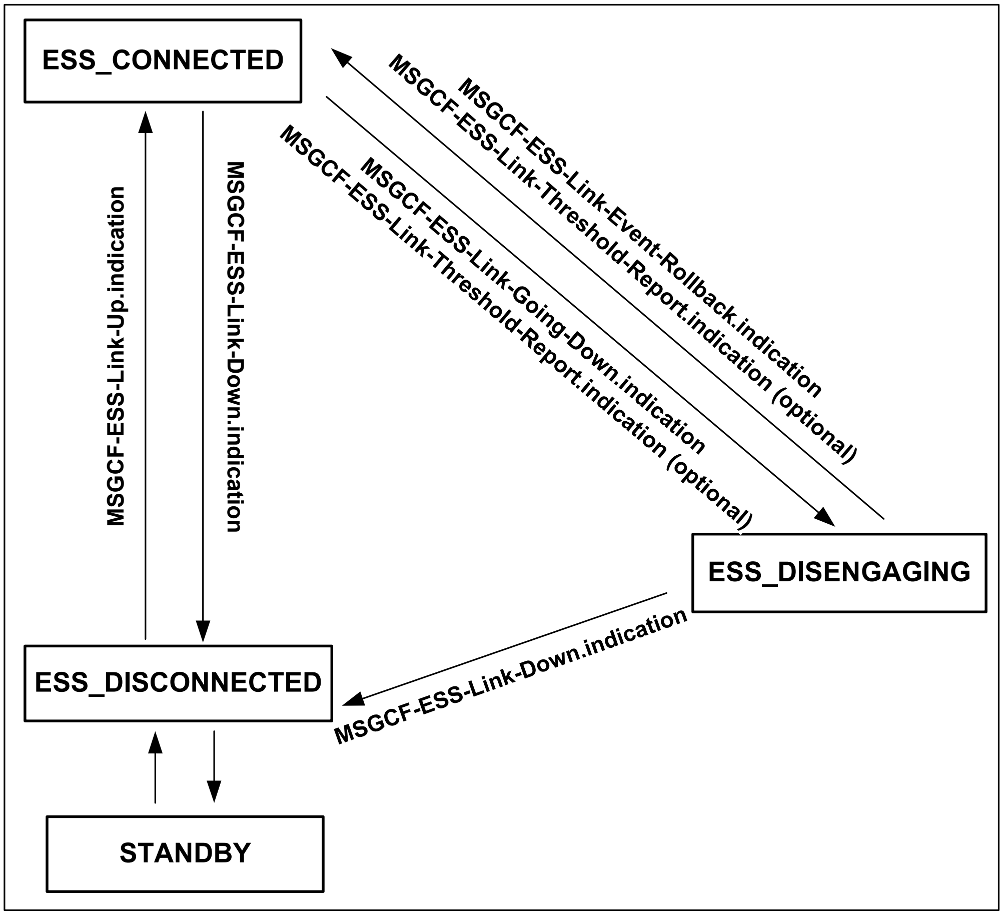

ESS_CONNECTED
MSGCF-ESS-Link-Up.indication
MSGCF-ESS-Link-Down.indication
MSGCF-ESS-Link-Going-Down.indication
MSGCF-ESS-Link-Threshold-Report.indication (optional)
MSGCF-ESS-Link-Event-Rollback.indication
MSGCF-ESS-Link-Threshold-Report.indication (optional)
ESS_DISENGAGING
MSGCF-ESS-Link-Down.indication
ESS_DISCONNECTED
STANDBY

Figure 6-26—MSGCF state machine

6.4.3.3 ESS_DISENGAGING

In the ESS_DISENGAGING state, the non-AP STA's SME anticipates that links to all APs within the ESS will be lost in a defined time interval, but the non-AP STA is still able to send Class 3 frames to peer LLC entities. The predictive failure of the link may be due to explicit disassociation by the peer, the imminent invalidation of cryptographic keys because of usage limits (such as sequence counter exhaustion), or predictive signal strength algorithms. In this state, it is recommended that a non-AP STA also initiate a search to find a new ESS.

6.4.3.4 STANDBY

In the STANDBY state, the non-AP STA is powered down and unable to communicate with any other IEEE 802.11 STAs.

6.4.4 Convergence function state transitions

6.4.4.1 Transitions to ESS_CONNECTED

6.4.4.1.1 From ESS_DISCONNECTED

To make this transition, a non-AP STA will have completed the network selection process and the relevant procedures to attach to the ESS, including IEEE 802.11 authentication, IEEE 802.11 association, and, if required, IEEE 802.11 RSN procedures. When this transition is completed, the MSGCF sends an MSGCF-ESS-Link-Up.indication primitive to higher layers.

6.4.4.1.2 From ESS_DISENGAGING

To make this transition, the SME cancels a previous event that predicted an ESS link failure. This may be due to network parameters indicating renewed link strength or a successful renewal of an expiring RSN SA. When this transition is complete, the MSGCF sends an MSGCF-ESS-Link-Event-Rollback.indication event to indicate that a prior link failure predictive event is no longer valid. If the transition was due to network parameters crossing a threshold, the MSGCF also issues an MSGCF-ESS-Link-Threshold-Report.indication to higher layers.

6.4.4.2 Transitions to ESS_ DISCONNECTED

6.4.4.2.1 From ESS_CONNECTED

This transition indicates that administrative action was taken to shut down the link, a sudden loss of signal strength or that RSN keys expired and could not be renewed. At the conclusion of this transition, the MSGCF issues an MSGCF-ESS-Link-Down.indication event to higher layer protocols.

6.4.4.2.2 From ESS_DISENGAGING

This transition indicates that the predictive link failure event has occurred. At the conclusion of this transition, the MSGCF issues an MSGCF-ESS-Link-Down.indication event to higher layer protocols.

6.4.4.2.3 From STANDBY

This transition occurs when the non-AP STA is powered on and initialized. No event is issued by the MSGCF.

6.4.4.3 Transitions to ESS_DISENGAGING

6.4.4.3.1 From ESS_CONNECTED

When the parameters as defined in Table 6-7 change or imminent action is taken to bring down the link, the SME may predict an imminent link failure and initiate a transition. Upon completion of this transition, the MSGCF issues an MSGCF-ESS-Link-Going-Down event. If the cause of the transition was the degradation of network parameters beyond the thresholds stored in the MIB, an MSGCF-ESS-Link-Threshold-Report.indication is also issued to higher layers.

6.4.4.4 Transitions to STANDBY

6.4.4.4.1 From ESS_DISCONNECTED

When the non-AP STA has disconnected from an ESS, it may be administratively powered off to extend battery life. No events are issued by the MSGCF upon completion of this transition.

6.4.5 Convergence function informational events

Informational events may occur in any state. When they occur, the SME updates the convergence function MIB with new parameters. Informational events do not cause state changes in Figure 6-26. Informational events are generated when new potential ESS links are discovered, when the network parameter thresholds are set or read, and when higher layer protocols issue commands to the non-AP STA through the MSGCF-ESS-Link-Command.request primitive.

6.4.6 MAC state generic convergence SAP

The MAC state generic convergence SAP is the interface between the convergence function and higher layer protocols. It presents a standardized interface for higher layer protocols to access the state of the MAC, whether that state information is available in the MLME, PLME, or SME.

Some events on the MAC state generic convergence SAP require event identifiers for use as a dialog token in event sequencing and rollback. The EventID is an unsigned integer that is initialized to one when the non-AP STA leaves the STANDBY state.

6.4.7 ESS status reporting

6.4.7.1 MSGCF-ESS-Link-Up

6.4.7.1.1 Function

This event is triggered when a new ESS has been made available for sending frames.

6.4.7.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MSGCF-ESS-Link-Up.indication(
                              NonAPSTAMacAddress,
                              ESSIdentifier
                              )
```

Name	Type	Valid range	Description
NonAPSTAMacAddress	MAC Address	Any valid individual MAC Address	The MAC address of the non-AP STA that is reporting that an IEEE 802.11 ESS has become available.
ESSIdentifier	String	N/A	An identifier for the network, composed of the string value of the SSID element used to identify the network, concatenated with the value of the HESSID if it is in use. The HESSID is encoded in upper-case ASCII characters with the octet values separated by dash characters, as described in IETF RFC 3580 [B35].

6.4.7.1.3 When generated

This primitive is generated when the ESS link to a network of APs is available to exchange data frames. The generation of this primitive may vary depending on the contents of dot11WEPDefaultKeysTable and dot11WEPKeyMappingsTable and the setting of dot11RSNAOptionImplemented.

If there are no entries in the dot11WEPDefaultKeysTable, no entry for the current AP in dot11WEPKeyMappingsTable, and dot11RSNAOptionImplemented is false, then the network does not use encryption. This event is generated upon receipt of an MLME-ASSOCIATE.confirm message with a result code of success.

If there are entries in the dot11WEPDefaultKeysTable, or an entry for the current AP in dot11WEPKeyMappingsTable, or dot11RSNAOptionImplemented is true, then the network requires the use of encryption on the link. Before declaring that the link is ready to exchange data frames, the convergence function receives an MLME-ASSOCIATE.confirm primitive with a result code of success and the SME emits an MLME-SETKEYS.request primitive. The latter primitive is used to determine that a WEP key is available, or that the RSN 4-Way Handshake has completed.

This event is not triggered by MLME-REASSOCIATE.confirm messages because MLME-REASSOCIATE.confirm messages are defined as transitions within the same ESS.

The MLME-ASSOCIATE.confirm primitive may be issued upon AP transitions. It is the objective of the MSGCF to generate this event only upon the initial connection to an IEEE 802.11 network, when the MSGCF state machine moves into the ESS_CONNECTED state.

6.4.7.1.4 Effect of receipt

This event is made available to higher layer protocols by the convergence function. Actions taken by higher layers are outside of scope of this standard, but may include router discovery, IP configuration, and other higher layer protocol operations.

6.4.7.2 MSGCF-ESS-Link-Down.indication

6.4.7.2.1 Function

This event is triggered to indicate that an IEEE 802.11 ESS is no longer available for sending frames.

6.4.7.2.2 Semantics of the service primitive

The event’s parameters are as follows:

```
   MSGCF-ESS-Link-Down.indication (
                                    NonAPSTAMacAddress,
                                    ESSIdentifier,
                                    ReasonCode
                                    )
```

Name	Type	Valid range	Description
NonAPSTAMac Address	MAC Address	Any valid individual MAC Address	The MAC address of the non-AP STA that is reporting that an IEEE 802.11 ESS is no longer available.
ESSIdentifier	String	N/A	An identifier for the network, composed of the string value of the SSID used to identify the network, concatenated with the value of the HESSID if it is in use.
ReasonCode	Enumerated	EXPLICT_DISCONNECT, KEY_EXPIRATION, LOW_POWER, VENDOR_SPECIFIC	Reason code, drawn from Table 6-2.

Table 6-2—Reason codes for network down

Name	Description
EXPLICIT_DISCONNECT	An explicit disconnection operation (Disassociation or Deauthentication) was initiated by the non-AP STA or the non-AP STA's current serving AP and the non-AP STA was unable to Reassociate with an alternate AP in the same ESS.
KEY_EXPIRATION	Keys used by an RSN SA have expired due to time or traffic limitations, or TKIP countermeasures have invalidated the key hierarchy.
LOW_POWER	If the SME reports that the IEEE 802.11 interface was shut down to conserve power, that event may be reported to higher level protocols.
VENDOR_SPECIFIC	Vendor-specific usage.

6.4.7.2.3 When generated

This event is generated when the SME declares that connectivity to an ESS is lost. It may be generated in the case of an explicit disconnection from the link peer, received as an MLME-DEAUTHENTICATE.indication or an MLME-DISASSOCIATE.indication primitive message. The SME should wait for a period of dot11ESSDisconnectFilterInterval before declaring connectivity lost to confirm that a non-AP STA is unable to reassociate with any alternate AP within the ESS.

6.4.7.2.4 Effect of receipt

This event is made available to higher layer protocols by the convergence function. Actions taken by those higher layers are outside the scope of this standard, but may include removing entries from routing and forwarding tables, and attempting to initiate handover of open application connections to network interfaces that are still active.

6.4.7.3 MSGCF-ESS-Link-Going-Down

6.4.7.3.1 Function

This event is triggered to indicate the expectation that IEEE 802.11 ESS will no longer be available for sending frames in the near future.

6.4.7.3.2 Semantics of the service primitive

The event parameters are as follows:

```
MSGCF-ESS-Link-Going-Down.indication(
                                        NonAPSTAMacAddress,
                                        ESSIdentifier,
                                        EventID,
                                        TimeInterval,
                                        ReasonCode
                                        )
```

Name	Type	Valid range	Description
NonAPSTAMac Address	MacAddress	Any valid individual MAC Address	The MAC address of the non-AP STA that is reporting that an IEEE 802.11 ESS is expected to go down.
ESSIdentifier	String	N/A	An identifier for the network, composed of the string value of the SSID element used to identify the network, concatenated with the value of the HESSID if it is in use.
EventID	Integer	N/A	An integer used to identify the event that is used in the case of event rollback.
TimeInterval	Integer	N/A	Time Interval, in time units, in which the link is expected to go down. Connectivity is expected to be available at least for time specified by TimeInterval.
Reason Code	Enumerated	EXPLICT_DISCONNECT, KEY_EXPIRATION, LOW_POWER, VENDOR_ SPECIFIC	Indicates the reason the link is expected to go down, drawn from Table 6-3.

Table 6-3—Reason codes for ESS link down

Name	Description
EXPLICIT_DISCONNECT	An explicit disconnection operation (Disassociation or Deauthentication) was initiated by the non-AP STA or the non-AP STA's current serving AP.
KEY_EXPIRATION	Keys used by an RSN SA have expired due to time or traffic limitations, or TKIP countermeasures have invalidated the key hierarchy.
LOW_POWER	If the SME reports that the IEEE 802.11 interface is going to be shut down to conserve power, that event may be reported to higher level protocols.
VENDOR_SPECIFIC	Vendor-specific usage.

6.4.7.3.3 When generated

This notification is generated by the MSGCF when the IEEE 802.11 ESS link is currently established and is expected to go down within the specified time interval. The network may be expected to go down because of an event whose timing is well understood, such as an explicit disconnection event observed on the MLME_SAP. It may also be expected as the result of a predictive algorithm that monitors link quality. The details of such a predictive algorithm used are beyond the scope of this standard.

The convergence function should attempt to deliver this event at least dot11ESSLinkDownTimeInterval time units before the link is predicted to go down. Different higher layer network protocols may require different levels of advance notice, and may configure the dot11ESSLinkDownTimeInterval attribute accordingly.

Not all thresholds in the dot11MACStateParameterTable are supported by every PHY. In the case when a threshold parameter is not supported (e.g., RSSI in Clause 15), it is not applied.

6.4.7.3.4 Effect of receipt

This event is made available to higher layer protocols by the convergence function. Actions taken by those higher layers are outside the scope of this standard, but may include beginning preparations for handover.

6.4.7.4 MSGCF-ESS-Link-Event-Rollback.indication

6.4.7.4.1 Function

This event is used to indicate that specific previous reports or events are no longer valid and should be disregarded.

6.4.7.4.2 Semantics of the service primitive

The event parameters are as follows:

```
MSGCF-ESS-Link-Event-Rollback.indication(
                                         NonAPSTAMacAddress,
                                         ESSIdentifier,
                                         EventID
                                         )
```

Name	Type	Valid range	Description
NonAPSTAMac Address	MacAddress	Any valid individual MAC Address	The MAC address of the non-AP STA that is reporting that a previous event relating to an IEEE 802.11 ESS is no longer valid.
ESSIdentifier	String	N/A	An identifier for the network, composed of the string value of the SSID element used to identify the network, concatenated with the value of the HESSID if it is in use.
EventID	Integer	N/A	An integer used to identify the event that is used in the case of event rollback.

6.4.7.4.3 When generated

This event is generated when a previous predictive event is no longer valid within its expiration time.

MSGCF-ESS-Link-Event-Rollback.indication is used in conjunction with MSGCF-ESS-Link-Going-Down. MSGCF-ESS-Link-Event-Rollback.indication events are issued when the prediction of link failure is no longer valid. Algorithms used to determine that link failure predictions are beyond the scope of this standard.

6.4.7.4.4 Effect of receipt

This event is made available to higher layer protocols by the convergence function to cancel any actions begun by the previous event. Actions taken by those higher layers are outside the scope of this standard, but may include cancelling any handover procedures started by the MSGCF-ESS-Link-Going-Down event.

6.4.7.5 MSGCF-ESS-Link-Detected.indication

6.4.7.5.1 Function

This event reports on the presence of a new IEEE 802.11 ESS.

6.4.7.5.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MSGCF-ESS-Link-Detected.indication(
                                   NonAPSTAMacAddress,
                                   ESSIdentifier,
                                   ESSDescription
                                   )
```

Name	Type	Valid range	Description
NonAPSTAMacAddress	MacAddress	Any valid individual MAC Address	The MAC address of the non-AP STA that is reporting the new network.
ESSIdentifier	String	N/A	An identifier for the network, composed of the string value of the SSID used to identify the network, concatenated with the value of the HESSID if it is in use.
ESSDescription	As defined in Table 6-4	N/A	A set of information about the ESS.

Table 6-4—ESS description

Name	Syntax	Description
SSID	String	The SSID used by the ESS.
InformationServiceSupport	As described in Table 6-5	A set of values indicating the type of information services supported on this network.
TriggerSupport	As described in Table 6-5	A set of values indicating the support for the types of triggers that can be used to propose that the station take action.
RSN	As defined in 8.4.2.27	The RSN configuration of the ESS.
Interworking	As defined in 8.4.2.94	Interworking configuration of the ESS.

Table 6-5—Trigger support values

Name	Description
MIH_CS_ES_Support	This network supports the IEEE 802.21 MIH Command Service and Event Service.
Vendor_Specific_Trigger_Support	This network supports a vendor-specific trigger service.

6.4.7.5.3 When generated

Support for MIH is indicated by the presence or absence of the relevant Advertisement Protocol IDs in the Advertisement Protocol element.To maintain the list of detected networks, the SME issues recurring MLME-SCAN.request primitives to the MLME. The SME may schedule these requests to avoid interruption of user traffic. Responses to these requests, received in the MLME-SCAN.confirm primitives, contain a list of detected networks. Each network is stored in the MIB in the dot11MACStateESSLinkDetectedTable. This table holds a list of networks, organized by Network Identifier. Each entry in the table contains a list of

BSSIDs within the network, as well as indications of support for MIH. Support for MIH is indicated by the presence or absence of the relevant Advertisement Protocol IDs in the Advertisement Protocol element. Each entry in the table is held for at least dot11ESSLinkDetectionHoldInterval time units. When a non-AP STA has not observed an ESS for longer than dot11ESSLinkDetectionHoldInterval, it may be removed from the table.

This event is generated when a new entry is made into the dot11MACStateESSLinkDetectedTable. Modifications to existing entries in the list, such as an update to the BSSID list, do not trigger this event.

6.4.7.5.4 Effect of receipt

This event is made available to higher layer protocols by the convergence function. Actions taken by those higher layers are outside the scope of this standard.

6.4.7.6 MSGCF-ESS-Link-Scan.request

6.4.7.6.1 Function

This function is used by higher layer protocols to request that the SME perform a scan operation for available ESSs.

6.4.7.6.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MSGCF-ESS-Link-Scan.request(
                                    SSID,
                                    HESSID,
                                    AccessNetworkType
                                    )
```

Name	Type	Valid range	Description
SSID	Octet string	0–32 octets	Specific or wildcard.
HESSID	As defined in 8.4.2.94	As defined in 8.4.2.94	The HESSID to search for. It can be set to all 1s for use as a wildcard to match all available HESSID values.
AccessNetworkType	As defined in 8.4.2.94	As defined in 8.4.2.94	This may be a specific value to match one type of networks, or all 1s to match all access network types.

6.4.7.6.3 When generated

This request is generated when higher protocol layers request a list of available ESSs.

6.4.7.6.4 Effect of receipt

The SME generates a corresponding MLME-SCAN.request primitive to find available networks.

6.4.7.7 MSGCF-ESS-Link-Scan.confirm

6.4.7.7.1 Function

This function reports information on available ESSs to higher protocol layers.

6.4.7.7.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MSGCF-ESS-Link-Scan.confirm(
                              NonAPSTAMacAddress,
                              ESSIdentifiers,
                              ESSDescriptions
                              )
```

Name	Type	Valid range	Description
NonAPSTAMacAddress	MacAddress	Any valid individual MAC Address	The MAC address of the non-AP STA that is reporting the new network.
ESSIdentifiers	Set of Strings	N/A	An identifier for the network composed of the string value of the SSID used to identify the network, concatenated with the value of the HESSID if it is in use.
ESSDescriptions	Set of ESSDescriptions, as defined in Table 6-4	N/A	A set of information about each discovered ESS.

6.4.7.7.3 When generated

This primitive is generated when scan results are available for reporting to higher protocol layers, in response to an MSGCF-ESS-Link-Scan.request primitive.

6.4.7.7.4 Effect of receipt

This event is made available to higher layer protocols by the convergence function. Actions taken by those higher layers are outside the scope of this standard.

6.4.8 Network configuration

6.4.8.1 MSGCF-ESS-Link-Capability.request

6.4.8.1.1 Function

This primitive requests a list of the capabilities supported by a network.

6.4.8.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MSGCF-ESS-Link-Capability.request(
                              NonAPSTAMacAddress,
                              ESSIdentifier
                              )
```

Name	Type	Valid range	Description
NonAPSTAMac Address	MacAddress	Any valid individual MAC Address	The MAC address of the non-AP STA that is reporting the new network.
ESSIdentifier	String	N/A	An identifier for the network, composed of the string value of the SSID element used to identify the network, concatenated with the value of the HESSID if it is in use.

6.4.8.1.3 When generated

This primitive is issued to service higher layer protocols by reporting on the capabilities of a particular network.

6.4.8.1.4 Effect of receipt

The convergence function retrieves the capabilities and reports them via the MSGCF-ESS-Link-Capability.confirm primitive.

6.4.8.2 MSGCF-ESS-Link-Capability.confirm

6.4.8.2.1 Function

This primitive reports the convergence function capabilities of the network to higher layer protocols.

6.4.8.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MSGCF-ESS-Link-Capability.confirm(
                                  NonAPSTAMacAddress,
                                  ESSIdentifier,
                                  EssLinkParameterSet,
                                  ReasonCode
                                  )
```

Name	Type	Valid range	Description
NonAPSTAMac Address	MacAddress	Any valid individual MAC Address	The MAC address of the non-AP STA that is reporting the new network.
ESSIdentifier	String	N/A	An identifier for the network, composed of the string value of the SSID element used to identify the network, concatenated with the value of the HESSID if it is in use.
EventCapability Set	As defined in Table 6-6	N/A	list of supported events.
ReasonCode	Enumerated	SUCCESS, UNKNOWN_NETWORK, UNKNOWN_CAPABILITIES	An error code, if applicable.

Table 6-6—Event Capability Set

Name	Type	Valid range	Description
NonAPSTAMacAddress	MacAddress	Any valid individual MAC Address	The MAC address of the non-AP STA that is reporting the new network.
ESS-Link-Up	Boolean	true, false	Indicates whether the MSGCF-ESS-Link-Up.indication event as defined in 6.4.7.1 is supported.
ESS-Link-Down	Boolean	true, false	Indicates whether the MSGCF-ESS-Link-Down.indication event as defined in 6.4.7.2 is supported.
ESS-Link-Going-Down	Boolean	true, false	Indicates whether the MSGCF-ESS-Link-Going-Down event as defined in 6.4.7.3 is supported.
ESS-Link-Event-Rollback	Boolean	true, false	Indicates whether the MSGCF-ESS-Link-Event-Rollback.indication event as defined in 6.4.7.4 is supported.
ESS-Link-Detected	Boolean	true, false	Indicates whether the MSGCF-ESS-Link-Detected.indication event as defined in 6.4.7.5 is supported.
ESS-Link-Threshold-Report	Boolean	true, false	Indicates whether the MSGCF-ESS-Link-Threshold-Report.indication event as defined in 6.4.9.1 is supported.
ESS-Link-Command	Boolean	true, false	Indicates whether the MSGCF-ESS-Link-Command.request primitive as defined in 6.4.10.1 is supported.

6.4.8.2.3 When generated

This primitive is generated in response to the MSGCF-ESS-Link-Capability.request primitive to report whether specific events are supported.

6.4.8.2.4 Effect of receipt

This event is made available to higher layer protocols by the convergence function.

6.4.8.3 MSGCF-Set-ESS-Link-Parameters.request

6.4.8.3.1 Function

This primitive sets thresholds for reporting of network events.

6.4.8.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MSGCF-Set-ESS-Link-Parameters.request(
                                    NonAPSTAMacAddress,
                                    ESSIdentifier,
                                    EssLinkParameterSet
                                    )
```

Name	Type	Valid range	Description
NonAPSTAMacAddress	MacAddress	Any valid individual MAC Address	The MAC address of the non-AP STA that is reporting the new network.
ESSIdentifier	String	N/A	An identifier for the network, composed of the string value of the SSID element used to identify the network, concatenated with the value of the HESSID if it is in use.
ESSLinkParameterSet	As defined in Table 6-7	N/A	The EssLinkParameterSet is used to configure when event reports are sent to higher protocol layers.

The ESSLinkParameterSet parameter is defined in Table 6-7. It may include any or all of the elements in Table 6-7.

Table 6-7—ESS Link Parameter Set

Name	Type	Valid range	Description
PeakOperationalRate	Integer	As defined in 8.4.2.3	The integer representing the desired peak modulation data rate used for data frame transmission.
MinimumOperationalRate	Integer	As defined in 8.4.2.3	The integer encoding of the desired minimum modulation data rate used in data frame transmission.
NetworkDowntimeInterval	Integer	0 – 65 535	Desired advance warning time interval, in TUs, for MSGCF-ESS-Link-Going-Down events.
DataFrameRSSI	Integer	–100 to 40	The received signal strength in dBm of received Data frames from the network. This may be time-averaged over recent history by a vendor-specific smoothing function.
BeaconRSSI	Integer	–100 to 40	The received signal strength in dBm of Beacon frames received on the channel. This may be time-averaged over recent history by a vendor-specific smoothing function.
BeaconSNR	Integer	0–100	The signal to noise ratio of the received data frames, in dB. This may be time-averaged over recent history by a vendor-specific smoothing function.
DataFrameSNR	Integer	0–100	The signal to noise ratio of the received Beacon frames, in dB. This may be time-averaged over recent history by a vendor-specific smoothing function.
DataThroughput	Integer	0 – 65 535	The data throughput in megabits per second, rounded to the nearest megabit. This may be time-averaged over recent history by a vendor-specific smoothing function.
MissedBeaconRate	Real	N/A	The rate at which beacons have not been received in missed beacons per second. This may be time-averaged over recent history by a vendor-specific smoothing function.

Table 6-7—ESS Link Parameter Set *(continued)*

Name	Type	Valid range	Description
FrameErrorRate	Real	N/A	The frame error rate of the network in errors per second. This may be time-averaged over recent history by a vendor-specific smoothing function.
VendorSpecific	Vendor Specific	As defined by 8.4.2.28	Additional vendor-specific parameters may be included in this event.

6.4.8.3.3 When generated

This event is generated when higher protocol layers wish to set the performance parameters for a network. Higher protocol layers are responsible for ensuring that the set of configured network parameters is consistent with all subscribers to those higher layer protocols.

6.4.8.3.4 Effect of receipt

Parameters supplied in the event are stored in the MIB, either in the dot11MACStateConfigTable or the dot11MACStateParameterTable.

6.4.8.4 MSGCF-Set-ESS-Link-Parameters.confirm

6.4.8.4.1 Function

This primitive indicates whether network parameters were accepted.

6.4.8.4.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MSGCF-Set-ESS-Link-Parameters.confirm(
                                      NonAPStaMacAddress,
                                      ESSIdentifier,
                                      EssLinkParameterSet,
                                      )
```

Name	Type	Valid range	Description
NonAPSTAMac Address	MacAddress	Any valid individual MAC Address	The MAC address of the non-AP STA that is reporting the new network.
ESSIdentifier	String	N/A	An identifier for the network, composed of the string value of the SSID element used to identify the network, concatenated with the value of the HESSID if it is in use.
EssLinkParamete rSet	As defined in Table 6-7	N/A	The EssLinkParameterSet is used to configure when event reports are sent to higher protocol layers.

6.4.8.4.3 When generated

This primitive is generated in response to the MSGCF-Set-ESS-Link-Parameters.request primitive and is used to indicate whether the parameter set was accepted.

6.4.8.4.4 Effect of receipt

The SME is notified of the new parameter set.

6.4.8.5 MSGCF-Get-ESS-Link-Parameters.request

6.4.8.5.1 Function

This primitive retrieves the current network parameters for a specific network.

6.4.8.5.2 Semantics of the service primitive

The primitive parameter is as follows:

```
MSGCF-Get-ESS-Link-Parameters.request(
                                      ESSIdentifier
                                      )
```

Name	Type	Valid range	Description
ESSIdentifier	String	N/A	An identifier for the network, composed of the string value of the SSID element used to identify the network, concatenated with the value of the HESSID if it is in use.

6.4.8.5.3 When generated

This primitive is used by higher layers to retrieve the currently stored parameters for a network.

6.4.8.5.4 Effect of receipt

The SME retrieves the network parameters and makes them available through the MSGCF-Get-ESS-Link-Parameters.confirm primitive.

6.4.8.6 MSGCF-Get-ESS-Link-Parameters.confirm

6.4.8.6.1 Function

This primitive reports the current network parameters.

6.4.8.6.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MSGCF-Get-ESS-Link-Parameters.confirm(
                                      ESSIdentifier,
                                      EssLinkParameterSet,
                                      )
```

Name	Type	Valid range	Description
ESSIdentifier	String	N/A	An identifier for the network, composed of the string value of the SSID element used to identify the network, concatenated with the value of the HESSID if it is in use.
EssLinkParameterSet	As defined 6.4.8.3	N/A	The EssLinkParameterSet is used to configure when event reports are sent to higher protocol layers.

6.4.8.6.3 When generated

This primitive is generated by the MSGCF as a result of the MSGCF-Get-ESS-Link-Parameters.request primitive.

6.4.8.6.4 Effect of receipt

The higher layer protocols are notified of the current network parameters.

6.4.9 Network events

6.4.9.1 MSGCF-ESS-Link-Threshold-Report.indication

6.4.9.1.1 Function

This event reports that the layer 2 network performance has crossed a threshold set by the operations described in Table 6-5.

6.4.9.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```
   MSGCF-ESS-Link-Threshold-Report.indication(
                                   NonAPSTAMacAddress,
                                   ESSIdentifier,
                                   EssLinkParameterSet,
                                   ThresholdCrossingDirectionSet
                                   )
```

Name	Type	Valid range	Description
NonAPSTAMacAddress	MacAddress	Any valid individual MAC Address	The MAC address of the non-AP STA that is reporting the threshold crossing.
ESSIdentifier	String	N/A	An identifier for the network, composed of the string value of the SSID element used to identify the network, concatenated with the value of the HESSID if it is in use.
EssLinkParameterSet	As defined in Table 6-7	N/A	A list of EssLinkParameterSets and their current values that have crossed preset thresholds for alerts.
ThresholdCrossingDirectionSet	Set of ThresholdCrossing Directions, one for each value in the EssLinkParameterSet	UPWARD, DOWNWARD	Whether the parameter has crossed the threshold while rising or falling.

6.4.9.1.3 When generated

The convergence function is responsible for monitoring network performance. If the monitored parameters cross the configured threshold, this event is generated to inform higher layer protocols.

6.4.9.1.4 Effect of receipt

This event is made available to higher layer protocols by the convergence function. Actions taken by those higher layers are outside the scope of this standard, but may include preparations for handover or assessing whether handover should be imminent.

6.4.10 Network command interface

6.4.10.1 MSGCF-ESS-Link-Command.request

6.4.10.1.1 Function

This primitive requests that a STA take action for a network.

6.4.10.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MSGCF-ESS-Link-Command.request(
                                    NonAPSTAMacAddress,
                                    ESSIdentifier,
                                    CommandType
                                    )
```

Name	Type	Valid range	Description
NonAPSTAMacAddress	MacAddress	Any valid individual MAC Address	The MAC address of the non-AP STA that is reporting the threshold crossing.
ESSIdentifier	String	N/A	An identifier for the network, composed of the string value of the SSID element used to identify the network, concatenated with the value of the HESSID if it is in use.
CommandType	Enumerated	DISCONNECT, LOW_POWER, POWER_UP, POWER_DOWN, SCAN	Type of command to perform on the link as described in the following subclauses.

6.4.10.1.3 When generated

This primitive is generated by a higher layer protocol.

6.4.10.1.4 Effect of receipt

The convergence function issues commands to the SME to implement the requested action on behalf of higher layers.

When the DISCONNECT command type is specified, the higher layer is requesting that the STA disconnect from its peer. When the SME on a non-AP STA receives this command, the SME issues an MLME-DEAUTHENTICATE.request to disconnect from the network, and the SME refrains from reconnecting to that network.

When the POWER_DOWN command type is specified, the SME powers down the non-AP STA. Before doing so, it may choose to notify the AP.

When the POWER_UP command type is specified, the SME starts the non-AP STA.

When the LOW_POWER command type is specified, the higher layer is requesting that the IEEE 802.11 interface be placed in a low power mode. This action is accomplished by issuing an MLME-POWERMGT.request primitive with the PowerManagementMode parameter set to POWER_SAVE.

When the SCAN command type is specified, the higher layer is requesting that the STA search for IEEE 802.11 networks. This action is accomplished by issuing an MLME-SCAN.request primitive. Detected networks are made available in the dot11MACStateESSLinkDetectedTable, as well as through the MSGCF-ESS-Link-Detected.indication event.

6.4.11 MAC state SME SAP—mobility management

6.4.11.1 MSSME-ESS-Link-Down-Predicted.indication

6.4.11.1.1 Function

This primitive indicates that the SME is predicting a link failure.

6.4.11.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```
   MSSME-ESS-Link-Going-Down.indication(
                                       NonAPSTAMacAddress,
                                       ESSIdentifier,
                                       TimeInterval,
                                       ReasonCode
                                       )
```

Name	Type	Valid range	Description
NonAPSTAMac Address	MacAddress	Any valid individual MAC Address	The MAC address of the non-AP STA that is reporting that an IEEE 802.11 ESS is expected to go down.
ESSIdentifier	String	N/A	An identifier for the network, composed of the string value of the SSID element used to identify the network, concatenated with the value of the HESSID if it is in use.
TimeInterval	Integer	N/A	Time Interval, in time units, in which the link is expected to go down. Connectivity is expected to be available at least for time specified by *TimeInterval*.
Reason Code	Enumerated	EXPLICIT_DISCONNECT, LINK_PARAMETER_DEGRADATION, KEY_EXPIRATION, LOW_POWER, QOS_UNAVAILABLE, VENDOR_SPECIFIC	Indicates the reason the link is expected to go down.

6.4.11.1.3 When generated

This notification is generated by the SME when the IEEE 802.11 network connection is currently established and is expected to go down. The details of the predictive algorithm used are beyond the scope of this standard. One method of implementing this function would be to generate this indication when link quality is fading and no better AP can be found.

6.4.11.1.4 Effect of receipt

This indication is received by the MSGCF and is used to generate the MSGCF-ESS-Link-Down.indication event due to link parameter degradation.

6.5 PLME SAP interface

6.5.1 General

The PHY management service interface consists of the generic PLMEGET and PLMESET primitives on PHY MIB attributes, as described previously, together with the PLME-RESET and PLME-CHARACTERISTICS primitives and the following specific primitives.

6.5.2 PLME-RESET.request

6.5.2.1 Function

This primitive is a request by the SME to reset the PHY. The PHY is always reset to the receive state to avoid accidental data transmission.

6.5.2.2 Semantics of the service primitive

This primitive has no parameters.

6.5.2.3 When generated

This primitive is generated at any time to reset the PHY.

6.5.2.4 Effect of receipt

Receipt of this primitive by the PHY causes the PHY entity to reset both the transmit and the receive state machines and places the PHY into the receive state.

6.5.3 PLME-CHARACTERISTICS.request

6.5.3.1 Function

This primitive is a request by the SME to provide the PHY operational characteristics.

6.5.3.2 Semantics of the service primitive

This primitive has no parameters.

6.5.3.3 When generated

This primitive is generated by the SME, at initialization time, to request the PHY entity to provide its operational characteristics.

6.5.3.4 Effect of receipt

The effect of receipt of this primitive by the PHY entity is the generation of a PLME-CHARACTERISTICS. confirm primitive that conveys its operational characteristics.

6.5.4 PLME-CHARACTERISTICS.confirm

6.5.4.1 Function

This primitive provides the PHY operational parameters.

6.5.4.2 Semantics of the service primitive

The primitive provides the following parameters:

```
PLME-CHARACTERISTICS.confirm(
                                        aSlotTime,
                                        aSIFSTime,
                                        aSignalExtension,
                                        aCCATime,
                                        aPHY-RX-START-Delay,
                                        aRxTxTurnaroundTime,
                                        aTxPLCPDelay,
                                        aRxPLCPDelay,
                                        aRxTxSwitchTime,
                                        aTxRampOnTime,
                                        aTxRampOffTime,
                                        aTxRFDelay,
                                        aRxRFDelay,
                                        aAirPropagationTime,
                                        aMACProcessingDelay,
                                        aPreambleLength,
                                        aRIFSTime,
                                        aSymbolLength,
                                        aSTFOneLength,
                                        aSTFTwoLength,
                                        aLTFOneLength,
                                        aLTFTwoLength,
                                        aPLCPHeaderLength,
                                        aPLCPSigTwoLength,
                                        aPLCPServiceLength,
                                        aPLCPConvolutionalTailLength,
                                        aMPDUDurationFactor,
                                        aMPDUMaxLength,
                                        aPSDUMaxLength,
                                        aPPDUMaxTime,
                                        aIUSTime,
                                        aDTT2UTTTime,
                                        aCWmin,
                                        aCWmax,
                                        aMaxCSIMatricesReportDelay
                                        aMaxTODError,
                                        aMaxTOAError,
                                        aTxPmdTxStartRFDelay,
                                        aTxPmdTxStartRMS
                                        )
```

The values assigned to the parameters is as specified in the PLME SAP interface specification contained within each PHY subclass of this standard. The parameter aMPDUDurationFactor is not used by all PHYs defined within this standard. The parameters aSignalExtension, aRIFSTime, aSymbolLength, aSTFOneLength, aSTFTwoLength, aLTFOneLength, aLTFTwoLength, aPLCPSigTwoLength, aPLCPServiceLength, aPLCPConvolutionalTailLength, aMPDUDurationFactor, aMPDUMaxLength, aPSDUMaxLength, aPPDUMaxTime, aIUSTime, aDTT2UTTTime, and aMaxCSIMatricesReportDelay are not used by all PHYs defined within this standard.

Name	Type	Description
aSlotTime	integer	The Slot Time (in microseconds) that the MAC uses for defining the PIFS and DIFS periods. See 9.3.7.
aSIFSTime	integer	The nominal time (in microseconds) that the MAC and PHY require in order to receive the last symbol of a frame at the air interface, process the frame, and respond with the first symbol on the air interface of the earliest possible response frame. See 9.3.7.
aSignalExtension	integer	Duration (in microseconds) of the signal extension (i.e., a period of no transmission) that is included at the end of certain PPDU formats; see 20.3.2 and 9.3.8.
aCCATime	integer	The maximum time (in microseconds) the CCA mechanism has available to assess the medium within every time slot to determine whether the medium is busy or idle.
aPHY-RX-START-Delay	integer	The delay, in microseconds, from a point in time specified by the PHY to the issuance of the PHY-RXSTART.indication primitive.
aRxTxTurnaroundTime	integer	The maximum time (in microseconds) that the PHY requires to change from receiving to transmitting the start of the first symbol. The following equation is used to derive the RxTxTurnaroundTime: aTxPLCPDelay + aRxTxSwitchTime + aTxRampOnTime + aTxRFDelay.
aTxPLCPDelay	integer	The nominal time (in microseconds) that the PLCP uses to deliver a symbol from the MAC interface to the transmit data path of the physical medium dependent (PMD).
aRxPLCPDelay	integer	The nominal time (in microseconds) that the PLCP uses to deliver the last bit of a received frame from the PMD receive path to the MAC.
aRxTxSwitchTime	integer	The nominal time (in microseconds) that the PMD takes to switch from Receive to Transmit.
aTxRampOnTime	integer	The maximum time (in microseconds) that the PMD takes to turn the Transmitter on.
aTxRampOffTime	integer	The nominal time (in microseconds) that the PMD takes to turn the Transmit Power Amplifier off.
aTxRFDelay	integer	The nominal time (in microseconds) between the issuance of a PMD_DATA.request primitive to the PMD and the start of the corresponding symbol at the air interface. The start of a symbol is defined to be 1/2 symbol period prior to the center of the symbol for FH, or 1/2 chip period prior to the center of the first chip of the symbol for DS, or 1/2 slot time prior to the center of the corresponding slot for infrared (IR).
aRxRFDelay	integer	The nominal time (in microseconds) between the end of a symbol at the air interface to the issuance of a PMD_DATA.indication primitive to the PLCP. The end of a symbol is defined to be 1/2 symbol period after the center of the symbol for FH, or 1/2 chip period after the center of the last chip of the symbol for DS, or 1/2 slot time after the center of the corresponding slot for IR.
aAirPropagationTime	integer	Twice the propagation time (in microseconds) for a signal to cross the maximum distance between the most distant allowable STAs that are slot synchronized.
aMACProcessingDelay	integer	The maximum time (in microseconds) available for the MAC to issue a PHY-TXSTART.request primitive pursuant to a PHY-RXEND.indication primitive (for response after SIFS) or PHY-CCA.indication(IDLE) primitive (for response at any slot boundary following a SIFS). This constraint on MAC performance is defined as a PHY-specific parameter because of its use, along with other PHY-specific time delays, in calculating the two PHY characteristics of primary concern to the MAC: aSlotTime and aSIFSTime. The relationship between aMACProcessingTime and the IFS and slot timing is described in 9.3.7 and illustrated in Figure 9-14.
aPreambleLength	integer	The current PHY's preamble length (in microseconds). If the actual value of the length of the modulated preamble is not an integral number of microseconds, the value is rounded up to the next higher value.

Name	Type	Description
aRIFSTime	integer	Value of the reduced interframe space (in microseconds), which is the time by which multiple transmissions from a single transmitter may be separated, when no SIFS-separated response transmission is expected. See 9.3.2.3.2
aSymbolLength	integer	The current PHY's Symbol length (in microseconds). If the actual value of the length is not an integral number of μs, the value is rounded up to the next higher value.
aSTFOneLength	integer	Length of the non-HT-STF (L-STF) for HT-mixed format, and the HT-greenfield STF (HT-GF-STF) for HT-greenfield format (in microseconds)
aSTFTwoLength	integer	Length of the HT-STF (in microseconds)
aLTFOneLength	integer	Length of the First HT-LTF (in microseconds)
aLTFTwoLength	integer	Length of the Additional HT-LTFs (in microseconds)
aPLCPHeaderLength	integer	The current PHY's PLCP header length (in microseconds), excluding aPLCPSigTwoLength if present. If the actual value of the length of the modulated header is not an integral number of microseconds, the value is rounded up to the next higher value.
aPLCPSigTwoLength	integer	Length of the HT SIGNAL field (HT-SIG) (in microseconds).
aPLCPServiceLength	integer	The length of the PLCP SERVICE field (in number of bits).
aPLCPConvolutionalTail Length	integer	The length of the sequence of convolutional code tail bits (in number of bits).
aMPDUDurationFactor	integer	The overhead added by the PHY to the MPDU as it is transmitted through the WM expressed as a scaling factor applied to the number of bits in the MPDU. The value of aMPDUDurationFactor is generated by the following equation: Truncate[((PPDUbits/PSDUbits)–1) × 10^9)]. The total time to transmit a PPDU over the air is generated by the following equation rounded up to the next integer μs: aPreambleLength + aPLCPHeaderLength + (((aMPDUDurationFactor × 8 × PSDUoctets) / 10^9) + (8 × PSDUoctets)) / data rate where data rate is in Mb/s. The total time (in μs) to the beginning of any octet in a PPDU from the first symbol of the preamble can be calculated using the duration factor in the following equation: Truncate[aPreambleLength + aPLCPHeaderLength + (((aMPDUDurationFactor × 8 × N) / 10^9) + (8 × N)) / data rate] + 1, where data rate is in Mb/s and where *N* counts the number of octets in the PPDU prior to the desired octet, but does not count the number of octets in the preamble PLCP header.
aMPDUMaxLength	integer	The maximum number of octets in an MPDU that can be conveyed by a PLCP protocol data unit (PPDU).
aPSDUMaxLength	integer	The maximum number of octets in a PSDU that can be conveyed by a PPDU.
aPPDUMaxTime	integer	The maximum duration of a PPDU in milliseconds.
aIUSTime	integer	The minimum time between the end of a PSMP-UTT and the start of the following PSMP-UTT in the same PSMP sequence.
aDTT2UTTTime	integer	The minimum time between the end of a PSMP-DTT and the start of the PSMP-UTT addressed to the same STA.
aCWmin	integer	The minimum size of the CW, in units of aSlotTime.
aCWmax	integer	The maximum size of the CW, in units of aSlotTime.
aMaxCSIMatriesReport Delay	integer	The maximum time (in milliseconds) between the reception of a frame containing a CSI Feedback Request or an NDP announcement and the transmission of the first CSI frame containing channel state information measured from the received Sounding Complete frame. See 9.29.2.4.4.
aMaxTODError	Integer	An estimate of the maximum error (in 10 ns units) in the TX_START_OF_FRAME_OFFSET value in the PHY-TXSTART.confirm(TXSTATUS) primitive. The estimated maximum error includes any error due to implementation component and environmental (including temperature) variability.

Name	Type	Description
aMaxTOAError	Integer	An estimate of the maximum error (in 10 ns units) in the RX_START_OF_FRAME_OFFSET value in the PHY-RXSTART.indicate(RXVECTOR) primitive. The estimated maximum error includes any error due to implementation component and environmental (including temperature) variability.
aTxPmdTxStartRFDelay	Integer	The delay (in units of 0.5 ns) between PMD_TXSTART.request being issued and the first frame energy sent by the transmitting port, for the current channel.
aTxPmdTxStartRMS	Integer	The RMS time of departure error (in units of 0.5 ns), where the time of departure error equals the difference between TIME_OF_DEPARTURE and the time of departure measured by a reference entity using a clock synchronized to the start time and mean frequency of the local PHY entity's clock.

6.5.4.3 When generated

This primitive is issued by the PHY entity in response to a PLME-CHARACTERISTICS.request primitive.

6.5.4.4 Effect of receipt

The receipt of this primitive provides the operational characteristics of the PHY entity.

6.5.5 PLME-DSSSTESTMODE.request

6.5.5.1 Function

This primitive requests that the DSSS PHY entity enter a test mode operation. The parameters associated with this primitive are considered as recommendations and are optional in any particular implementation.

6.5.5.2 Semantics of the service primitive

The primitive parameters are as follows:

```
PLME-DSSSTESTMODE.request(
                              TEST_ENABLE,
                              TEST_MODE,
                              SCRAMBLE_STATE,
                              SPREADING_STATE,
                              DATA_TYPE,
                              DATA_RATE;
                              PREAMBLE_TYPE;
                              MODULATION_CODE_TYPE;
                              )
```

Name	Type	Valid range	Description
TEST_ENABLE	Boolean	true, false	If true, enables the PHY test mode according to the remaining parameters.
TEST_MODE	integer	1, 2, 3	TEST_MODE selects one of three operational states: 01 = transparent receive 02 = continuous transmit 03 = 50% duty cycle
SCRAMBLE_STATE	Boolean	true, false	If true, sets the operational state of the scrambler to ON.

Name	Type	Valid range	Description
SPREADING_STATE	Boolean	true, false	If true, selects the operational state of the chipping.
DATA_TYPE	integer	1, 2, 3	Selects one of three data patterns to be used for the transmit portions of the tests, e.g., all ones, all zeros, and random data patterns.
DATA_RATE	integer	2, 3, 4, 5, 6, 9, 11, 12, 18, 22, 24, 27, 36, 44, 48, 54, 66, 72, 96, 108	Selects among rates: 02 = 1 Mb/s 03 = 1.5 Mb/s 04 = 2 Mb/s 05 = 2.5 Mb/s 06 = 3 Mb/s 09 = 4.5 Mb/s 11 = 5.5 Mb/s 12 = 6 Mb/s 18 = 9 Mb/s 22 = 11 Mb/s 24 = 12 Mb/s 27 = 13.5 Mb/s 36 = 18 Mb/s 44 = 22 Mb/s 48 = 24 Mb/s 54 = 27 Mb/s 66 = 33 Mb/s 72 = 36 Mb/s 96 = 48 Mb/s 108 = 54 Mb/s
PREAMBLE_TYPE	Boolean	null, 0, 1	Selects the preamble length: 0 = long 1 = short Can be null.
MODULATION_CODE_TYPE	Integer	null, 0, 1, 2	Selects among modulation options: 0 = no optional modulation modes 1 = optional ERP-PBCC modes 2 = optional DSSS-OFDM modes Can be null.

The rate for DATA_Rate=05 is rounded up to the next higher 0.5 Mb/s value.

6.5.5.3 When generated

This primitive is generated at any time to enter the DSSS PHY test mode.

6.5.5.4 Effect of receipt

Receipt of this primitive by the PHY causes the DSSS PHY entity to enter the test mode of operation.

6.5.6 PLME-DSSSTESTOUTPUT.request

6.5.6.1 Function

This optional primitive is a request by the SME to enable selected test signals from the PHY. The parameters associated with this primitive are considered as recommendations and are optional in any particular implementation.

6.5.6.2 Semantics of the service primitive

The primitive parameter is as follows:

```
PLME-DSSSTESTOUTPUT.request(
                              TEST_OUTPUT
                              )
```

Name	Type	Valid range	Description
TEST_OUTPUT	Boolean	true, false	If true, enables the selected test signals for testing DS PHY.

TEST_OUTPUT enables and disables selected signals for debugging and testing the PHY. Some signals that can be available for output are the PHY-TXSTART.request, PHY-RXSTART.indication(RXVECTOR), and PHY-CCA.indication primitives, the chipping clock, the data clock, the symbol clock, transmit (TX) data, and receive (RX) data.

6.5.6.3 When generated

This primitive is generated at any time to enable the test outputs when in the DSSS PHY test mode.

6.5.6.4 Effect of receipt

Receipt of this primitive by the DSSS PHY causes the DSSS PHY entity to enable the test outputs using the modes set by the most recent PLME-DSSSTESTMODE.request primitive.

6.5.7 PLME-TXTIME.request

6.5.7.1 Function

This primitive is a request for the PHY to calculate the time required to transmit onto the WM a PPDU containing a specified length PSDU, and using a specified format, data rate, and signalling.

6.5.7.2 Semantics of the service primitive

This primitive provides the following parameter:

```
PLME-TXTIME.request(
                          TXVECTOR
                          )
```

The TXVECTOR represents a list of parameters that the MAC sublayer provides to the local PHY entity in order to transmit a PSDU, as further described in 7.3.4.5, 18.4 and 20.4 (which defines the local PHY entity).

6.5.7.3 When generated

This primitive is issued by the MAC sublayer to the PHY entity when the MAC sublayer needs to determine the time required to transmit a particular PSDU.

6.5.7.4 Effect of receipt

The effect of receipt of this primitive by the PHY entity is to generate a PHY-TXTIME.confirm primitive that conveys the required transmission time.

6.5.8 PLME-TXTIME.confirm

6.5.8.1 Function

This primitive indicates the time required to transmit the PPDU described in the corresponding PLME-TXTIME.request.

6.5.8.2 Semantics of the service primitive

This primitive provides the following parameter:

```
PLME-TXTIME.confirm(
                                        TXTIME
                                        )
```

The TXTIME represents the time, in microseconds, required to transmit the PPDU described in the corresponding PLME-TXTIME.request primitive. If the calculated time includes a fractional microsecond, the TXTIME value is rounded up to the next higher integer.

6.5.8.3 When generated

This primitive is issued by the local PHY entity in response to a PLME-TXTIME.request primitive.

6.5.8.4 Effect of receipt

The receipt of this primitive provides the MAC sublayer with the PPDU transmission time.

7. PHY service specification

7.1 Scope

The PHY services provided to the IEEE 802.11 WLAN MAC are described in this clause. Different PHYs are defined as part of this standard. Each PHY can consist of two protocol functions as follows:

a) A PHY convergence function, which adapts the capabilities of the PMD system to the PHY service. This function is supported by the PLCP, which defines a method of mapping the IEEE 802.11 MPDUs into a framing format suitable for sending and receiving user data and management information between two or more STAs using the associated PMD system.

b) A PMD system, whose function defines the characteristics of, and method of transmitting and receiving data through, a WM between two or more STAs.

Each PMD sublayer may require the definition of a unique PLCP. If the PMD sublayer already provides the defined PHY services, the PHY convergence function might be null.

7.2 PHY functions

The protocol reference model for the IEEE 802.11 architecture is shown in Figure 4-14 (in 4.9). Most PHY definitions contain three functional entities: the PMD function, the PHY convergence function, and the layer management function.

The PHY service is provided to the MAC entity at the STA through a SAP, called the PHY-SAP, as shown in Figure 4-14. A set of primitives might also be defined to describe the interface between the PLCP sublayer and the PMD sublayer, called the PMD_SAP.

7.3 Detailed PHY service specifications

7.3.1 Scope and field of application

The services provided by the PHY to the IEEE 802.11 MAC are specified in this subclause. These services are described in an abstract way and do not imply any particular implementation or exposed interface.

7.3.2 Overview of the service

The PHY function as shown in Figure 4-14 is separated into two sublayers: the PLCP sublayer and the PMD sublayer. The function of the PLCP sublayer is to provide a mechanism for transferring MPDUs between two or more STAs over the PMD sublayer.

7.3.3 Overview of interactions

The primitives associated with communication between the IEEE 802.11 MAC sublayer and the IEEE 802.11 PHY fall into two basic categories:

a) Service primitives that support MAC peer-to-peer interactions;

b) Service primitives that have local significance and support sublayer-to-sublayer interactions.

7.3.4 Basic service and options

7.3.4.1 General

All of the service primitives described here are considered mandatory unless otherwise specified.

7.3.4.2 PHY-SAP peer-to-peer service primitives

Table 7-1 indicates the primitives for peer-to-peer interactions.

Table 7-1—PHY-SAP peer-to-peer service primitives

Primitive	Request	Indicate	Confirm
PHY-DATA	X	X	X

7.3.4.3 PHY-SAP sublayer-to-sublayer service primitives

Table 7-2 indicates the primitives for sublayer-to-sublayer interactions.

Table 7-2—PHY-SAP sublayer-to-sublayer service primitives

Primitive	Request	Indicate	Confirm
PHY-TXSTART	X		X
PHY-TXEND	X		X
PHY-CCARESET	X		X
PHY-CCA		X	
PHY-RXSTART		X	
PHY-RXEND		X	
PHY-CONFIG	X		X

7.3.4.4 PHY-SAP service primitives parameters

Table 7-3 shows the parameters used by one or more of the PHY-SAP service primitives.

Table 7-3—PHY-SAP service primitive parameters

Parameter	Associated primitive	Value
DATA	PHY-DATA.request PHY-DATA.indication	Octet value X'00'–X'FF'
TXVECTOR	PHY-TXSTART.request	A set of parameters
STATUS	PHY-CCA.indication	(BUSY, [channel-list]) (IDLE)
RXVECTOR	PHY-RXSTART.indication PHY-RXEND.indication	A set of parameters
RXERROR	PHY-RXEND.indication	NoError, FormatViolation, CarrierLost, UnsupportedRate

Table 7-3—PHY-SAP service primitive parameters *(continued)*

Parameter	Associated primitive	Value
IPI-STATE	PHY-CCARESET.request PHY-CCARESET.confirm	IPI-ON, IPI-OFF
IPI-REPORT	PHY-CCA.indication PHY-CCARESET.confirm	A set of IPI values for the preceding time interval
PHYCONFIG_VECTOR	PHY-CONFIG	A set of parameters
TXSTATUS	PHY-TXSTART.confirm	A set of parameters

7.3.4.5 Vector descriptions

Several service primitives include a parameter vector. This vector is a list of parameters that may vary depending on the PHY type. Table 7-4 lists the minimum parameter values required by the MAC or PHY in each of the parameter vectors. Parameters in the vectors that are management rather than MAC may be specific to the PHY and are listed in the clause covering that PHY.

Table 7-4—Vector descriptions

Parameter	Associated vector	Value
DATARATE	TXVECTOR, RXVECTOR	PHY dependent. The name of the field used to specify the Tx data rate and report the Rx data rate may vary for different PHYs.
LENGTH	TXVECTOR, RXVECTOR	PHY dependent
ACTIVE_RXCHAIN_SET	PHYCONFIG_VECTOR	The ACTIVE_RXCHAIN_SET parameter indicates which receive chains of the available receive chains are active. The length of the field is 8 bits. A 1 in bit position *n* indicates that the receive chain numbered *n* is used. At most 4 bits out of 8 may be set to 1.
OPERATING_CHANNEL	PHYCONFIG_VECTOR	The operating channel the PHY is configured use.
CHANNEL_OFFSET	PHYCONFIG_VECTOR	Enumerated type: CH_OFFSET_NONE indicates operation in 20 MHz HT STAs. CH_OFFSET_ABOVE indicates operation in 40 MHz with the secondary channel above the primary. CH_OFFSET_BELOW indicates operation in 40 MHz with the secondary channel below the primary.

The Clause 20 PHY TXVECTOR and RXVECTOR contain additional parameters related to the operation of the Clause 20 PHY modes of operation as described in 20.2. In certain modes of operation, the DATARATE parameter is replaced by a MCS value. The mapping from Clause 20 MCS to data rate is defined in 20.6.

7.3.5 PHY-SAP detailed service specification

7.3.5.1 Introduction

Subclause 7.3.5 describes the services provided by each PHY primitive.

7.3.5.2 PHY-DATA.request

7.3.5.2.1 Function

This primitive defines the transfer of an octet of data from the MAC sublayer to the local PHY entity.

7.3.5.2.2 Semantics of the service primitive

The primitive provides the following parameter:

```
  PHY-DATA.request(
                              DATA
                              )
```

The DATA parameter is an octet of value X'00' to X'FF'.

7.3.5.2.3 When generated

This primitive is generated by the MAC sublayer to transfer an octet of data to the PHY entity. This primitive can only be issued following a transmit initialization response (a PHY-TXSTART.confirm primitive) from the PHY.

7.3.5.2.4 Effect of receipt

The receipt of this primitive by the PHY entity causes the PLCP transmit state machine to transmit an octet of data. When the PHY entity receives the octet, it issues a PHY-DATA.confirm primitive to the MAC sublayer.

7.3.5.3 PHY-DATA.indication

7.3.5.3.1 Function

This primitive indicates the transfer of data from the PHY to the local MAC entity.

7.3.5.3.2 Semantics of the service primitive

The primitive provides the following parameter:

```
  PHY-DATA.indication(
                              DATA
                              )
```

The DATA parameter is an octet of value X'00' to X'FF'.

7.3.5.3.3 When generated

The PHY-DATA.indication primitive is generated by a receiving PHY entity to transfer the received octet of data to the local MAC entity. The time between receipt of the last bit of the provided octet from the WM and the receipt of this primitive by the MAC entity is the sum of aRXRFDelay + aRxPLCPDelay.

7.3.5.3.4 Effect of receipt

The effect of receipt of this primitive by the MAC is unspecified.

7.3.5.4 PHY-DATA.confirm

7.3.5.4.1 Function

This primitive is issued by the PHY to the local MAC entity to confirm the transfer of data from the MAC entity to the PHY.

7.3.5.4.2 Semantics of the service primitive

The semantics of the primitive are as follows:

```
   PHY-DATA.confirm
```

This primitive has no parameters.

7.3.5.4.3 When generated

This primitive is issued by the PHY to the MAC entity when the PLCP has completed the transfer of data from the MAC entity to the PHY. The PHY issues this primitive in response to every PHY-DATA.request primitive issued by the MAC sublayer.

7.3.5.4.4 Effect of receipt

The receipt of this primitive by the MAC causes the MAC to start the next MAC entity request.

7.3.5.5 PHY-TXSTART.request

7.3.5.5.1 Function

This primitive is a request by the MAC sublayer to the local PHY entity to start the transmission of a PSDU.

7.3.5.5.2 Semantics of the service primitive

The primitive provides the following parameter:

```
   PHY-TXSTART.request(
                              TXVECTOR
                              )
```

The TXVECTOR represents a list of parameters that the MAC sublayer provides to the local PHY entity in order to transmit a PSDU. This vector contains both PLCP and PHY management parameters. The minimum required PHY parameters are listed in 7.3.4.5.

7.3.5.5.3 When generated

This primitive is issued by the MAC sublayer to the PHY entity when the MAC sublayer needs to begin the transmission of a PSDU.

7.3.5.5.4 Effect of receipt

The effect of receipt of this primitive by the PHY entity is to start the local transmit state machine.

The behavior expected by the MAC pursuant to the issuance of PHY-TXSTART.request primitive is shown in Figure 9-14 (in 9.3.7).

7.3.5.6 PHY-TXSTART.confirm

7.3.5.6.1 Function

This primitive is issued by the PHY to the local MAC entity to confirm the start of a transmission and to indicate parameters related to the start of the transmission. The PHY issues this primitive in response to every PHY-TXSTART.request primitive issued by the MAC sublayer.

7.3.5.6.2 Semantics of the service primitive

The semantics of the primitive are as follows:

```
PHY-TXSTART.confirm(
                        TXSTATUS
                        )
```

The TXSTATUS represents a list of parameters that the local PHY entity provides to the MAC sublayer related to the transmission of an MPDU. This vector contains both PLCP and PHY operational parameters. The required PHY parameters are listed in 7.3.4.4.

7.3.5.6.3 When generated

This primitive is issued by the PHY to the MAC entity once all of the following conditions are met:

- The PHY has received a PHY-TXSTART.request primitive from the MAC entity.
- The PLCP has issued PMD.TX STATUS.request primitive if dot11MgmtOptionTODActivated is true and the TXVECTOR parameter TIME_OF_DEPARTURE_REQUESTED in the PHY-TXSTART.request(TXVECTOR) primitive is true.
- The PHY is ready to begin accepting outgoing data octets from the MAC.

7.3.5.6.4 Effect of receipt

The receipt of this primitive by the MAC entity causes the MAC to start the transfer of data octets. Parameters in the TXSTATUS vector may be included in transmitted frames so that recipients on multiple channels can compensate for differences in the transmit time of the frames, and so to determine the time differences of air propagation times between transmitter and pairs of recipients and hence to compute the location of the transmitter via multilateration. See Annex T. In addition, the TXSTATUS vector may include the TX_START_OF_FRAME_OFFSET.

7.3.5.7 PHY-TXEND.request

7.3.5.7.1 Function

This primitive is a request by the MAC sublayer to the local PHY entity that the current transmission of the PSDU be completed.

7.3.5.7.2 Semantics of the service primitive

The semantics of the primitive are as follows:

```
PHY-TXEND.request
```

This primitive has no parameters.

7.3.5.7.3 When generated

This primitive is generated when the MAC sublayer has received the last PHY-DATA.confirm primitive from the local PHY entity for the PSDU currently being transferred.

7.3.5.7.4 Effect of receipt

The effect of receipt of this primitive by the local PHY entity is to stop the transmit state machine.

7.3.5.8 PHY-TXEND.confirm

7.3.5.8.1 Function

This primitive is issued by the PHY to the local MAC entity to confirm the completion of a transmission. The PHY issues this primitive in response to every PHY-TXEND.request primitive issued by the MAC sublayer.

7.3.5.8.2 Semantics of the service primitive

The semantics of the primitive are as follows:

```
PHY-TXEND.confirm
```

This primitive has no parameters.

7.3.5.8.3 When generated

This primitive is issued by the PHY to the MAC entity when the PHY has received a PHY-TXEND.request primitive immediately after transmitting the end of the last bit of the last data octet indicating that the symbol containing the last data octet has been transferred and any Signal Extension has expired.

7.3.5.8.4 Effect of receipt

The receipt of this primitive by the MAC entity provides the time reference for the contention backoff protocol.

7.3.5.9 PHY-CCARESET.request

7.3.5.9.1 Function

This primitive is a request by the MAC sublayer to the local PHY entity to reset the CCA state machine and to turn IPI reporting on and off by means of the IPI-STATE parameter.

7.3.5.9.2 Semantics of the service primitive

The primitive provides the following parameter:

```
PHY-CCARESET.request(
                          IPI-STATE
                          )
```

The IPI-STATE parameter is present if dot11RadioMeasurementActivated is true. The IPI-STATE parameter can be one of two values: IPI-ON or IPI-OFF. The parameter value is IPI-ON when the MAC sublayer is requesting the PHY entity to report IPI values when the PHY is neither receiving nor transmitting an MPDU. IPI-ON turns on IPI reporting in the PHY entity. IPI-OFF turns off IPI reporting in the PHY entity.

7.3.5.9.3 When generated

This primitive is generated by the MAC sublayer for the local PHY entity at the end of a NAV timer. This request can be used by some PHY implementations that may synchronize antenna diversity with slot timings.

7.3.5.9.4 Effect of receipt

The effect of receipt of this primitive by the PHY entity is to reset the PLCP CS/CCA timers to the state appropriate for the end of a received frame. If IPI-STATE parameter is IPI-ON, the PHY entity collects IPI values when it is not transmitting or receiving and provides those values to the MAC sublayer using the IPI-REPORT parameter.

7.3.5.10 PHY-CCARESET.confirm

7.3.5.10.1 Function

This primitive is issued by the PHY to the local MAC entity to confirm that the PHY has reset the CCA state machine and to provide observed IPI values when IPI reporting is turned on.

7.3.5.10.2 Semantics of the service primitive

The primitive provides the following parameters:

```
  PHY-CCARESET.confirm(
                              IPI-STATE,
                              IPI-REPORT
                              )
```

The IPI-STATE parameter is present if dot11RadioMeasurementActivated is true. The IPI-STATE parameter can be one of two values: IPI-ON or IPI-OFF. The IPI-STATE value shall be set to the value of IPI-STATE received by the PHY entity in the most recent PHY-CCARESET.request primitive.

The IPI-REPORT parameter is present if dot11RadioMeasurementActivated is true and if IPI reporting was turned on prior to the receipt of the latest PHY-CCARESET.request primitive. The IPI-REPORT parameter provides a set of IPI values for a time interval. The set of IPI values are recent values observed by the PHY entity since the generation of the most recent PHY-TXEND.confirm, PHY-RXEND.indication, PHY-CCARESET.confirm, or PHY_CCA.indication primitive, whichever occurred latest.

7.3.5.10.3 When generated

This primitive is issued by the PHY to the MAC entity when the PHY has received a PHY-CCA-RESET.request primitive.

7.3.5.10.4 Effect of receipt

The effect of receipt of this primitive by the MAC is unspecified.

7.3.5.11 PHY-CCA.indication

7.3.5.11.1 Function

This primitive is an indication by the PHY to the local MAC entity of the current state of the medium and to provide observed IPI values when IPI reporting is turned on.

7.3.5.11.2 Semantics of the service primitive

The primitive provides the following parameters:

```
PHY-CCA.indication(
                        STATE,
                        IPI-REPORT,
                        channel-list
                        )
```

The STATE parameter can be one of two values: BUSY or IDLE. The parameter value is BUSY if the assessment of the channel(s) by the PHY determines that the channel(s) are not available. Otherwise, the value of the parameter is IDLE.

The IPI-REPORT parameter is present if dot11RadioMeasurementActivated is true and if IPI reporting has been turned on by the IPI-STATE parameter. The IPI-REPORT parameter provides a set of IPI values for a time interval. The set of IPI values may be used by the MAC sublayer for Radio Measurement purposes. The set of IPI values are recent values observed by the PHY entity since the generation of the most recent PHY-TXEND.confirm, PHY-RXEND.indication, PHY-CCARESET.confirm, or PHY_CCA.indication primitive, whichever occurred latest.

When STATE is IDLE or when, for the type of PHY in operation, CCA is determined by a single channel, the channel-list parameter is absent. Otherwise, it carries a set indicating which channels are busy, represented by the values {primary}, {primary, secondary}, and {secondary}.

7.3.5.11.3 When generated

This primitive is generated within aCCATime of the occurrence of a change in the status of the channel(s) from channel idle to channel busy or from channel busy to channel idle. This includes the period of time when the PHY is receiving data. Refer to specific PHY clauses for details about CCA behavior for a given PHY.

If the STA is an HT STA and the operating channel width is 20 MHz, the PHY maintains the channel busy indication until the period indicated by the LENGTH field has expired, where the LENGTH field is

- In a valid SIG field if the format of the PPDU is NON_HT
- In a valid HT-SIG field if the format of the PPDU is HT_MF or HT_GF

If the STA is an HT STA and the operating channel width is 40 MHz, the PHY maintains the channel busy indication until the period indicated by the LENGTH field has expired, where the LENGTH field is

- In a valid SIG field if the format of the PPDU is NON_HT and the PPDU is received in the primary channel
- In a valid HT-SIG field if the format of the PPDU is HT_MF or HT_GF provided that the PPDU is either a 20 MHz PPDU received in the primary channel or a 40 MHz PPDU

7.3.5.11.4 Effect of receipt

The effect of receipt of this primitive by the MAC is unspecified.

7.3.5.12 PHY-RXSTART.indication

7.3.5.12.1 Function

This primitive is an indication by the PHY to the local MAC entity that the PLCP has received a valid start of a PPDU, including a valid PLCP header.

7.3.5.12.2 Semantics of the service primitive

The primitive provides the following parameter:

```
PHY-RXSTART.indication(
                                RXVECTOR
                                )
```

The RXVECTOR represents a list of parameters that the PHY provides the local MAC entity upon receipt of a valid PLCP header or upon receipt of the last PSDU data bit in the received frame. The required parameters are listed in 7.3.4.5.

7.3.5.12.3 When generated

This primitive is generated by the local PHY entity to the MAC sublayer when the PHY has successfully validated the PLCP header at the start of a new PPDU.

After generating a PHYRXSTART.indication primitive, the PHY is expected to maintain physical medium busy status (not generating a PHY-CCA.indication(IDLE) primitive) during the period required by that PHY to transfer a frame of the indicated LENGTH at the indicated DATARATE. This physical medium busy condition should be maintained even if a PHY-RXEND.indication(CarrierLost) or a PHYRXEND.indication(Format-Violation) primitive is generated by the PHY prior to the end of this period.

7.3.5.12.4 Effect of receipt

The effect of receipt of this primitive by the MAC is unspecified.

7.3.5.13 PHY-RXEND.indication

7.3.5.13.1 Function

This primitive is an indication by the PHY to the local MAC entity that the PSDU currently being received is complete.

7.3.5.13.2 Semantics of the service primitive

The primitive provides the following parameters:

```
PHY-RXEND.indication(
                                RXERROR,
                                RXVECTOR
                                )
```

The RXERROR parameter can convey one or more of the following values: NoError, FormatViolation, CarrierLost, or UnsupportedRate. A number of error conditions may occur after the PLCP's receive state machine has detected what appears to be a valid preamble and SFD. The following describes the parameter returned for each of those error conditions.

— *NoError*. This value is used to indicate that no error occurred during the receive process in the PLCP.

— *FormatViolation*. This value is used to indicate that the format of the received PPDU was in error.

— *CarrierLost*. This value is used to indicate that during the reception of the incoming PSDU, the carrier was lost and no further processing of the PSDU can be accomplished.

— *UnsupportedRate*. This value is used to indicate that during the reception of the incoming PPDU, a nonsupported date rate was detected.

The RXVECTOR represents a list of parameters that the PHY provides the local MAC entity upon receipt of a valid PLCP header or upon receipt of the last PSDU data bit in the received frame. RXVECTOR is an included parameter only when dot11RadioMeasurementActivated is true. This vector may contain both MAC and MAC management parameters. The required parameters are listed in 7.3.4.5.

7.3.5.13.3 When generated

This primitive is generated by the PHY for the local MAC entity to indicate that the receive state machine has completed a reception with or without errors. When a Signal Extension is present, the primitive is generated at the end of the Signal Extension.

In the case of an RXERROR value of NoError, the MAC uses the PHY-RXEND.indication primitive as reference for channel access timing, as shown in Figure 9-14 (in 9.3.7).

7.3.5.13.4 Effect of receipt

The effect of receipt of this primitive is for the MAC to begin inter-frame space processing, as described in 9.3.7.

7.3.5.14 PHY-CONFIG.request

7.3.5.14.1 Function

This primitive is a request by the MAC sublayer to the local PHY entity to configure the PHY.

7.3.5.14.2 Semantics of the service primitive

The primitive provides the following parameter:

```
  PHY-CONFIG.request(
                                PHYCONFIG_VECTOR
                                )
```

7.3.5.14.3 When generated

This primitive is generated by the MAC sublayer for the local PHY entity when it desires to change the configuration of the PHY.

7.3.5.14.4 Effect of receipt

The effect of receipt of this primitive by the PHY is to apply the parameters provided with the primitive and to configure the PHY for future operation.

7.3.5.15 PHY-CONFIG.confirm

7.3.5.15.1 Function

This primitive is issued by the PHY to the local MAC entity to confirm that the PHY has applied the parameters provided in the PHY-CONFIG.request primitive.

7.3.5.15.2 Semantics of the service primitive

The semantics of the primitive are as follows:

```
                PHY-CONFIG.confirm
```

This primitive has no parameters.

7.3.5.15.3 When generated

This primitive is issued by the PHY to the MAC entity when the PHY has received and successfully applied the parameters in the PHY-CONFIG.request primitive.

7.3.5.15.4 Effect of receipt

The effect of the receipt of this primitive by the MAC is unspecified.

7.4 PHY management

The MIB comprises the managed objects, attributes, actions, and notifications required to manage a STA. The definition of these managed objects, attributes, actions, and notifications, as well as their structure, is presented in Annex C.

8. Frame formats

8.1 General requirements

The format of the MAC frames is specified in this clause. A STA shall be able properly to construct a subset of the frames specified in this clause for transmission and to decode a (potentially different) subset of the frames specified in this clause upon validation following reception. The particular subset of these frames that a STA constructs and decodes is determined by the functions supported by that particular STA. All STAs shall be able to validate every received frame using the frame check sequence (FCS) and to interpret certain fields from the MAC headers of all frames.

A compliant STA shall transmit frames using only the frame formats described in Clause 8.

8.2 MAC frame formats

8.2.1 Basic components

Each frame consists of the following basic components:

a) A *MAC header*, which comprises frame control, duration, address, optional sequence control information, optional QoS Control information (QoS data frames only), and optional HT Control fields (+HTC frames only);

b) A variable-length *frame body*, which contains information specific to the frame *type* and *subtype*;

c) A *FCS*, which contains an IEEE 32-bit CRC.

8.2.2 Conventions

Structures defined in the MAC sublayer are described as a sequence of fields in specific order. Each figure in Clause 8 depicts the fields/subfields as they appear in the MAC frame and in the order in which they are passed to the physical layer convergence procedure (PLCP), from left to right.

In figures, all bits within fields are numbered, from 0 to k, where the length of the field is $k + 1$ bits. Bits within numeric fields that are longer than a single bit are depicted in increasing order of significance, i.e., with the lowest numbered bit having the least significance. The octet boundaries within a field can be obtained by taking the bit numbers of the field modulo 8. Octets within numeric fields that are longer than a single octet are depicted in increasing order of significance, from lowest numbered bit to highest numbered bit. The octets in fields longer than a single octet are sent to the PLCP in order from the octet containing the lowest numbered bits to the octet containing the highest numbered bits.

Any field containing a CRC is an exception to this convention and is transmitted commencing with the coefficient of the highest-order term.

MAC addresses are assigned as ordered sequences of bits. The Individual/Group bit is always transferred first and is bit 0 of the first octet.

Organizationally unique identifiers (OUIs) and Organization Identifiers are specified in two forms: an ordered sequence of octets, and a numeric form. Treating the OUI or Organization Identifier as an ordered sequence of octets, the leftmost octet is always transferred first. This is equivalent to transmitting the most significant octet of the numeric form first.

Values specified in decimal are coded in natural binary unless otherwise stated. The values in Table 8-1 are in binary, with the bit assignments shown in the table. Values in other tables are shown in decimal notation.

Reception, in references to frames or fields within frames (e.g., received Beacon frames or a received Duration/ID field), applies to MPDUs or MAC management protocol data units (MMPDUs) indicated from the PHY layer without error and validated by FCS within the MAC sublayer. Without further qualification, *reception* by the MAC sublayer implies that the frame contents are valid, and that the protocol version is supported (see 8.2.4.1.2), with no implication regarding frame addressing or regarding whether the frame type or other fields in the MAC header are meaningful to the MAC entity that has received the frame.

A frame that contains the HT Control field, including the Control Wrapper frame, is referred to as a +HTC frame.

A QoS Data frame that is transmitted by a mesh STA is referred to as a Mesh Data frame.

Parentheses enclosing portions of names or acronyms are used to designate a set of related names that vary based on the inclusion of the parenthesized portion. For example,

— *QoS +CF-Poll frame* refers to the three QoS data subtypes that include "+CF-Poll": the QoS Data+CF-Poll frame, subtype 1010; QoS Data+CF-Ack+CF-Poll frame, subtype 1011; and QoS CF-Ack+CF-Poll frame, subtype 1111.

— *QoS CF-Poll frame* refers specifically to the QoS CF-Poll frame, subtype 1110.

— *QoS (+)CF-Poll frame* refers to all four QoS data subtypes with CF-Poll: the QoS CF-Poll frame, subtype 1110; the QoS CF-Ack+CF-Poll frame, subtype 1111; the QoS Data+CF-Poll frame, subtype 1010; and the QoS Data+CF-Ack+CF-Poll frame, subtype 1011.

— *QoS (+)Null frame* refers to all three QoS data subtypes with "no data": the QoS Null (no data) frame, subtype 1100; the QoS CF-Poll (no data) frame, subtype 1110; and the QoS CF-Ack+CF-Poll frame, subtype 1111.

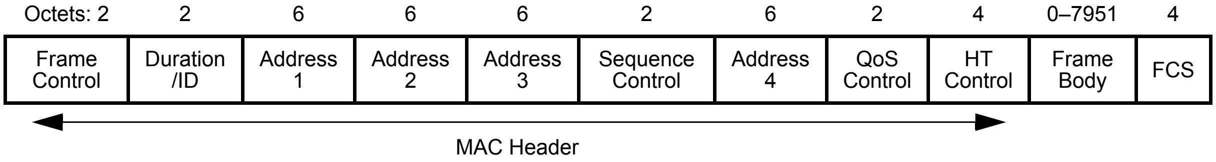
— *QoS +CF-Ack frame* refers to the three QoS data subtypes that include "+CF-Ack": the QoS Data+CF-Ack frame, subtype 1001; QoS Data+CF-Ack+CF-Poll frame, subtype 1011; and QoS CF-Ack+CF-Poll frame, subtype 1111.

— Whereas *(QoS) CF-Poll frame* refers to the QoS CF-Poll frame, subtype 1110, and the CF-Poll frame, subtype 0110.

Reserved fields and subfields are set to 0 upon transmission and are ignored upon reception.

8.2.3 General frame format

The MAC frame format comprises a set of fields that occur in a fixed order in all frames. Figure 8-1 depicts the general MAC frame format. The first three fields (Frame Control, Duration/ID, and Address 1) and the last field (FCS) in Figure 8-1 constitute the minimal frame format and are present in all frames, including reserved types and subtypes. The fields Address 2, Address 3, Sequence Control, Address 4, QoS Control, HT Control, and Frame Body are present only in certain frame types and subtypes. Each field is defined in 8.2.4. The format of each of the individual subtypes of each frame type is defined in 8.3. The components of management frame bodies are defined in 8.4. The formats of management frames of subtype Action are defined in 8.5.



Octets: 2	2	6	6	6	2	6	2	4	0–7951	4
Frame Control	Duration /ID	Address 1	Address 2	Address 3	Sequence Control	Address 4	QoS Control	HT Control	Frame Body	FCS

Figure 8-1—MAC frame format

The Frame Body field is of variable size. The maximum frame body size is determined by the maximum MSDU size (2304 octets), plus the length of the Mesh Control field (6, 12, or 18 octets) if present, the maximum unencrypted MMPDU size excluding the MAC header and FCS (2304 octets) or the maximum A-MSDU size (3839 or 7935 octets, depending upon the STA’s capability), plus any overhead from security encapsulation.

8.2.4 Frame fields

8.2.4.1 Frame Control field

8.2.4.1.1 General

The Frame Control field consists of the following subfields: Protocol Version, Type, Subtype, To DS, From DS, More Fragments, Retry, Power Management, More Data, Protected Frame, and Order. The format of the Frame Control field is illustrated in Figure 8-2.

	B0 B1	B2 B3	B4 B7	B8	B9	B10	B11	B12	B13	B14	B15
	Protocol Version	Type	Subtype	To DS	From DS	More Fragments	Retry	Power Management	More Data	Protected Frame	Order
Bits:	2	2	4	1	1	1	1	1	1	1	1

Figure 8-2—Frame Control field

8.2.4.1.2 Protocol Version field

The Protocol Version field is 2 bits in length and is invariant in size and placement across all revisions of this standard. For this standard, the value of the protocol version is 0. All other values are reserved. The revision level will be incremented only when a fundamental incompatibility exists between a new revision and the prior edition of the standard. See 9.24.2.

8.2.4.1.3 Type and Subtype fields

The Type field is 2 bits in length, and the Subtype field is 4 bits in length. The Type and Subtype fields together identify the function of the frame. There are three frame types: control, data, and management. Each of the frame types has several defined subtypes. In data frames, the most significant bit (MSB) of the Subtype field, b7, is defined as the QoS subfield. Table 8-1 defines the valid combinations of type and subtype. (The numeric values in Table 8-1 are shown in binary.)

Table 8-1—Valid type and subtype combinations

Type value b3 b2	Type description	Subtype value b7 b6 b5 b4	Subtype description
00	Management	0000	Association request
00	Management	0001	Association response
00	Management	0010	Reassociation request
00	Management	0011	Reassociation response
00	Management	0100	Probe request
00	Management	0101	Probe response
00	Management	0110	Timing Advertisement

Table 8-1—Valid type and subtype combinations *(continued)*

Type value b3 b2	Type description	Subtype value b7 b6 b5 b4	Subtype description
00	Management	0111	Reserved
00	Management	1000	Beacon
00	Management	1001	ATIM
00	Management	1010	Disassociation
00	Management	1011	Authentication
00	Management	1100	Deauthentication
00	Management	1101	Action
00	Management	1110	Action No Ack
00	Management	1111	Reserved
01	Control	0000–0110	Reserved
01	Control	0111	Control Wrapper
01	Control	1000	Block Ack Request (BlockAckReq)
01	Control	1001	Block Ack (BlockAck)
01	Control	1010	PS-Poll
01	Control	1011	RTS
01	Control	1100	CTS
01	Control	1101	ACK
01	Control	1110	CF-End
01	Control	1111	CF-End + CF-Ack
10	Data	0000	Data
10	Data	0001	Data + CF-Ack
10	Data	0010	Data + CF-Poll
10	Data	0011	Data + CF-Ack + CF-Poll
10	Data	0100	Null (no data)
10	Data	0101	CF-Ack (no data)
10	Data	0110	CF-Poll (no data)
10	Data	0111	CF-Ack + CF-Poll (no data)
10	Data	1000	QoS Data
10	Data	1001	QoS Data + CF-Ack
10	Data	1010	QoS Data + CF-Poll
10	Data	1011	QoS Data + CF-Ack + CF-Poll
10	Data	1100	QoS Null (no data)
10	Data	1101	Reserved
10	Data	1110	QoS CF-Poll (no data)
10	Data	1111	QoS CF-Ack + CF-Poll (no data)
11	Reserved	0000–1111	Reserved

Each Subtype field bit position is used to indicate a specific modification of the basic data frame (subtype 0). Frame Control bit 4 is set to 1 in data subtypes that include +CF-Ack, bit 5 is set to 1 in data subtypes that include +CF-Poll, bit 6 is set to 1 in data subtypes that contain no Frame Body field, and bit 7 is set to 1 in the QoS data subtypes, which have QoS Control fields in their MAC headers.

8.2.4.1.4 To DS and From DS fields

The meaning of the combinations of values for the To DS and From DS fields are shown in Table 8-2.

Table 8-2—To/From DS combinations in data frames

To DS and From DS values	Meaning
To DS = 0 From DS = 0	A data frame direct from one STA to another STA within the same IBSS, a data frame direct from one non-AP STA to another non-AP STA within the same BSS, or a data frame outside the context of a BSS, as well as all management and control frames.
To DS = 1 From DS = 0	A data frame destined for the DS or being sent by a STA associated with an AP to the Port Access Entity in that AP.
To DS = 0 From DS = 1	A data frame exiting the DS or being sent by the Port Access Entity in an AP, or a group addressed Mesh Data frame with Mesh Control field present using the three-address MAC header format.
To DS = 1 From DS = 1	A data frame using the four-address MAC header format. This standard defines procedures for using this combination of field values only in a mesh BSS.

8.2.4.1.5 More Fragments field

The More Fragments field is 1 bit in length and is set to 1 in all data or management type frames that have another fragment of the current MSDU or current MMPDU to follow. It is set to 0 in all other frames.

8.2.4.1.6 Retry field

The Retry field is 1 bit in length and is set to 1 in any data or management type frame that is a retransmission of an earlier frame. It is set to 0 in all other frames. A receiving STA uses this indication to aid in the process of eliminating duplicate frames.

8.2.4.1.7 Power Management field

The Power Management field is 1 bit in length and is used to indicate the power management mode of a STA. The value of this field is either reserved (as defined below) or remains constant in each frame from a particular STA within a frame exchange sequence (see Annex G). The value indicates the mode of the STA after the successful completion of the frame exchange sequence.

In an infrastructure BSS, the following applies:

- The Power Management field is reserved in all management frames that are not bufferable management frames.
- The Power Management field is reserved in all management frames transmitted by a STA to an AP with which it is not associated.
- The Power Management field is reserved in all frames transmitted by the AP.
- Otherwise, a value of 1 indicates that the STA will be in PS mode. A value of 0 indicates that the STA will be in active mode.

In an IBSS, the following applies:

- The Power Management field is reserved in all management frames that are not bufferable management frames and that are not individually addressed Probe Request frames.
- Otherwise, a value of 1 indicates that the STA will be in PS mode. A value of 0 indicates that the STA will be in active mode.

In an MBSS, the following applies:

- A value of 0 in group addressed frames, in management frames transmitted to nonpeer STAs, and in Probe Response frames indicates that the mesh STA will be in active mode towards all neighbor mesh STAs. A value of 1 in group addressed frames, in management frames transmitted to nonpeer STAs, and in Probe Response frames indicates that the mesh STA will be in deep sleep mode towards all nonpeer mesh power STAs.
- A value of 0 in individually addressed frames transmitted to a peer mesh STA indicates that the mesh STA will be in active mode towards this peer mesh STA A value of 1 in individually addressed frames transmitted to a peer mesh STA, except Probe Response frames, indicates that the mesh STA will be in either light sleep mode or deep sleep mode towards this peer mesh STA. When the QoS Control field is present in the frame, the Mesh Power Save Level subfield in the QoS Control field indicates whether the mesh STA will be in light sleep mode or in deep sleep mode for the recipient mesh STA as specified in 8.2.4.5.11.

The mesh power mode transition rules are described in 13.14.3.

8.2.4.1.8 More Data field

The More Data field is 1 bit in length and is used to indicate to a STA in PS mode that more BUs are buffered for that STA at the AP. The More Data field is valid in individually addressed data or management type frames transmitted by an AP to a STA in PS mode. A value of 1 indicates that at least one additional buffered BU is present for the same STA.

The More Data field is optionally set to 1 in individually addressed data type frames transmitted by a CF-Pollable STA to the PC in response to a CF-Poll to indicate that the STA has at least one additional buffered MSDU available for transmission in response to a subsequent CF-Poll.

For a STA in which the More Data Ack subfield of its QoS Capability element is 1 and that has APSD enabled, an AP optionally sets the More Data field to 1 in ACK frames to this STA to indicate that the AP has a pending transmission for the STA.

For a STA with TDLS peer PSM enabled and the More Data Ack subfield equal to 1 in the QoS Capability element of its transmitted TDLS Setup Request frame or TDLS Setup Response frame, a TDLS peer STA optionally sets the More Data field to 1 in ACK frames to this STA to indicate that it has a pending transmission for the STA.

The More Data field is 1 in individually addressed frames transmitted by a mesh STA to a peer mesh STA that is either in light sleep mode or in deep sleep mode for the corresponding mesh peering, when additional BUs remain to be transmitted to this peer mesh STA.

The More Data field is set to 0 in all other individually addressed frames.

The More Data field is set to 1 in group addressed frames transmitted by the AP when additional group addressed bufferable units (BUs) remain to be transmitted by the AP during this beacon interval. The More Data field is set to 0 in group addressed frames transmitted by the AP when no more group addressed BUs remain to be transmitted by the AP during this beacon interval and in all group addressed frames transmitted by non-AP STAs.

The More Data field is 1 in group addressed frames transmitted by a mesh STA when additional group addressed BUs remain to be transmitted. The More Data field is 0 in group addressed frames transmitted by a mesh STA when no more group addressed BUs remain to be transmitted.

8.2.4.1.9 Protected Frame field

The Protected Frame field is 1 bit in length. The Protected Frame field is set to 1 if the Frame Body field contains information that has been processed by a cryptographic encapsulation algorithm. The Protected Frame field is set to 1 only within data frames and within management frames of subtype Authentication, and individually addressed robust management frames. The Protected Frame field is set to 0 in all other frames. When the Protected Frame field is equal to 1, the Frame Body field is protected utilizing the cryptographic encapsulation algorithm and expanded as defined in Clause 11. The Protected Frame field is set to 0 in Data frames of subtype Null Function, CF-ACK (no data), CF-Poll (no data), CF-ACK+CF-Poll (no data), QoS Null (no data), QoS CF-Poll (no data), and QoS CF-ACK+CF-Poll (no data) (see, for example, 11.4.2.2 and 11.4.3.1 that show that the frame body needs to be 1 octet or longer to apply the encapsulation).

8.2.4.1.10 Order field

The Order field is 1 bit in length. It is used for two purposes:

- It is set to 1 in a non-QoS data frame transmitted by a non-QoS STA to indicate that the frame contains an MSDU, or fragment thereof, that is being transferred using the StrictlyOrdered service class.
- It is set to 1 in a QoS data or management frame transmitted with a value of HT_GF or HT_MF for the FORMAT parameter of the TXVECTOR to indicate that the frame contains an HT Control field.

Otherwise, the Order field is set to 0.

8.2.4.2 Duration/ID field

The Duration/ID field is 16 bits in length. The contents of this field vary with frame type and subtype, with whether the frame is transmitted during the CFP, and with the QoS capabilities of the sending STA. The contents of the field are defined as follows:

a) In control frames of subtype PS-Poll, the Duration/ID field carries the association identifier (AID) of the STA that transmitted the frame in the 14 least significant bits (LSB), and the 2 most significant bits (MSB) both set to 1. The value of the AID is in the range 1–2007.

b) In frames transmitted by the PC and non-QoS STAs, during the CFP, the Duration/ID field is set to a fixed value of 32 768.

c) In all other frames sent by non-QoS STAs and control frames sent by QoS STAs, the Duration/ID field contains a duration value as defined for each frame type in 8.3.

d) In data and management frames sent by QoS STAs, the Duration/ID field contains a duration value as defined for each frame type in 8.2.5.

See 9.24.3 on the processing of this field in received frames.

The encoding of the Duration/ID field is given in Table 8-3.

The Duration/ID fields in the MAC headers of MPDUs in an A-MPDU all carry the same value.

NOTE—The reference point for the Duration/ID field is the end of the PPDU carrying the MPDU. Setting the Duration/ID field to the same value in the case of A-MPDU aggregation means that each MPDU consistently specifies the same NAV setting.

Table 8-3—Duration/ID field encoding

Bits 0–13	Bit 14	Bit 15	Usage
0–32 767		0	Duration value (in microseconds) within all frames other than PS-Poll frames transmitted during the CP, and under HCF for frames transmitted during the CFP
0	0	1	Fixed value under point coordination function (PCF) within frames transmitted during the CFP
1–16 383	0	1	Reserved
0	1	1	Reserved
1–2007	1	1	AID in PS-Poll frames
2008–16 383	1	1	Reserved

8.2.4.3 Address fields

8.2.4.3.1 General

There are four address fields in the MAC frame format. These fields are used to indicate the basic service set identifier (BSSID), source address (SA), destination address (DA), transmitting STA address (TA), and receiving STA address (RA). Certain frames may not contain some of the address fields.

Certain address field usage is specified by the relative position of the address field (1–4) within the MAC header, independent of the type of address present in that field. For example, receiver address matching is always performed on the contents of the Address 1 field in received frames, and the receiver address of CTS and ACK frames is always obtained from the Address 2 field in the corresponding RTS frame, or from the frame being acknowledged.

8.2.4.3.2 Address representation

Each Address field contains a 48-bit address as defined in 9.2 of IEEE Std 802-2001.

8.2.4.3.3 Address designation

A MAC sublayer address is one of the following two types:

a) *Individual address.* The address assigned to a particular STA on the network.

b) *Group address.* A multidestination address, which may be in use by one or more STAs on a given network. The two kinds of group addresses are as follows:

   1) *Multicast-group address.* An address associated by higher level convention with a group of logically related STAs.

   2) *Broadcast address.* A distinguished, predefined group address that always denotes the set of all STAs on a given LAN. All ones are interpreted to be the broadcast address. This group is predefined for each communication medium to consist of all STAs actively connected to that medium; it is used to broadcast to all the active STAs on that medium.

The address space is also partitioned into locally administered and universal (globally administered) addresses. The nature of a body and the procedures by which it administers these universal (globally administered) addresses is beyond the scope of this standard. See IEEE Std 802-2001 for more information.

8.2.4.3.4 BSSID field

The BSSID field is a 48-bit field of the same format as an IEEE 802 MAC address. When dot11OCBActivated is false, the value of this field uniquely identifies each BSS. The value of this field, in an infrastructure BSS, is the MAC address currently in use by the STA in the AP of the BSS.

The value of this field in an IBSS is a locally administered IEEE MAC address formed from a 46-bit random number generated according to the procedure defined in 10.1.4. The individual/group bit of the address is set to 0. The universal/local bit of the address is set to 1. This mechanism is used to provide a high probability of selecting a unique BSSID.

The value of all 1s is used to indicate the wildcard BSSID. The wildcard value is not used in the BSSID field except where explicitly permitted in this standard. When dot11OCBActivated is true, the wildcard value shall be used in the BSSID field. When dot11OCBActivated is false and the BSSID field contains the wildcard value, the Address 1 (DA) field is also set to all 1s to indicate the broadcast address.

8.2.4.3.5 DA field

The DA field contains an IEEE MAC individual or group address that identifies the MAC entity or entities intended as the final recipient(s) of the MSDU (or fragment thereof) or A-MSDU, as defined in 8.3.2.1, contained in the frame body field.

8.2.4.3.6 SA field

The SA field contains an IEEE MAC individual address that identifies the MAC entity from which the transfer of the MSDU (or fragment thereof) or A-MSDU, as defined in 8.3.2.1, contained in the frame body field was initiated. The individual/group bit is always transmitted as a 0 in the source address.

8.2.4.3.7 RA field

The RA field contains an IEEE MAC individual or group address that identifies the intended immediate recipient STA(s), on the WM, for the information contained in the frame body field.

8.2.4.3.8 TA field

The TA field contains an IEEE MAC individual address that identifies the STA that has transmitted, onto the WM, the MPDU contained in the frame body field. The Individual/Group bit is always transmitted as a 0 in the transmitter address.

8.2.4.4 Sequence Control field

8.2.4.4.1 Sequence Control field structure

The Sequence Control field is 16 bits in length and consists of two subfields, the Sequence Number and the Fragment Number. The format of the Sequence Control field is illustrated in Figure 8-3. The sequence Control field is not present in control frames.

	B0 B3	B4 B15
	Fragment Number	Sequence Number
Bits:	4	12

Figure 8-3—Sequence Control field

8.2.4.4.2 Sequence Number field

The Sequence Number field is a 12-bit field indicating the sequence number of an MSDU, A-MSDU, or MMPDU. Each MSDU, A-MSDU, or MMPDU transmitted by a STA is assigned a sequence number. Sequence numbers are not assigned to control frames, as the Sequence Control field is not present.

Each fragment of an MSDU or MMPDU contains a copy of the sequence number assigned to that MSDU or MMPDU. The sequence number remains constant in all retransmissions of an MSDU, MMPDU, or fragment thereof.

8.2.4.4.3 Fragment Number field

The Fragment Number field is a 4-bit field indicating the number of each fragment of an MSDU or MMPDU. The fragment number is set to 0 in the first or only fragment of an MSDU or MMPDU and is incremented by one for each successive fragment of that MSDU or MMPDU. The fragment number is set to 0 in the only fragment of an A-MSDU. The fragment number remains constant in all retransmissions of the fragment.

8.2.4.5 QoS Control field

8.2.4.5.1 QoS Control field structure

The QoS Control field is a 16-bit field that identifies the TC or TS to which the frame belongs as well as various other QoS-related, A-MSDU related, and mesh-related information about the frame that varies by frame type, subtype, and type of transmitting STA. The QoS Control field is present in all data frames in which the QoS subfield of the Subtype field is equal to 1 (see 8.2.4.1.3). Each QoS Control field comprises five or eight subfields, as defined for the particular sender (HC or non-AP STA) and frame type and subtype. The usage of these subfields and the various possible layouts of the QoS Control field are described 8.2.4.5.2 to 8.2.4.5.12 and illustrated in Table 8-4.

See 9.12.1 for constraints on the contents of the QoS Control field when present in an A-MPDU.

Table 8-4—QoS Control field

Applicable frame (sub) types	Bits 0–3	Bit 4	Bits 5-6	Bit 7	Bits 8	Bit 9	Bit 10	Bits 11-15
QoS CF-Poll and QoS CF-Ack+CF-Poll frames sent by HC	TID	EOSP	Ack Policy	Reserved	TXOP Limit			
QoS Data+CF-Poll and QoS Data+CF-Ack+CF-Poll frames sent by HC	TID	EOSP	Ack Policy	A-MSDU Present	TXOP Limit			
QoS Data and QoS Data+CF-Ack frames sent by HC	TID	EOSP	Ack Policy	A-MSDU Present	AP PS Buffer State			
QoS Null frames sent by HC	TID	EOSP	Ack Policy	Reserved	AP PS Buffer State			
QoS Data and QoS Data+CF-Ack frames sent by non-AP STAs that are not a TPU buffer STA or a TPU sleep STA in a nonmesh BSS	TID	0	Ack Policy	A-MSDU Present	TXOP Duration Requested			
	TID	1	Ack Policy	A-MSDU Present	Queue Size			

Table 8-4—QoS Control field *(continued)*

Applicable frame (sub) types	Bits 0–3	Bit 4	Bits 5-6	Bit 7	Bits 8	Bit 9	Bit 10	Bits 11-15
QoS Null frames sent by non-AP STAs that are not a TPU buffer STA or a TPU sleep STA in a nonmesh BSS	TID	0	Ack Policy	Reserved	TXOP Duration Requested			
	TID	1	Ack Policy	Reserved	Queue Size			
QoS Data and QoS Data+CF-Ack frames sent by TPU buffer STAs in a nonmesh BSS	TID	EOSP	Ack Policy	A-MSDU Present	Reserved			
QoS Null frames sent by TPU buffer STAs in a nonmesh BSS	TID	EOSP	Ack Policy	Reserved	Reserved			
QoS Data and QoS Data+CF-Ack frames sent by TPU sleep STAs in a nonmesh BSS	TID	Reserved	Ack Policy	A-MSDU Present	Reserved			
QoS Null frames sent by TPU sleep STAs in a nonmesh BSS	TID	Reserved	Ack Policy	Reserved	Reserved			
All frames sent by mesh STAs in a mesh BSS	TID	EOSP	Ack Policy	A-MSDU Present	Mesh Control Present	Mesh Power Save Level	RSPI	Reserved

8.2.4.5.2 TID subfield

The TID subfield identifies the TC or TS to which the corresponding MSDU (or fragment thereof) or A-MSDU in the Frame Body field belongs. The TID subfield also identifies the TC or TS of traffic for which a TXOP is being requested, through the setting of TXOP duration requested or queue size. The encoding of the TID subfield depends on the access policy (see 8.4.2.32) and is shown in Table 8-5. Additional information on the interpretation of the contents of this field appears in 5.1.1.4.

Table 8-5—TID subfield

Access policy	Usage	Allowed values in bits 0–3 (TID subfield)
EDCA	UP for either TC or TS, regardless of whether admission control is required	0–7
HCCA	TSID	8–15
HEMM	TSID, regardless of the access mechanism used	8–15

For QoS Data+CF-Poll, the TID subfield in the QoS Control field indicates the TID of the data. For all QoS (+)CF-Poll frames of subtype Null, the TID subfield in the QoS Control field indicates the TID for which the poll is intended. The requirement to respond to that TID is nonbinding, and a STA may respond with any frame. For STAs where dot11OCBActivated is true, traffic streams are not used and the TID always corresponds to a TC.

8.2.4.5.3 EOSP (end of service period) subfield

The EOSP subfield is 1 bit in length and is used by the HC to indicate the end of the current service period (SP). The HC sets the EOSP subfield to 1 in its transmission and retransmissions of the SP's final frame to end a scheduled/unscheduled SP and sets it to 0 otherwise.

The mesh STA uses the EOSP subfield to indicate the end of the current mesh peer service period (MPSP) in which it operates as the owner. The mesh STA sets the EOSP subfield to 1 in its transmission and retransmissions of the MPSP's final frame to end an MPSP, and sets it to 0 otherwise. See 13.14.9.4 for details.

8.2.4.5.4 Ack Policy subfield

The Ack Policy subfield is 2 bits in length and identifies the acknowledgment policy that is followed upon the delivery of the MPDU. The interpretation of these 2 bits is given in Table 8-6.

Table 8-6—Ack Policy subfield in QoS Control field of QoS data frames

Bits in QoS Control field		Meaning
Bit 5	Bit 6	
0	0	Normal Ack or Implicit Block Ack Request. In a frame that is a non-A-MPDU frame: The addressed recipient returns an ACK or QoS +CF-Ack frame after a short interframe space (SIFS) period, according to the procedures defined in 9.3.2.8 and 9.19.3.5. This is the only permissible value for the Ack Policy subfield for individually addressed QoS Null (no data) frames. In a frame that is part of an A-MPDU: The addressed recipient returns a BlockAck MPDU, either individually or as part of an A-MPDU starting a SIFS after the PPDU carrying the frame, according to the procedures defined in 9.3.2.9, 9.21.7.5, 9.21.8.3, 9.25.3, 9.25.4, and 9.29.3.
1	0	No Ack The addressed recipient takes no action upon receipt of the frame. More details are provided in 9.22. The Ack Policy subfield is set to this value in all individually addressed frames in which the sender does not require acknowledgment. This combination is also used for group addressed frames that use the QoS frame format. This combination is not used for QoS data frames with a TID for which a Block Ack agreement exists. This is the only permissible value for the Ack Policy subfield for group addressed QoS Null (no data) frames.

Table 8-6—Ack Policy subfield in QoS Control field of QoS data frames *(continued)*

Bits in QoS Control field		Meaning
Bit 5	Bit 6	
0	1	No explicit acknowledgment or PSMP Ack. When bit 6 of the Frame Control field (see 8.2.4.1.3) is set to 1: There may be a response frame to the frame that is received, but it is neither the ACK nor any data frame of subtype +CF-Ack. For QoS CF-Poll and QoS CF-Ack+CF-Poll data frames, this is the only permissible value for the Ack Policy subfield. When bit 6 of the Frame Control field (see 8.2.4.1.3) is set to 0: The acknowledgment for a frame indicating PSMP Ack when it appears in a PSMP downlink transmission time (PSMP-DTT) is to be received in a later PSMP uplink transmission time (PSMP-UTT). The acknowledgment for a frame indicating PSMP Ack when it appears in a PSMP-UTT is to be received in a later PSMP-DTT. NOTE—Bit 6 of the Frame Control field (see 8.2.4.1.3) indicates the absence of a data payload. When equal to 1, the QoS data frame contains no payload, and any response is generated in response to a QoS CF-Poll or QoS CF-Ack+CF-Poll frame, but does not signify an acknowledgment of data. When set to 0, the QoS data frame contains a payload, which is acknowledged as described in 9.26.1.7.
1	1	Block Ack The addressed recipient takes no action upon the receipt of the frame except for recording the state. The recipient can expect a BlockAckReq frame in the future to which it responds using the procedure described in 9.21.

An MSDU is sent using an acknowledgment policy of Normal Ack, Implicit Block Ack Request, PSMP Ack or Block Ack if the service class parameter in the MA-UNITDATA.request primitive is QoSAck and of No Ack if the service class parameter in the MA-UNITDATA.request primitive is equal to QoSNoAck.

8.2.4.5.5 TXOP Limit subfield

The TXOP Limit subfield is an 8-bit field that is present in QoS data frames of subtypes that include CF-Poll and specifies the time limit on a TXOP granted by a QoS (+)CF-Poll frame from an HC in a BSS. In QoS data frames with subtypes that include CF-Poll, the addressed STA is granted a TXOP that begins a SIFS period after this frame and lasts no longer than the number of 32 μs periods specified by the TXOP limit value. The range of time values is 32 μs to 8160 μs. A TXOP limit value of 0 implies that one MPDU or one QoS Null frame is to be transmitted immediately following the QoS (+)CF-Poll frame.

8.2.4.5.6 Queue Size subfield

The Queue Size subfield is an 8-bit field that indicates the amount of buffered traffic for a given TC or TS at the STA sending this frame. The Queue Size subfield is present in QoS data frames sent by non-AP STAs with bit 4 of the QoS Control field equal to 1. The AP may use information contained in the Queue Size subfield to determine the TXOP duration assigned to the STA.

The queue size value is the total size, rounded up to the nearest multiple of 256 octets and expressed in units of 256 octets, of all MSDUs and A-MSDUs buffered at the STA (excluding the MSDU or A-MSDU of the present QoS data frame) in the delivery queue used for MSDUs and A-MSDUs with TID values equal to the value in the TID subfield of this QoS Control field. A queue size value of 0 is used solely to indicate the absence of any buffered traffic in the queue used for the specified TID. A queue size value of 254 is used for all sizes greater than 64 768 octets. A queue size value of 255 is used to indicate an unspecified or unknown size. If a QoS data frame is fragmented, the queue size value may remain constant in all fragments even if the amount of queued traffic changes as successive fragments are transmitted.

8.2.4.5.7 TXOP Duration Requested subfield

The TXOP Duration Requested subfield is present in QoS data frames sent by STAs associated in a BSS with bit 4 of the QoS Control field equal to 0. The TXOP Duration Requested subfield is an 8-bit field that indicates the duration, in units of 32 µs, that the sending STA determines it needs for its next TXOP for the specified TID. A value of 0 indicates that no TXOP is requested for the specified TID in the current SP. A nonzero value represents a requested TXOP duration in the range 32 µs to 8 160 µs in increments of 32 µs. See 9.19.3.5.2.

8.2.4.5.8 AP PS Buffer State subfield

The AP PS Buffer State subfield, defined in Figure 8-4, is an 8-bit field that indicates the PS buffer state at the AP for a STA. The AP PS Buffer State subfield is further subdivided into three subfields: Buffer State Indicated, Highest-Priority Buffered AC, and AP Buffered Load.

	B8	B9	B10 B11	B12 B15
	Reserved	Buffer State Indicated	Highest Priority Buffered AC	QoS AP Buffered Load
Bits:	1	1	2	4

Figure 8-4—QoS AP PS Buffer State subfield

The Buffered State Indicated subfield is 1 bit in length and is used to indicate whether the AP PS buffer state is specified. A value of 1 indicates that the AP PS buffer state is specified.

The Highest-Priority Buffered AC subfield is 2 bits in length and is used to indicate the AC of the highest priority traffic remaining that is buffered at the AP, excluding the MSDU or A-MSDU of the present frame.

The AP Buffered Load subfield is 4 bits in length and is used to indicate the total buffer size, rounded up to the nearest multiple of 4096 octets and expressed in units of 4096 octets, of all MSDUs and A-MSDUs buffered at the QoS AP (excluding the MSDU or A-MSDU of the present QoS data frame). An AP Buffered Load field value of 15 indicates that the buffer size is greater than 57 344 octets. An AP Buffered Load subfield value of 0 is used solely to indicate the absence of any buffered traffic for the indicated highest priority buffered AC when the Buffer State Indicated bit is 1.

When the Buffered State Indicated subfield is equal to 0, the Highest-Priority Buffered AC subfield and the AP Buffered Load subfield are reserved.

8.2.4.5.9 A-MSDU Present subfield

The A-MSDU Present subfield is 1 bit in length and indicates the presence of an A-MSDU. The A-MSDU Present subfield is set to 1 to indicate that the Frame Body field contains an entire A-MSDU as defined in 8.3.2.2. The A-MSDU Present subfield is set to 0 to indicate that the Frame Body field contains an MSDU or fragment thereof as defined in 8.3.2.1.

8.2.4.5.10 Mesh Control Present subfield

The Mesh Control Present subfield is 1 bit in length, and indicates the presence of a Mesh Control field in the frame body. When the Mesh Control Present subfield is 1, the Frame Body field contains a Mesh Control field as defined in 8.2.4.7.3. The mesh STA sets the Mesh Control Present subfield to 1 in the Mesh Data frame containing an unfragmented MSDU, an A-MSDU, or the first fragment of an MSDU.

8.2.4.5.11 Mesh Power Save Level subfield

The Mesh Power Save Level subfield is 1 bit in length and indicates whether the mesh STA's peer-specific mesh power mode will be deep sleep mode or light sleep mode after the successful completion of the frame exchange sequence.

When the Power Management field in the Frame Control field in the frame is 1, the following applies:

In individually addressed Mesh Data frames, a value of 0 indicates that the mesh STA's peer-specific mesh power mode for the recipient mesh STA will be light sleep mode (see 13.14.8.4). In individually addressed Mesh Data frames, a value of 1 indicates that the mesh STA's peer-specific mesh power mode for the recipient mesh STA will be deep sleep mode (see 13.14.8.5).

In group addressed Mesh Data frames, a value of 0 indicates that none of the peer-specific mesh power modes of the mesh STA will be deep sleep mode. In group addressed Mesh Data frames, a value of 1 indicates that at least one of the peer-specific mesh power modes of the mesh STA is deep sleep mode.

The Mesh Power Save Level subfield is reserved if the Power Management field in the Frame Control field is 0.

8.2.4.5.12 Receiver Service Period Initiated (RSPI) subfield

The Receiver Service Period Initiated (RSPI) subfield is 1 bit in length. The subfield is set to 0 to indicate that the mesh peer service period, of which the receiver of this frame is the owner, is not initiated. The subfield is set to 1 to indicate that the mesh peer service period, of which the receiver of this frame is the owner, is initiated. The use of the RSPI subfield is described in 13.14.9.2. The RSPI subfield is reserved in group addressed frames.

8.2.4.6 HT Control field

The HT Control field is always present in a Control Wrapper frame and is present in QoS Data and management frames as determined by the Order bit of the Frame Control field as defined in 8.2.4.1.10.

NOTE—The only Control frame subtype for which HT Control field is present is the Control Wrapper frame. A control frame that is described as +HTC (e.g., RTS+HTC, CTS+HTC, BlockAck+HTC or BlockAckReq+HTC) implies the use of the Control Wrapper frame to carry that control frame.

The format of the 4-octet HT Control field is shown in Figure 8-5.

	B0 B15	B16 B17	B18 B19	B20 B21	B22 B23	B24	B25 B29	B30	B31
	Link Adaptation Control	Calibration Position	Calibration Sequence	Reserved	CSI/ Steering	NDP Announce-ment	Reserved	AC Constraint	RDG/ More PPDU
Bits:	16	2	2	2	2	1	5	1	1

Figure 8-5—HT Control field

The format of the Link Adaptation Control subfield of the HT Control field is defined in Figure 8-6.

	B0	B1	B2 B5	B6 B8	B9 B15
	Reserved	TRQ	MAI	MFSI	MFB/ASELC
Bits:	1	1	4	3	7

Figure 8-6—Link Adaptation Control subfield

The subfields of the Link Adaptation Control subfield are defined in Table 8-7.

Table 8-7—Subfields of Link Adaptation Control subfield

Subfield	Meaning	Definition
TRQ	Training request	Set to 1 to request the responder to transmit a sounding PPDU. Set to 0 to indicate that the responder is not requested to transmit a sounding PPDU. See 9.29.2 and 9.31.2.
MAI	MCS request (MRQ) or ASEL indication	Set to 14 (indicating ASELI) to indicate that the MFB/ASELC subfield is interpreted as ASELC. Otherwise, the MAI subfield is interpreted as shown in Figure 8-7, and the MFB/ASELC subfield is interpreted as MCS feedback (MFB).
MFSI	MCS feedback sequence identifier	Set to the received value of MSI contained in the frame to which the MFB information refers. Set to 7 for unsolicited MFB.
MFB/ASELC	MCS feedback and antenna selection command/data	When the MAI subfield is equal to the value ASELI, this subfield is interpreted as defined in Figure 8-8 and Table 8-9. Otherwise, this subfield contains recommended MFB. A value of 127 indicates that no feedback is present.

The structure of the MAI subfield of the Link Adaptation Control subfield is defined in Figure 8-7. The subfields of the MAI subfield are defined in Table 8-8.

	B0	B1 B3
	MRQ	MSI
Bits:	1	3

Figure 8-7—MAI subfield

Table 8-8—Subfields of the MAI subfield

Subfield	Meaning	Definition
MRQ	MCS request	Set to 1 to indicate that MFB is requested. Set to 0 to indicate that no MFB is requested.
MSI	MRQ sequence identifier	When the MRQ subfield is equal to 1, the MSI subfield contains a sequence number in the range 0 to 6 that identifies the specific request.When the MRQ subfield is equal to 0, the MSI subfield is reserved.

The ASELC subfield of the Link Adaptation Control subfield contains the ASEL Command and ASEL Data subfields, as shown in Figure 8-8. The encoding of these subfields is shown in Table 8-9.

	B0 B2	B3 B6
	ASEL Command	ASEL Data
Bits:	3	4

Figure 8-8—ASELC subfield

Table 8-9—ASEL Command and ASEL Data subfields

ASEL Command	Interpretation of ASEL Command	ASEL Data
0	Transmit Antenna Selection Sounding Indication (TXASSI)	Number of remaining sounding PPDUs to be transmitted 0 to 15 See NOTE
1	Transmit Antenna Selection Sounding Request (TXASSR) or Transmit ASEL Sounding Resumption	0 when the command is Transmit ASEL Sounding Request A number in the range of values of 1 to 15, the number being the number of the first sounding PPDU to be transmitted when the command is Transmit ASEL Sounding Resumption, where 0 corresponds to the first sounding PPDU in the original ASEL training sequence
2	Receive Antenna Selection Sounding Indication (RXASSI)	Number of remaining sounding PPDUs to be received 0 to 15 See NOTE
3	Receive Antenna Selection Sounding Request (RXASSR)	Number of sounding PPDUs required 0 to 15
4	Sounding Label	Sequence number of the sounding PPDU corresponding to a channel state information (CSI) frame in ASEL feedback 0 to 15
5	No Feedback Due to ASEL Training Failure or Stale Feedback	A number in the range of values of 0 to 15, the number being the number of the first sounding PPDU that was not received properly, where 0 corresponds to the first sounding PPDU in the ASEL training sequence, or 0 if no sounding PPDUs were received properly, or 0 if this is a request for a full retraining sequence
6	Transmit Antenna Selection Sounding Indication requesting feedback of explicit CSI (TXASSI-CSI)	Number of remaining sounding PPDUs to be transmitted 0 to 15 See NOTE
7	Reserved	Reserved

NOTE—If the HT Control field is carried in a sounding PPDU, then the value of the ASEL Data field contains the remaining number of sounding frames following the current one. If null data packet (NDP) sounding frame is used, then the value in the ASEL Data field contains the number of NDPs following a non-NDP+HTC. The NDP Announcement subfield in the HT Control field is set to 1 to indicate NDP sounding.

The Calibration Position and Calibration Sequence subfields of the HT Control field are defined in Table 8-10.

The Calibration Sequence subfield identifies an instance of the calibration procedure. The subfield is included in each frame within a calibration procedure, and its value is unchanged for frames within the same calibration procedure.

Table 8-10—Calibration control subfields

Subfield	Meaning	Definition
Calibration Position	Position in calibration sounding exchange sequence	Set to 0 indicates this is not a calibration frame. Set to 1 indicates calibration start. Set to 2 indicates sounding response. Set to 3 indicates sounding complete.
Calibration Sequence	Calibration sequence identifier	The field is included in each frame within the calibration procedure and its value is unchanged for the frame exchanges during one calibration procedure. See 9.29.2.4.3.

The CSI/Steering subfield of the HT Control field indicates the type of feedback, as shown in Table 8-11.

Table 8-11—CSI/Steering subfield values

Value	Definition
0	No feedback required
1	CSI
2	Noncompressed beamforming
3	Compressed beamforming

The NDP Announcement subfield of the HT Control field indicates that an NDP will be transmitted after the frame (according to the rules described in 9.31). It is set to 1 to indicate that an NDP will follow; otherwise, it is set to 0.

The AC Constraint subfield of the HT Control field indicates whether the mapped AC of an RD data frame is constrained to a single AC, as defined in Table 8-12.

Table 8-12—AC Constraint subfield values

Value	Description
0	The response to a reverse direction grant (RDG) may contain data frames from any TID
1	The response to an RDG may contain data frames only from the same AC as the last data frame received from the RD initiator

The RDG/More PPDU subfield of the HT Control field is interpreted differently depending on whether it is transmitted by an RD initiator or an RD responder, as defined in Table 8-13.

Table 8-13—RDG/More PPDU subfield values

Value	Role of transmitting STA	Interpretation of value
0	RD initiator	No reverse grant
	RD responder	The PPDU carrying the frame is the last transmission by the RD responder
1	RD initiator	An RDG is present, as defined by the Duration/ID field
	RD responder	The PPDU carrying the frame is followed by another PPDU

8.2.4.7 Frame Body field

8.2.4.7.1 General

The Frame Body is a variable-length field that contains information specific to individual frame types and subtypes. The minimum length of the frame body is 0 octets. The maximum length of the frame body is defined by the maximum length MSDU plus the length of Mesh Control field as defined in 8.2.4.7.3, if present, plus any overhead for encryption as defined in Clause 11, or by the maximum length A-MSDU plus any overhead for encryption as defined in Clause 11.

8.2.4.7.2 Overhead for encryption

The overhead for encryption is described in Clause 11. When the Mesh Control field is present in the frame body, the Mesh Control field is encrypted as a part of data.

8.2.4.7.3 Mesh Control field

The Mesh Control field is present in the unfragmented Mesh Data frame, in the first fragment of the Mesh Data frame, and in the management frame of subtype Action, Category Multihop Action (Multihop Action frame) transmitted by a mesh STA.

In Mesh Data frames, when the Mesh Control Present subfield in the QoS Control field is 1, the Mesh Control field is prepended to the MSDU and located as follows:

- When the frame body contains an MSDU (or a fragment thereof) and the frame is not encrypted, the Mesh Control field is located in the first octets of the frame body.
- When the frame body contains an MSDU (or a fragment thereof) and the frame is encrypted, the Mesh Control field is located in the first octets of the encrypted data portion.
- When the frame body contains an A-MSDU, the Mesh Control field is located in the Aggregate MSDU subframe header as shown in Figure 8-33.

In the Multihop Action frame, the Mesh Control field is present as specified in 8.5.18.

The Mesh Control field is of variable length (6, 12, or 18 octets). The structure of the Mesh Control field is defined in Figure 8-9.

	Mesh Flags	Mesh TTL	Mesh Sequence Number	Mesh Address Extension
Octets:	1	1	4	0, 6, or 12

Figure 8-9—Mesh Control field

The Mesh Flags subfield is 1 octet in length and contains the Address Extension Mode subfield. The structure of the Mesh Flags subfield is shown in Figure 8-10.

	B0 B1	B2 B7
	Address Extension Mode	Reserved
Bits	Bits: 2	6

Figure 8-10—Mesh Flags subfield

The Address Extension Mode subfield indicates the contents of the Mesh Address Extension subfield. Table 8-14 defines valid values for the Address Extension Mode and describes the corresponding contents of the Mesh Address Extension subfield. If the Address Extension Mode is 0, the Mesh Address Extension subfield is not present. For values 1 and 2, the Mesh Address Extension subfield is present following the Mesh Sequence Number subfield.

Table 8-14—Valid values for the Address Extension Mode

Address Extension Mode value	Address Extension Mode description	Mesh Address Extension subfield length (octets)	Applicable frame types
0	No Mesh Address Extension subfield	0	Data, Management (Multihop Action, group addressed)
1	Mesh Address Extension subfield contains Address 4	6	Management (Multihop Action, individually addressed), Data (proxied, group addressed)
2	Mesh Address Extension subfield contains Address 5 and Address 6	12	Data (proxied, individually addressed)
3	Reserved	—	—

The Mesh TTL subfield is 1 octet in length and contains an unsigned integer corresponding to the remaining number of hops the MSDU/MMPDU is forwarded. How the Mesh TTL is used in both individually and group addressed frames is described in 9.32.4 and 9.32.5.

The Mesh Sequence Number subfield is 4 octets in length and contains an unsigned integer sequence number counter value. Source mesh STAs assign mesh sequence numbers from a single modulo-2^{32} counter, starting at 0 and incrementing by 1 for each MSDU or MMPDU that is transmitted with a Mesh Control field. Usage of the Mesh Sequence Number is described in 9.32.7.

NOTE—It is believed that a 32-bit sequence number is sufficient as the rollover would occur after a period of 5 days assuming a source continuously transmitting at a rate of 10^4 frames per second.

The Mesh Address Extension subfield, shown in Figure 8-11, is 6 or 12 octets in length and is present only when the Address Extension Mode subfield of the Mesh Flags subfield is a nonzero nonreserved value. The Mesh Address Extension subfield provides additional address fields for mesh address extension as defined in Table 8-14. The interpretation of the extended Address fields is described in 9.32.3.

	Address 4	Address 5	Address 6
Octets:	6	6	6

Figure 8-11—Mesh Address Extension subfield

The Address 4 subfield is present when the Address Extension Mode subfield in the Mesh Flags subfield is 1 . It carries a fourth address that is not included as a part of the MAC header for these frames.

The Address 5 subfield and Address 6 subfield are present when the Address Extension Mode subfield in the Mesh Flags subfield is 2. It carries the addresses of source and destination end station of the end-to-end IEEE 802 communication in cases where either (or both) of the end stations are not mesh STAs at the beginning or end of a single mesh path. (See Figure 9-42.)

NOTE—This is useful, for example, when the end stations of IEEE 802 communication are nonmesh, external STAs that communicate over a mesh BSS via proxy mesh gates.

Details on the usage of these optional address fields are given in 9.32.3.

8.2.4.8 FCS field

The FCS field is a 32-bit field containing a 32-bit CRC. The FCS is calculated over all the fields of the MAC header and the Frame Body field. These are referred to as the *calculation fields*.

The FCS is calculated using the following standard generator polynomial of degree 32:

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x + 1$$

The FCS is the ones complement of the sum (modulo 2) of the following:

a) The remainder of $x^{k} \times (x^{31} + x^{30} + x^{29} + \ldots + x^{2} + x + 1)$ divided (modulo 2) by $G(x)$, where k is the number of bits in the calculation fields, and

b) The remainder after multiplication of the contents (treated as a polynomial) of the calculation fields by x^{32} and then division by $G(x)$.

The FCS field is transmitted commencing with the coefficient of the highest-order term.

As a typical implementation, at the transmitter, the initial remainder of the division is preset to all ones and is then modified by division of the calculation fields by the generator polynomial $G(x)$. The ones complement of this remainder is transmitted, with the highest-order bit first, as the FCS field.

At the receiver, the initial remainder is preset to all ones and the serial incoming bits of the calculation fields and FCS, when divided by $G(x)$, results (in the absence of transmission errors) in a unique nonzero remainder value. The unique remainder value is the polynomial:

$$x^{31} + x^{30} + x^{26} + x^{25} + x^{24} + x^{18} + x^{15} + x^{14} + x^{12} + x^{11} + x^{10} + x^{8} + x^{6} + x^{5} + x^{4} + x^{3} + x + 1$$

8.2.5 Duration/ID field (QoS STA)

8.2.5.1 General

The value in the Duration/ID field in a frame transmitted by a QoS STA is defined in 8.2.5.2 through 8.2.5.8.

All times are calculated in microseconds. If a calculated duration includes a fractional microsecond, that value inserted in the Duration/ID field is rounded up to the next higher integer.

8.2.5.2 Setting for single and multiple protection under enhanced distributed channel access (EDCA)

Within a frame (excluding data frames containing QoS CF-Poll, PSMP frames, and frames that have the RDG/More PPDU subfield equal to 1) transmitted under EDCA by a STA that initiates a TXOP, there are two classes of duration settings: single protection and multiple protection. In single protection, the value of the Duration/ID field of the frame can set a NAV value at receiving STAs that protects up to the end of any following data, management, or response frame plus any additional overhead frames as described below. In multiple protection, the value of the Duration/ID field of the frame can set a NAV that protects up to the estimated end of a sequence of multiple frames. Frames that have the RDG/More PPDU subfield equal to 1 always use multiple protection. PSMP frames always use multiple protection. The STA selects between single and multiple protection when it transmits the first frame of a TXOP. All subsequent frames transmitted by the STA in the same TXOP use the same class of duration settings.

The Duration/ID field is determined as follows:

a) Single protection settings.
   1) For an RTS that is not part of a dual clear-to-send (CTS) exchange, the Duration/ID field is set to the estimated time, in microseconds, required to transmit the pending frame, plus one CTS frame, plus one ACK or BlockAck frame if required, plus any NDPs required, plus explicit feedback if required, plus applicable IFS durations.
   2) For all CTS frames sent by STAs as the first frame in the exchange under EDCA and with the receiver address (RA) matching the MAC address of the transmitting STA, the Duration/ID field is set to one of the following:
      i) If there is a response frame, the estimated time required to transmit the pending frame, plus one SIFS interval, plus the response frame (ACK or BlockAck), plus any NDPs required, plus explicit feedback if required, plus an additional SIFS interval
      ii) If there is no response frame, the time required to transmit the pending frame, plus one SIFS interval
   3) For a BlockAckReq frame, the Duration/ID field is set to the estimated time required to transmit one ACK or BlockAck frame, as applicable, plus one SIFS interval.
   4) For a BlockAck frame that is not sent in response to a BlockAckReq or an implicit Block Ack request, the Duration/ID field is set to the estimated time required to transmit an ACK frame plus a SIFS interval.
   5) For management frames, non-QoS data frames (i.e., with bit 7 of the Frame Control field equal to 0), and individually addressed data frames with the Ack Policy subfield equal to Normal Ack only, the Duration/ID field is set to one of the following:
      i) If the frame is the final fragment of the TXOP, the estimated time required for the transmission of one ACK frame (including appropriate IFS values)
      ii) Otherwise, the estimated time required for the transmission of one ACK frame, plus the time required for the transmission of the following MPDU and its response if required, plus applicable IFS durations.

6) For individually addressed QoS data frames with the Ack Policy subfield equal to No Ack or Block Ack, for management frames of subtype Action No Ack, and for group addressed frames, the Duration/ID field is set to one of the following:

   i) If the frame is the final fragment of the TXOP, 0

   ii) Otherwise, the estimated time required for the transmission of the following frame and its response frame, if required (including appropriate IFS values)

b) Multiple protection settings. The Duration/ID field is set to a value D as follows:

   1) If $T_{TXOP} = 0$ and $T_{END\_NAV} = 0$, then $D = T_{SINGLE\text{-}MSDU} - T_{\text{PPDU}}$

   2) Else if $T_{TXOP} = 0$ and $T_{END\_NAV} > 0$, then $D = T_{END\text{-}NAV} - T_{PPDU}$

   3) Else if $T_{END\text{-}NAV} = 0$, then $\min(T_{PENDING}, T_{TXOP} - T_{PPDU}) \leq D \leq T_{TXOP} - T_{PPDU}$

   4) Else $T_{END-NAV} - T_{PPDU} \leq D \leq T_{TXOP-REMAINING} - T_{PPDU}$

   where

   $T_{SINGLE\text{-}MSDU}$ is the estimated time required for the transmission of the allowed frame exchange sequence defined in 8.4.2.31 (for a TXOP limit value of 0), including applicable IFS durations

   $T_{PENDING}$ is the estimated time required for the transmission of

   — Pending MPDUs of the same AC

   — Any associated immediate response frames

   — Any NDP transmissions and explicit feedback response frames

   — Applicable IFS durations

   — Any RDG

   T_{TXOP} is the value of dot11EDCATableTXOPLimit (dot11EDCAQAPTableTXOPLimit for the AP) for that AC

   $T_{TXOP\text{-}REMAINING}$ is T_{TXOP} less the time already used time within the TXOP

   $T_{END\text{-}NAV}$ is the remaining duration of any NAV set by the TXOP holder, or 0 if no NAV has been established

   T_{PPDU} is the time required for transmission of the current PPDU

8.2.5.3 Setting for QoS CF-Poll frames

Within a data frame containing QoS CF-Poll, the Duration/ID field value is set to one of the following:

a) One SIFS duration plus the TXOP limit, if the TXOP limit is nonzero, or

b) The time required for the transmission of one MPDU of nominal MSDU size and the associated ACK frame plus two SIFS intervals, if the TXOP limit is 0.

8.2.5.4 Setting for frames sent by a TXOP holder under HCCA

Within a frame sent by a TXOP holder under hybrid coordination function (HCF) controlled channel access (HCCA), to provide NAV protection for the entire controlled access phase (CAP), the Duration/ID field is set to one of the following values:

a) For an RTS frame

   1) If the pending frame is the final frame, the time required to transmit the pending frame, plus one CTS frame, plus one ACK frame if required, plus three SIFS intervals

   2) If the pending frame is not the final frame in the TXOP, the remaining duration of the TXOP

b) For a CTS frame

   1) If the pending frame is the sole frame in the TXOP, one of the following:

      i) If there is a response frame, the time required to transmit the pending frame, plus one SIFS interval, plus the response frame (ACK or BlockAck), plus an additional SIFS interval

      ii) If there is no response frame, the time required to transmit the pending frame, plus one SIFS interval

   2) If the pending frame is not the final frame in the TXOP, the remaining duration of the TXOP

c) Otherwise

   1) If the frame is a nonfinal frame in a TXOP with multiple frame exchanges, the remaining duration of the TXOP

   2) If the frame is the sole or final frame in the TXOP, the actual remaining time needed for this frame exchange sequence

8.2.5.5 Settings within a PSMP sequence

Within a PSMP frame, the Duration/ID field is set to a value that is no less than the time required to complete all PSMP-DTT and PSMP-UTT periods described in the frame.

Within the PSMP-DTT and PSMP-UTT of a PSMP sequence, the Duration/ID field is set to the Duration/ID value of the preceding PSMP frame minus the time between the end of the PSMP frame and the end of the PPDU carrying the frame.

NOTE—In other words, all frames transmitted within a PSMP sequence locate the same NAV endpoint.

8.2.5.6 Settings within a dual CTS sequence

Within a frame (“Frame1”) (excluding a second CTS (CTS2) transmission, as defined in 9.3.2.7) sent by a QoS STA that is not a TXOP holder in a PPDU that contains an immediate response or that is sent by an RD responder, the Duration/ID field is set to the Duration/ID value from the frame(s) (“Frames2”) that elicited the response or that carried the RDG minus the time interval between the end of the PPDU that carried Frame1 and the end of the PPDU that carries Frames2.

Within a frame (“Frame1”) (excluding a CTS2 transmission, as defined in 9.3.2.7) sent by a QoS STA that is a TXOP holder, the Duration/ID field is set according to the rules in the following subclauses:

— 8.2.5.2 rule b) for multiple protection if Frame1 is not a QoS+CF-Poll frame and the TXOP holder is not operating under HCCA or PSMP

— 8.2.5.3 if Frame1 is a QoS+CF-Poll frame and the TXOP holder is not operating under HCCA or PSMP

— 8.2.5.4 if the TXOP holder is operating under HCCA

— 8.2.5.5. if the TXOP holder is operating under PSMP

Within the CTS2 of a dual CTS exchange, defined in 9.3.2.7, the Duration/ID field is set to the value of the Duration/ID field of the RTS that initiated the exchange minus the time required to transmit the first clear-to-sent (CTS1), CTS2, and the applicable IFS intervals.

8.2.5.7 Setting for control response frames

This subclause describes how to set the Duration/ID field for CTS, ACK, and BlockAck control response frames transmitted by a QoS STA.

For a CTS frame that is not part of a dual CTS sequence transmitted in response to an RTS frame, the Duration/ID field is set to the value obtained from the Duration/ID field of the RTS frame that elicited the response minus the time, in microseconds, between the end of the PPDU carrying the RTS frame and the end of the PPDU carrying the CTS frame.

For an ACK frame, the Duration/ID field is set to the value obtained from the Duration/ID field of the frame that elicited the response minus the time, in microseconds between the end of the PPDU carrying the frame that elicited the response and the end of the PPDU carrying the ACK frame.

For a BlockAck frame transmitted in response to a BlockAckReq frame or transmitted in response to a frame containing an implicit Block Ack request, the Duration/ID field is set to the value obtained from the Duration/ID field of the frame that elicited the response minus the time, in microseconds between the end of the PPDU carrying the frame that elicited the response and the end of the PPDU carrying the BlockAck frame.

8.2.5.8 Setting for other response frames

For any frame transmitted by a STA that is not the TXOP holder and is not specified by 8.2.5.1 through 8.2.5.7, the Duration/ID field is set to the value obtained from the Duration/ID field of the frame that elicited the response minus the time, in microseconds, between the end of the PPDU carrying the frame that elicited the response and the end of the PPDU carrying the frame.

8.3 Format of individual frame types

8.3.1 Control frames

8.3.1.1 Format of control frames

In the following descriptions, “immediately previous” frame means a frame whose reception concluded within the SIFS interval preceding the start of the current frame.

The subfields within the Frame Control field of control frames are set as illustrated in Figure 8-12.

	B0 B1	B2 B3	B4 B7	B8	B9	B10	B11	B12	B13	B13	B15
	Protocol Version	Type (Control)	Subtype	To DS (0)	From DS (0)	More Frag (0)	Retry (0)	Power Man-age-ment	More Data (0)	Protected Frame (0)	Order (0)
Bits:	2	2	4	1	1	1	1	1	1	1	1

Figure 8-12—Frame Control field subfield values within control frames

8.3.1.2 RTS frame format

The frame format for the RTS frame is as defined in Figure 8-13.

Octets:	2	2	6	6	4
	Frame Control	Duration	RA	TA	FCS

MAC Header

Figure 8-13—RTS frame

The RA field of the RTS frame is the address of the STA, on the WM, that is the intended immediate recipient of the pending individually addressed data, management, or control frame.

The TA field is the address of the STA transmitting the RTS frame.

For all RTS frames sent by non-QoS STAs, the duration value is the time, in microseconds, required to transmit the pending data or management frame, plus one CTS frame, plus one ACK frame, plus three SIFS intervals. If the calculated duration includes a fractional microsecond, that value is rounded up to the next higher integer. For RTS frames sent by QoS STAs, see 8.2.5.

8.3.1.3 CTS frame format

The frame format for the CTS frame is as defined in Figure 8-14.

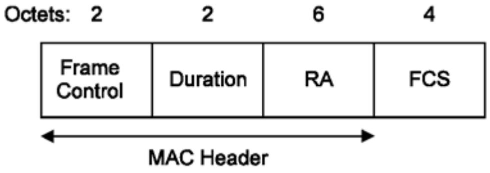


Figure 8-14—CTS frame

When the CTS frame follows an RTS frame, the RA field of the CTS frame is copied from the TA field of the immediately previous RTS frame to which the CTS is a response. When the CTS is the first frame in a frame exchange, the RA field is set to the MAC address of the transmitter.

For all CTS frames transmitted by a non-QoS STA in response to RTS frames, the duration value is the value obtained from the Duration field of the immediately previous RTS frame, minus the time, in microseconds, required to transmit the CTS frame and its SIFS interval. If the calculated duration includes a fractional microsecond, that value is rounded up to the next higher integer.

At a non-QoS STA, if the CTS is the first frame in the exchange and the pending data or management frame requires acknowledgment, the duration value is the time, in microseconds, required to transmit the pending data or management frame, plus two SIFS intervals plus one ACK frame. At a non-QoS STA, if the CTS is the first frame in the exchange and the pending data or management frame does not require acknowledgment, the duration value is the time, in microseconds, required to transmit the pending data or management frame, plus one SIFS interval. If the calculated duration includes a fractional microsecond, that value is rounded up to the next higher integer.

For other CTS transmissions by a QoS STA, the duration value is set as defined in 8.2.5.

8.3.1.4 ACK frame format

The frame format for the ACK frame is as defined in Figure 8-15.

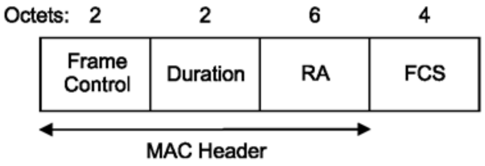


Figure 8-15—ACK frame

The RA field of the ACK frame is copied from the Address 2 field of the immediately previous individually addressed data, management, BlockAckReq, BlockAck, or PS-Poll frames.

For ACK frames sent by non-QoS STAs, if the More Fragments bit was equal to 0 in the Frame Control field of the immediately previous individually addressed data or management frame, the duration value is set to 0. In other ACK frames sent by non-QoS STAs, the duration value is the value obtained from the Duration/ID field of the immediately previous data, management, PS-Poll, BlockAckReq, or BlockAck frame minus the time, in microseconds, required to transmit the ACK frame and its SIFS interval. If the calculated duration includes a fractional microsecond, that value is rounded up to the next higher integer.

In all other ACK frames, the duration value is specified by 8.2.5.

8.3.1.5 PS-Poll frame format

The frame format for the PS-Poll frame is as defined in Figure 8-16.

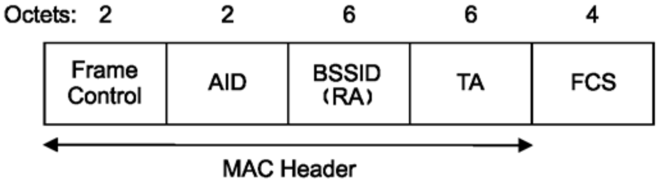


Figure 8-16—PS-Poll frame

The BSSID is the address of the STA contained in the AP. The TA field is the address of the STA transmitting the frame. The AID is the value assigned to the STA transmitting the frame by the AP in the association response frame that established that STA’s current association.

The AID value always has its two MSBs set to 1.

8.3.1.6 CF-End frame format

The frame format for the CF-End frame is as defined in Figure 8-17.

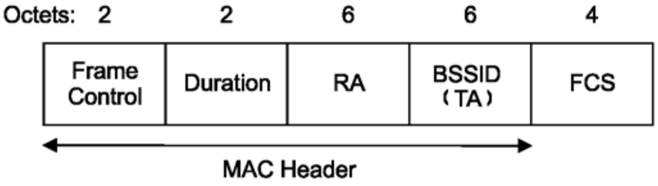


Figure 8-17—CF-End frame

The BSSID field is the address of the STA contained in the AP. The RA field is the broadcast group address.

The Duration field is set to 0.

8.3.1.7 CF-End+CF-Ack frame format

The frame format for the CF-End+CF-Ack frame is as defined in Figure 8-18.

Octets:	2	2	6	6	4
	Frame Control	Duration	RA	BSSID (TA)	FCS

MAC Header

Figure 8-18—CF-End+CF-Ack frame

The BSSID field is the address of the STA contained in the AP. The RA field is the broadcast group address.

The Duration field is set to 0.

8.3.1.8 BlockAckReq frame format

8.3.1.8.1 Overview

The frame format of the BlockAckReq frame is defined in Figure 8-19.

Octets:	2	2	6	6	2	Variable	4
	Frame Control	Duration/ID	RA	TA	BAR Control	BAR Information	FCS

MAC Header

Figure 8-19—BlockAckReq frame

The Duration/ID field value is set as defined in 8.2.5.

The RA field of the BlockAckReq frame is the address of the recipient STA.

The TA field is the address of the STA transmitting the BlockAckReq frame.

The BAR Control field is shown in Figure 8-20.

	B0	B1	B2	B3 B11	B12 B15
	BAR Ack Policy	Multi-TID	Compressed Bitmap	Reserved	TID_INFO
Bits:	1	1	1	9	4

Figure 8-20—BAR Control field

For BlockAckReq frames sent under Delayed and HT-Delayed agreements, the BAR Ack Policy subfield of the BAR Control field has the meaning shown in Table 8-15. For BlockAckReq frames sent under other types of agreement, the BAR Ack Policy subfield is reserved.

Table 8-15—BAR Ack Policy subfield

Value	Meaning
0	Normal Acknowledgment. The BAR Ack Policy subfield is set to this value when the sender requires immediate acknowledgment. The addressee returns an ACK. See 9.26.1.7.
1	No Acknowledgment. The addressee sends no immediate response upon receipt of the frame. The BAR Ack Policy subfield is set to this value when the sender does not require immediate acknowledgment. The value 1 is not used in a Basic BlockAckReq frame outside a PSMP sequence. The value 1 is not used in an Multi-TID BlockAckReq frame.

The values of the Multi-TID and Compressed Bitmap subfields determine which of three possible BlockAckReq frame variants is represented, as indicated in Table 8-16.

Table 8-16—BlockAckReq frame variant encoding

Multi-TID subfield value	Compressed Bitmap subfield value	BlockAckReq frame variant
0	0	Basic BlockAckReq
0	1	Compressed BlockAckReq
1	0	Reserved
1	1	Multi-TID BlockAckReq

The meaning of the TID_INFO subfield of the BAR Control field depends on the BlockAckReq frame variant type. The meaning of this subfield is explained within the subclause for each of the BlockAckReq frame variants.

The meaning of the BAR Information field of the BlockAckReq frame depends on the BlockAckReq frame variant type. The meaning of this field is explained within the subclause for each of the BlockAckReq frame variants.

NOTE—Reference to "a BlockAckReq" frame without any other qualification from other subclauses applies to any of the variants, unless specific exclusions are called out.

8.3.1.8.2 Basic BlockAckReq variant

The TID_INFO subfield of the BAR Control field of the Basic BlockAckReq frame contains the TID for which a Basic BlockAck frame is requested.

The BAR Information field of the Basic BlockAckReq frame contains the Block Ack Starting Sequence Control subfield, as shown in Figure 8-21. The Starting Sequence Number subfield of the Block Ack Starting Sequence Control subfield contains the sequence number of the first MSDU for which this Basic BlockAckReq frame is sent. The Fragment Number subfield is set to 0.

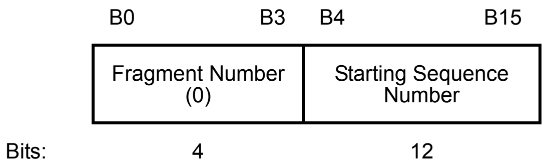


Figure 8-21—Block Ack Starting Sequence Control field

8.3.1.8.3 Compressed BlockAckReq variant

The TID_INFO subfield of the BAR Control field of the Compressed BlockAckReq frame contains the TID for which a BlockAck frame is requested.

The BAR Information field of the Compressed BlockAckReq frame contains the Block Ack Starting Sequence Control subfield, as shown in Figure 8-21. The Starting Sequence Number subfield of the Block Ack Starting Sequence Control subfield contains the sequence number of the first MSDU or A-MSDU for which this BlockAckReq frame is sent. The Fragment Number subfield of the Block Ack Starting Sequence Control subfield is set to 0.

8.3.1.8.4 Multi-TID BlockAckReq variant

The TID_INFO subfield of the BAR Control field of the Multi-TID BlockAckReq frame determines the number of TIDs present in the Multi-TID BlockAckReq frame as given by TID_INFO + 1, e.g., a value of 2 in the TID_INFO subfield means that three TID values are present in the Multi-TID BlockAckReq frame's BAR Information field.

The BAR Information field of the Multi-TID BlockAckReq frame comprises multiple sets of Per TID Info subfields and Block Ack Starting Sequence Control subfields, as shown in Figure 8-22. The Per TID Info subfield is shown in Figure 8-23. The Block Ack Starting Sequence Control subfield is shown in Figure 8-21. The Starting Sequence Number subfield of the Block Ack Starting Sequence Control subfield contains the sequence number of the first MSDU or A-MSDU for which this BlockAckReq frame is sent. The Fragment Number subfield of the Block Ack Starting Sequence Control subfield is set to 0.

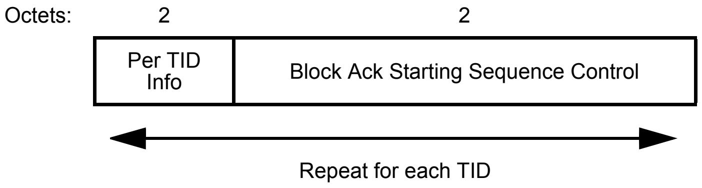


Figure 8-22—BAR Information field (Multi-TID BlockAckReq)

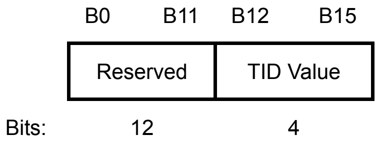


Figure 8-23—Per TID Info subfield

8.3.1.9 BlockAck frame format

8.3.1.9.1 Overview

The frame format of the BlockAck frame is defined in Figure 8-24.

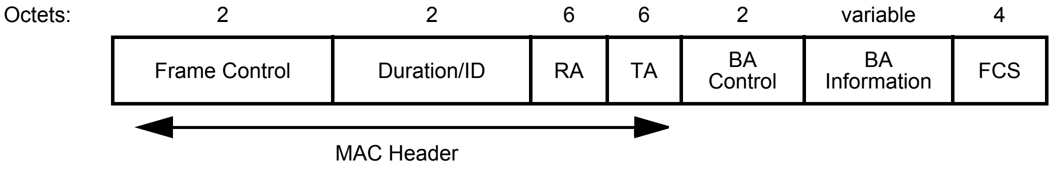


Figure 8-24—BlockAck frame

The Duration/ID field value is set as defined in 8.2.5.

The RA field of the BlockAck frame is the address of the recipient STA that requested the Block Ack.

The TA field is the address of the STA transmitting the BlockAck frame.

The BA Control field is defined in Figure 8-25.

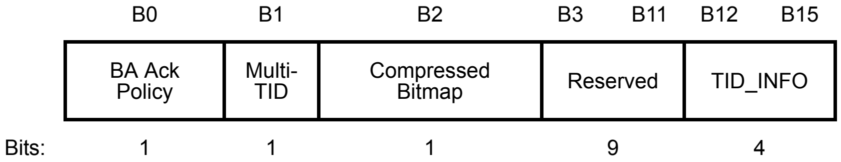


Figure 8-25—BA Control field

For BlockAck frames sent under Delayed and HT-Delayed agreements, the BA Ack Policy subfield of the BA Control field has the meaning shown in Table 8-17. For BlockAck frames sent under other types of agreement, the BA Ack Policy subfield is reserved.

Table 8-17—BA Ack Policy subfield

Value	Meaning
0	Normal Acknowledgment. The BA Ack Policy subfield is set to this value when the sender requires immediate acknowledgment. The addressee returns an ACK. The value 0 is not used for data sent under HT-delayed BlockAck during a PSMP sequence.
1	No Acknowledgment. The addressee sends no immediate response upon receipt of the frame. The BA Ack Policy is set to this value when the sender does not require immediate acknowledgment. The value 1 is not used in a Basic BlockAck frame outside a PSMP sequence. The value 1 is not used in an Multi-TID BlockAck frame.

The values of the Multi-TID and Compressed Bitmap subfields of the BA Control field determine which of three possible BlockAck frame variants is represented, as indicated in the Table 8-18.

Table 8-18—BlockAck frame variant encoding

Multi-TID subfield value	Compressed Bitmap subfield value	BlockAck frame variant
0	0	Basic BlockAck
0	1	Compressed BlockAck
1	0	Reserved
1	1	Multi-TID BlockAck

NOTE—Reference to “a BlockAck” frame without any other qualification from other subclauses applies to any of the variants, unless specific exclusions are called out.

The meaning of the TID_INFO subfield of the BA Control field depends on the BlockAck frame variant type. The meaning of this subfield is explained within the subclause for each of the BlockAck frame variants.

The meaning of the BA Information field depends on the BlockAck frame variant type. The meaning of this field is explained within the subclause for each of the BlockAck frame variants.

8.3.1.9.2 Basic BlockAck variant

The TID_INFO subfield of the BA Control field of the Basic BlockAck frame contains the TID for which this BlockAck frame is sent.

The BA Information field of the Basic BlockAck frame comprises the Block Ack Starting Sequence Control subfield and the Block Ack Bitmap subfield, as shown in Figure 8-26. The format of the Block Ack Starting Sequence Control subfield is shown in Figure 8-21. The Starting Sequence Number subfield of the Block Ack Starting Sequence Control subfield contains the sequence number of the first MSDU for which this Basic BlockAck frame is sent and is set to the same value as in the immediately previously received Basic BlockAckReq frame.

	Block Ack Starting Sequence Control	Block Ack Bitmap
Octets:	2	128

Figure 8-26—BA Information field (BlockAck)

The Block Ack Bitmap subfield is 128 octets in length and is used to indicate the received status of up to 64 MSDUs. Bit position *n* of the Block Ack bitmap, if equal to 1, acknowledges receipt of an MPDU with an MPDU sequence control value equal to (Block Ack Starting Sequence Control + *n*). Bit position *n* of the Block Ack bitmap, if equal to 0, indicates that an MPDU with MPDU sequence control value equal to (Block Ack Starting Sequence Control + *n*) has not been received. Each of the MPDU Sequence Control field and Block Ack Starting Sequence Control subfield values are treated as a 16-bit unsigned integer. For unused fragment numbers of an MSDU, the corresponding bits in the bitmap are set to 0.

8.3.1.9.3 Compressed BlockAck variant

The TID_INFO subfield of the BA Control field of the Compressed BlockAck frame contains the TID for which this BlockAck frame is sent.

The BA Information field of the Compressed BlockAck frame comprises the Block Ack Starting Sequence Control subfield and the Block Ack Bitmap subfield, as shown in Figure 8-27. The Starting Sequence Number subfield of the Block Ack Starting Sequence Control subfield contains the sequence number of the first MSDU or A-MSDU for which this BlockAck frame is sent. The value of this subfield is defined in 9.21.7.5. The Fragment Number subfield of the Block Ack Starting Sequence Control subfield is set to 0.

	Block Ack Starting Sequence Control	Block Ack Bitmap
Octets:	2	8

Figure 8-27—BA Information field (Compressed BlockAck)

The Block Ack Bitmap subfield of the BA Information field of the Compressed BlockAck frame is 8 octets in length and is used to indicate the received status of up to 64 MSDUs and A-MSDUs. Each bit that is equal to 1 in the compressed Block Ack bitmap acknowledges the successful reception of a single MSDU or A-MSDU in the order of sequence number, with the first bit of the Block Ack bitmap corresponding to the MSDU or A-MSDU with the sequence number that matches the value of the Starting Sequence Number subfield of the Block Ack Starting Sequence Control subfield.

8.3.1.9.4 Multi-TID BlockAck variant

The TID_INFO subfield of the BA Control field of the Multi-TID BlockAck frame contains the number of TIDs, less one, for which information is reported in the BA Information field. For example, a value of 2 in the TID_INFO subfield means that information for three TIDs is present.

The BA Information field of the Multi-TID BlockAck frame comprises 1 or more instances of the Per TID Info, Block Ack Starting Sequence Control, and the Block Ack Bitmap subfields, as shown in Figure 8-28. The Per TID Info subfield is shown in Figure 8-23, and the Block Ack Starting Sequence Control subfield is shown in Figure 8-21.

The Starting Sequence Number subfield of the Block Ack Starting Sequence Control subfield is the sequence number of the first MSDU or A-MSDU for which this BlockAck frame is sent. The value of this subfield is defined in 9.21.7.5. The Fragment Number subfield of the Block Ack Starting Sequence Control subfield is set to 0. The first instance of the Per TID Info, Block Ack Starting Sequence Control, and Block Ack Bitmap subfields that is transmitted corresponds to the lowest TID value, with subsequent instances ordered by increasing values of the Per TID Info subfield.

Octets:	2	2	8
	Per TID Info	Block Ack Starting Sequence Control	Block Ack Bitmap

Repeat for each TID

Figure 8-28—BA Information field (Multi-TID BlockAck)

The Block Ack Bitmap subfield of the BA Information field of the Multi-TID BlockAck frame contains an 8-octet Block Ack bitmap. Each bit that is equal to 1 in the Block Ack bitmap acknowledges the successful reception of a single MSDU or A-MSDU in the order of sequence number with the first bit of the Block Ack bitmap corresponding to the MSDU or A-MSDU with the sequence number that matches the value of the Starting Sequence Number subfield of the Block Ack Starting Sequence Control subfield.

8.3.1.10 Control Wrapper frame

The format of the Control Wrapper frame is shown in Figure 8-29.

	Frame Control	Duration/ ID	Address 1	Carried Frame Control	HT Control	Carried Frame	FCS
Octets:	2	2	6	2	4	variable	4

Figure 8-29—Control Wrapper frame

The Control Wrapper frame is used to carry any other control frame (i.e., excluding the Control Wrapper frame) together with a HT Control field.

The Frame Control field is defined in 8.3.1. The value for the subtype field is the value from Table 8-1 of 8.2.4.1.3 that corresponds to Control Wrapper frame.

The value for the Duration/ID field of the Control Wrapper frame is generated by following the rules for the Duration/ID field of the control frame that is being carried.

The value for the Address 1 field of the Control Wrapper frame is generated by following the rules for the Address 1 field of the control frame that is being carried.

The Carried Frame Control field contains the value of the Frame Control field of the carried control frame.

The HT Control field is defined in 8.2.4.6.

The Carried Frame field contains the fields that follow the Address 1 field of the control frame that is being carried, excluding the FCS field.

The FCS field is defined in 8.2.4.8.

8.3.2 Data frames

8.3.2.1 Data frame format

The format of a data frame is defined in Figure 8-30. The Frame Control, Duration/ID, Address 1, Address 2, Address 3, and Sequence Control fields are present in all data frame subtypes. The presence of the Address 4 field is determined by the setting of the To DS and From DS subfields of the Frame Control field (see below). The QoS Control field is present when the QoS subfield of the Subtype field is set to 1.

Octets: 2	2	6	6	6	2	6	2	4	0–7951	4
Frame Control	Duration/ ID	Address 1	Address 2	Address 3	Sequence Control	Address 4	QoS Control	HT Control	Frame Body	FCS

◄────────────────────────►
MAC Header

Figure 8-30—Data frame

NOTE—The maximum frame body size shown in Figure 8-30 is for CCMP encryption of a maximum-size A-MSDU (note that TKIP encryption is not allowed in this case and any Mesh Control fields are part of the A-MSDU subframes). The maximum frame body size if A-MSDUs are not used is 2338 octets for CCMP encryption of a maximum-size MSDU and 2342 octets for TKIP encryption of a maximum-size MSDU, including in both cases an 18-octet Mesh Control field. The frame body size might in all cases be greater if a vendor-specific cipher suite is used.

Data frames with a value of 1 in the QoS subfield of the Subtype field are collectively referred to as *QoS data frames*. Each of these data subtypes contains QoS in their names, and this frame format is distinguished by the presence of a QoS Control field in the MAC header.

A QoS STA always uses QoS data frames for data transmissions to other QoS STAs. A QoS STA uses frames with the QoS subfield of the Subtype field set to 0 for data transmissions to non-QoS STAs. A non-QoS STA always uses frames with the QoS subfield of the Subtype field set to 0 for data transmissions to other STAs. All STAs use frames with the QoS subfield of the Subtype field set to 0 for broadcast data frames unless a transmitting STA knows that all STAs in a BSS have QoS capability, in which case the transmitting STAs use QoS data frames. All STAs use frames with the QoS subfield of the Subtype field set to 0 for group addressed data frames unless it is known to the transmitter that all STAs in the BSS that are members of the multicast group have QoS capability, in which case STAs use QoS data frames.

The content of the address fields of data frames are dependent upon the values of the To DS and From DS fields in the Frame Control field and whether the Frame Body field contains either an MSDU (or fragment thereof) or an entire A-MSDU, as determined by the A-MSDU Present subfield of the QoS Control field (see 8.2.4.5.9). The content of the address fields is defined in Table 8-19. Where the content of a field is shown as not applicable (N/A), the field is omitted. Note that Address 1 always holds the receiver address of the intended receiver (or, in the case of group addressed frames, receivers), and that Address 2 always holds the address of the STA that is transmitting the frame.

Table 8-19—Address field contents

To DS	From DS	Address 1	Address 2	Address 3		Address 4	
				MSDU case	A-MSDU case	MSDU case	A-MSDU case
0	0	RA = DA	TA = SA	BSSID	BSSID	N/A	N/A
0	1	RA = DA	TA = BSSID	SA	BSSID	N/A	N/A
1	0	RA = BSSID	TA = SA	DA	BSSID	N/A	N/A
1	1	RA	TA	DA	BSSID	SA	BSSID

A STA uses the contents of the Address 1 field to perform address matching for receive decisions. A mesh STA also uses the address matching rules described in 9.32.4, when it receives an individually addressed frame. When a STA other than mesh STA (nonmesh STA) receives a frame with the Address 1 field equal to a group address, the STA also validates the BSSID to verify either that the group addressed frame originated from a STA in the BSS of which the receiving STA is a member, or that it contains the wildcard BSSID value, indicating a data frame sent outside the context of a BSS (dot11OCBActivated is true in the transmitting STA). When a mesh STA receives a frame with the Address 1 field equal to a group address, the mesh STA also validates the TA to ensure that the group addressed frame originated from one of its peer mesh STA. A mesh STA also uses the address matching rules described in 9.32.5.

A STA uses the contents of the Address 2 field to direct the acknowledgment if an acknowledgment is necessary.

The DA field contains the destination of the MSDU (or fragment thereof) or A-MSDU in the Frame Body field.

The SA field contains the address of the MAC entity that initiated the MSDU (or fragment thereof) or A-MSDU in the Frame Body field.

When a data frame carries an MSDU (or fragment thereof), the DA and SA values related to that MSDU are carried in the Address 1, Address 2, Address 3, and Address 4 fields (according to the setting of the To DS and From DS fields) as defined in Table 8-19.

When a data frame carries an A-MSDU, the DA and SA values related to each MSDU carried by the A-MSDU are carried within the A-MSDU. One or both of these fields may also be present in the Address 1 and Address 2 fields as indicated in Table 8-19.

NOTE—If a DA or SA value also appears in any of these address fields, the value is necessarily the same for all MSDUs within the A-MSDU because this is guaranteed by the To DS and From DS field settings.

The RA field is the individual address of the STA that is the immediate intended receiver of the frame or the group address of the STAs that are the immediate intended receivers of the frame.

The TA field is the address of the STA that is transmitting the frame.

The BSSID of the Data frame is determined as follows:

a) If the STA is contained within an AP or is associated with an AP, the BSSID is the address currently in use by the STA contained in the AP.
b) If the STA is a member of an IBSS, the BSSID is the BSSID of the IBSS.
c) If the STA is transmitting a data frame when dot11OCBActivated is true, the BSSID is the wildcard BSSID.
d) If the STA is a member of an MBSS, the BSSID is the address of the transmitter and is equal to the Data frame’s TA.

The Sequence Control field is defined in 8.2.4.4. The Sequence Control field for QoS (+)Null frames is ignored by the receiver upon reception.

The QoS Control field is defined in 8.2.4.5.

The HT Control field is defined in 8.2.4.6. The presence of the HT Control field is determined by the Order subfield of the Frame Control field, as specified in 8.2.4.1.10.

The frame body consists of either:

— The MSDU (or a fragment thereof), the Mesh Control field (present if the frame is transmitted by a mesh STA and the Mesh Control Present subfield of the QoS Control field is 1, otherwise absent), and a security header and trailer (present if the Protected Frame subfield in the Frame Control field is 1, otherwise absent)
— The A-MSDU and a security header and trailer (present if the Protected Frame subfield in the Frame Control field is 1, otherwise absent)

The presence of an A-MSDU in the frame body is indicated by setting the A-MSDU Present subfield of the QoS Control field to 1, as shown in Table 8-4.

For data frames of subtype Null (no data), CF-Ack (no data), CF-Poll (no data), and CF-Ack+CF-Poll (no data) and for the corresponding QoS data frame subtypes, the Frame Body field is null (i.e., has a length of 0 octets); these subtypes are used for MAC control purposes. For data frames of subtypes Data, Data+CF-Ack, Data+CF-Poll, and Data+CF-Ack+CF-Poll, the Frame Body field contains all of, or a fragment of, an MSDU after any encapsulation for security. For data frames of subtypes QoS Data, QoS Data+CF-Ack, QoS Data+CF-Poll, and QoS Data+CF-Ack+CF-Poll, the Frame Body field contains an MSDU (or fragment thereof) or A-MSDU after any encapsulation for security. For data frames of subtype QoS Data that are transmitted by a mesh STA, the Frame Body field also contains a Mesh Control field, as described in 8.2.4.7.3.

The maximum length of the Frame Body field can be determined from the maximum MSDU length plus the length of the Mesh Control field (if present) plus any overhead from encapsulation for encryption (i.e., it is always possible to send a maximum length MSDU, with any encapsulations provided by the MAC layer within a single data MPDU). When the frame body carries an A-MSDU, the size of the frame body field is limited by:

- The PHY's maximum PLCP service data unit (PSDU) length
- If A-MPDU aggregation is used, a maximum MPDU length of 4095 octets (see 8.6)

Within all data frames sent by STAs during the CFP under PCF, the Duration/ID field is set to 32 768. Within all data frames sent by the QoS STA, the Duration/ID field contains a duration value as defined in 8.2.5. Within all data frames sent during the CP by non-QoS STAs, the Duration/ID field is set according to the following rules:

- If the Address 1 field contains a group address, the duration value is set to 0.
- If the More Fragments bit is 0 in the Frame Control field of a frame and the Address 1 field contains an individual address, the duration value is set to the time, in microseconds, required to transmit one ACK frame, plus one SIFS interval.
- If the More Fragments bit is 1 in the Frame Control field of a frame and the Address 1 field contains an individual address, the duration value is set to the time, in microseconds, required to transmit the next fragment of this data frame, plus two ACK frames, plus three SIFS intervals.

The duration value calculation for the data frame is based on the rules in 9.7 that determine the data rate at which the control frames in the frame exchange sequence are transmitted. If the calculated duration includes a fractional microsecond, that value is rounded up to the next higher integer. All STAs process Duration/ID field values less than or equal to 32 767 from valid data frames (without regard for the RA, DA, and/or BSSID address values that might be present in these frames) to update their NAV settings as appropriate under the coordination function rules.

8.3.2.2 A-MSDU format

An A-MSDU is a sequence of A-MSDU subframes as shown in Figure 8-31. Each A-MSDU subframe consists of an A-MSDU subframe header followed by an MSDU and 0 to 3 octets of padding as shown in Figure 8-32. Each A-MSDU subframe (except the last) is padded so that its length is a multiple of 4 octets. The last A-MSDU subframe has no padding.

A-MSDU subframe 1	A-MSDU subframe 2	…	A-MSDU subframe n

Figure 8-31—A-MSDU structure

Octets:	6	6	2	0–2304	0–3
	DA	SA	Length	MSDU	Padding

A-MSDU subframe header

Figure 8-32—A-MSDU subframe structure

The A-MSDU subframe header contains three fields: DA, SA, and Length. The order of these fields and the bits within these fields are the same as the IEEE 802.3 frame format. The DA and SA fields of the A-MSDU subframe header contain the values passed in the MA-UNITDATA.request and MA-UNITDATA.indication primitives. The Length field contains the length in octets of the MSDU.

An A-MSDU contains only MSDUs whose DA and SA parameter values map to the same receiver address (RA) and transmitter address (TA) values, i.e., all the MSDUs are intended to be received by a single receiver, and necessarily they are all transmitted by the same transmitter. The rules for determining RA and TA are independent of whether the frame body carries an A-MSDU.

NOTE—It is possible to have different DA and SA parameter values in A-MSDU subframe headers of the same A-MSDU as long as they all map to the same Address 1 and Address 2 parameter values.

The MPDU containing the A-MSDU is carried in any of the following data frame subtypes: QoS Data, QoS Data + CF-Ack, QoS Data + CF-Poll, QoS Data + CF-Ack + CF-Poll. The A-MSDU structure is contained in the frame body of a single MPDU. If encrypted, the MPDU is encrypted as a single unit.

NOTE 1—The value of TID present in the QoS Control field of the MPDU carrying the A-MSDU indicates the TID for all MSDUs in the A-MSDU. Because this value of TID is common to all MSDUs in the A-MSDU, only MSDUs delivered to the MAC by an MA-UNITDATA.request primitive with an integer priority parameter that maps to the same TID can be aggregated together using A-MSDU.

NOTE 2—The maximum MPDU length that can be transported using A-MPDU aggregation is 4095 octets. An A-MSDU cannot be fragmented. Therefore, an A-MSDU of a length that exceeds 4065 octets (4095 minus the QoS data MPDU overhead) cannot be transported in an A-MPDU.

When Mesh Data frames are aggregated, the Aggregate MSDU subframe header includes Mesh DA, Mesh SA, Length, and Mesh Control. The A-MSDU subframe structure for Mesh Data is defined in Figure 8-33.

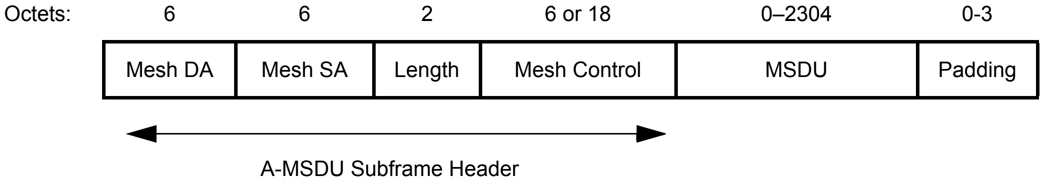


Figure 8-33—A-MSDU Subframe structure for Mesh Data

The Mesh DA and Mesh SA fields contain the addresses of the destination mesh STA and the source mesh STA, respectively, determined in 9.32.3.

The Length field contains the length in octets of the MSDU.

The format of the Mesh Control field is described in 8.2.4.7.3.

NOTE—It is possible to have different Mesh DA, Mesh SA, and Mesh Control in Subframe Headers of the same A-MSDU as long as they all map to the same Address 1 and Address 2 values.

8.3.3 Management frames

8.3.3.1 Format of management frames

The format of a management frame is defined in Figure 8-34. The Frame Control, Duration, Address 1, Address 2, Address 3, and Sequence Control fields are present in all management frame subtypes. The maximum unencrypted MMPDU size, excluding the MAC header and FCS, is 2304 octets.

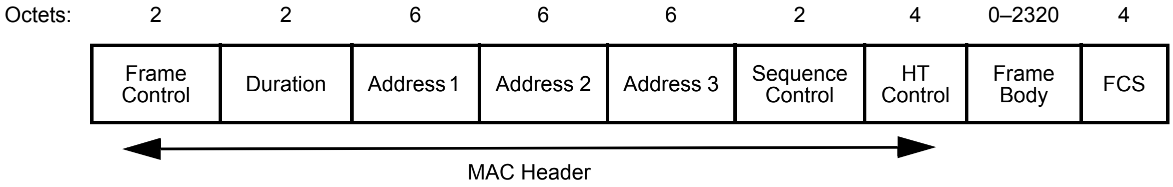


Figure 8-34—Management frame format

NOTE—The maximum frame body size shown in Figure 8-34 is for CCMP encryption with a maximum-size MMPDU (note TKIP encryption is not allowed and any Mesh Control field is held within the MMPDU, not as a separate header). The frame body size might be greater if a vendor-specific cipher suite is used.

A STA uses the contents of the Address 1 field to perform the address matching for receive decisions. In the case where the Address 1 field contains a group address and the frame subtype is other than Beacon or the frame subtype Action, Category Multihop Action (Multihop Action frame), the Address 3 field also is validated to verify that the group addressed frame originated from a STA in the BSS of which the receiving STA is a member or from a mesh STA to which mesh peering is maintained. Details of addressing and forwarding of the group addressed frame in an MBSS are defined in 9.32.5. When the Address 1 field contains a group address and the frame subtype is either Probe Request or Action with Category Public, a wildcard BSSID value matches all receiving STA's BSSIDs. If the frame subtype is Beacon, other address matching rules apply, as specified in 10.1.3.5. Frames of subtype Probe Request with a group address in the Address 1 field are additionally processed as described in 10.1.4.3.2. If the frame subtype is Action, the Category is Public, and the Action is 20/40 BSS Coexistence Management, then additional address matching rules for receive decisions apply as specified in 10.15 and 10.17.

The address fields for all management frames except Multihop Action frames are as follows:

a) The Address 1 field of the management frame is the RA (=DA) and is determined as the destination of the frame.
b) The Address 2 field of the management frame is the TA (=SA) and is determined as the address of the STA transmitting the frame.
c) The Address 3 field of the management frame is set and determined as follows:
    1) In management frames of subtype Probe Request, the Address 3 field is the BSSID. The BSSID is either a specific BSSID as described in item 4) below or the wildcard BSSID as defined in the procedures specified in 10.1.4.
    2) In management frames of subtype Action, Category Public, the Address 3 field is the BSSID. The BSSID value is set according to 10.19.
    3) If dot11OCBActivated is true, the Address 3 field is the wildcard BSSID.
    4) Otherwise:
        i) If the STA is contained within an AP or is associated with an AP, the Address 3 field is the BSSID. The BSSID is the address currently in use by the STA contained in the AP.
        ii) If the STA is contained within an AP or is transmitting the management frame to an AP, the Address 3 field is the BSSID. The BSSID is the address currently in use by the STA contained in the AP.
        iii) If the STA is transmitting the management frame to one or more members of an IBSS, the Address 3 field is the BSSID of the IBSS.
        iv) If the STA is a mesh STA, the Address 3 field is the TA.

The address fields for the Multihop Action frame are as follows:

— The Address 1 field is the RA and is determined as the address of the receiver of the frame.

- The Address 2 field is the TA and is determined as the address of the transmitter of the frame.
- The Address 3 field is the DA and is determined as the address of the destination mesh STA of the frame.

NOTE—Address 4 is included in the Mesh Control field.

Within all management frames sent by STAs during the CFP under PCF, the Duration field is set to the value 32 768. Within all management frames sent by the QoS STA, the Duration field contains a duration value as defined in 8.2.5. Within all management frames sent during the CP by non-QoS STAs, the Duration field is set according to the following rules:

- If the DA field contains a group address, the duration value is set to 0.
- If the More Fragments bit is 0 in the Frame Control field of a frame and the DA field contains an individual address, the duration value is set to the time, in microseconds, required to transmit one ACK frame, plus one SIFS interval.
- If the More Fragments bit is 1 in the Frame Control field of a frame, and the DA field contains an individual address, the duration value is set to the time, in microseconds, required to transmit the next fragment of this management frame, plus two ACK frames, plus three SIFS intervals.

The duration value calculation for the management frame is based on the rules in 9.7 that determine the data rate at which the control frames in the frame exchange sequence are transmitted. If the calculated duration includes a fractional microsecond, that value is rounded up to the next higher integer. All STAs process Duration field values less than or equal to 32 767 from valid management frames to update their NAV settings as appropriate under the coordination function rules.

The HT Control field is defined in 8.2.4.6. The presence of the HT Control field is determined by the Order subfield of the Frame Control field, as specified in 8.2.4.1.10.

The frame body consists of the fields followed by the elements defined for each management frame subtype. All fields and elements are mandatory unless stated otherwise and appear in the specified, relative order. STAs that encounter an element ID they do not recognize in the frame body of a received management frame ignore that element and continue to parse the remainder of the management frame body (if any) for additional elements with recognizable element IDs. See 9.24.7. Unused element ID codes are reserved.

Gaps may exist in the ordering of fields and elements within frames. The order that remains is ascending.

8.3.3.2 Beacon frame format

The frame body of a management frame of subtype Beacon contains the information shown in Table 8-20.

Table 8-20—Beacon frame body

Order	Information	Notes
1	Timestamp	
2	Beacon interval	
3	Capability	
4	Service Set Identifier (SSID)	If dot11MeshActivated is true, the SSID element is the wildcard value as described in 8.4.2.2.
5	Supported rates	
6	Frequency-Hopping (FH) Parameter Set	The FH Parameter Set element is present within Beacon frames generated by STAs using FH PHYs.

Table 8-20—Beacon frame body *(continued)*

Order	Information	Notes
7	DSSS Parameter Set	The DSSS Parameter Set element is present within Beacon frames generated by STAs using Clause 16, Clause 17, and Clause 19 PHYs. The element is present within Beacon frames generated by STAs using a Clause 20 PHY in the 2.4 GHz band.
8	CF Parameter Set	The CF Parameter Set element is present only within Beacon frames generated by APs supporting a PCF. This element is not present if dot11HighThroughputOptionImplemented is true and the Dual CTS Protection field of the HT Operation element is 1.
9	IBSS Parameter Set	The IBSS Parameter Set element is present only within Beacon frames generated by STAs in an IBSS.
10	Traffic indication map (TIM)	The TIM element is present only within Beacon frames generated by APs or mesh STAs.
11	Country	The Country element is present if dot11MultiDomainCapabilityActivated is true or dot11SpectrumManagementRequired is true or dot11RadioMeasurementActivated is true.
12	FH Parameters	FH Parameters as specified in 8.4.2.11 are optionally present if dot11MultiDomainCapabilityActivated is true.
13	FH Pattern Table	FH Pattern Table information as specified in 8.4.2.12 are optionally present if dot11MultiDomainCapabilityActivated is true.
14	Power Constraint	The Power Constraint element is present if dot11SpectrumManagementRequired is true and is optionally present if dot11RadioMeasurementActivated is true.
15	Channel Switch Announcement	Channel Switch Announcement element is optionally present if dot11SpectrumManagementRequired is true.
16	Quiet	The Quiet element is optionally present if dot11SpectrumManagementRequired is true or dot11RadioMeasurementActivated is true.
17	IBSS DFS	IBSS DFS element is present if dot11SpectrumManagementRequired is true in an IBSS.
18	TPC Report	The TPC Report element is present if dot11SpectrumManagementRequired is true or dot11RadioMeasurementActivated is true.
19	ERP	The ERP element is present within Beacon frames generated by STAs using extended rate PHYs (ERPs) defined in Clause 19 and is optionally present in other cases.
20	Extended Supported Rates	The Extended Supported Rates element is present if there are more than eight supported rates, and it is optional otherwise.
21	RSN	The RSNE is present within Beacon frames generated by STAs that have dot11RSNAActivated equal to true.
22	BSS Load	The BSS Load element is present if dot11QosOptionImplemented and dot11QBSSLoadImplemented are both true.
23	EDCA Parameter Set	The EDCA Parameter Set element is present if dot11QosOptionImplemented is true, and dot11MeshActivated is false, and the QoS Capability element is not present.

Table 8-20—Beacon frame body *(continued)*

Order	Information	Notes
24	QoS Capability	The QoS Capability element is present if dot11QosOptionImplemented is true, and dot11MeshActivated is false, and EDCA Parameter Set element is not present.
25	AP Channel Report	If dot11RMAPChannelReportActivated is true, one AP Channel Report element is present for each operating class that has at least 1 channel to report.
26	BSS Average Access Delay	The BSS Average Access Delay element is present if dot11RMBSSAverageAccessDelayActivated is true and the value of the AP Average Access Delay field is not equal to 255 (measurement not available); otherwise, the BSS Average Access Delay element is optionally present if dot11RMBSSAverageAccessDelayActivated is true.
27	Antenna	The Antenna element is present if dot11RMAntennaInformationActivated is true and the value of the Antenna ID field is not equal to 0 (unknown antenna); otherwise, the Antenna element is optionally present if dot11RMAntennaInformationActivated is true.
28	BSS Available Admission Capacity	The BSS Available Admission Capacity element is present if dot11RMBSSAvailableAdmissionCapacityActivated is true with the following exceptions: 1) when Available Admission Capacity Bitmask equals 0 (Available Admission Capacity List contains no entries), or 2) when the BSS Load element is present and the Available Admission Capacity Bitmask states that only AC_VO is present in the Available Admission Capacity List field.
29	BSS AC Access Delay	The BSS AC Access Delay element is present if dot11RMBSSAverageAccessDelayActivated is true and at least one field of the element is not equal to 255 (measurement not available); otherwise, the BSS AC Access Delay element is optionally present if dot11RMBSSAverageAccessDelayActivated is true.
30	Measurement Pilot Transmission	The Measurement Pilot Transmission element is present if dot11RMMeasurementPilotActivated is a value between 2 and 7.
31	Multiple BSSID	One or more Multiple BSSID elements are present if dot11RMMeasurementPilotActivated is a value between 2 and 7 and the AP is a member of a Multiple BSSID Set (see 10.11.14) with two or more members, or if dot11MgmtOptionMultiBSSIDActivated is true, or if dot11InterworkingServiceActivated is true and the AP is a member of a Multiple BSSID Set with two or more members and at least one dot11GASAdvertisementID MIB attribute exists.
32	RM Enabled Capabilities	RM Enabled Capabilities element is present if dot11RadioMeasurementActivated is true.
33	Mobility Domain	The Mobility Domain element (MDE) is present if dot11FastBSSTransitionActivated is true.
34	DSE registered location	The DSE Registered Location element is present if dot11LCIDSERequired is true.
35	Extended Channel Switch Announcement	The Extended Channel Switch Announcement element is optionally present if dot11ExtendedChannelSwitchActivated is true.
36	Supported Operating Classes	The Supported Operating Classes element is present if dot11ExtendedChannelSwitchActivated is true.

Table 8-20—Beacon frame body *(continued)*

Order	Information	Notes
37	HT Capabilities	The HT Capabilities element is present when dot11HighThroughputOptionImplemented attribute is true.
38	HT Operation	The HT Operation element is included by an AP and a mesh STA when dot11HighThroughputOptionImplemented attribute is true.
39	20/40 BSS Coexistence	The 20/40 BSS Coexistence element is optionally present when the dot112040BSSCoexistenceManagementSupport attribute is true.
40	Overlapping BSS Scan Parameters	The Overlapping BSS Scan Parameters element is optionally present if the dot11FortyMHzOptionImplemented attribute is true.
41	Extended Capabilities	The Extended Capabilities element is optionally present if any of the fields in this element are nonzero.
42	FMS Descriptor	The FMS Descriptor element is present if dot11MgmtOptionFMSActivated is true.
43	QoS Traffic Capability	The QoS Traffic Capability element is optionally present if dot11MgmtOptionACStationCountActivated is true.
44	Time Advertisement	The Time Advertisement element is present every dot11TimeAdvertisementIntervalDTIMs if dot11MgmtOptionUTCTSFOffsetActivated is true.
45	Interworking	The Interworking element is present if dot11InterworkingServiceActivated is true.
46	Advertisement Protocol	Advertisement Protocol element is present if dot11InterworkingServiceActivated is true and at least one dot11GASAdvertisementID MIB attribute exists.
47	Roaming Consortium	The Roaming Consortium element is present if dot11InterworkingServiceActivated is true and the dot11RoamingConsortiumTable has at least one entry.
48	Emergency Alert Identifier	One or more Emergency Alert Identifier elements are present if dot11EASActivated is true and there are one or more EAS message(s) active in the network.
49	Mesh ID	The Mesh ID element is present if dot11MeshActivated is true.
50	Mesh Configuration	The Mesh Configuration element is present if dot11MeshActivated is true.
51	Mesh Awake Window	The Mesh Awake Window element is optionally present if dot11MeshActivated is true.
52	Beacon Timing	The Beacon Timing element is optionally present if both dot11MeshActivated and dot11MBCAActivated are true.
53	MCCAOP Advertisement Overview	The MCCAOP Advertisement Overview element is optionally present if both dot11MeshActivated and dot11MCCAActivated are true.
54	MCCAOP Advertisement	One or more MCCAOP Advertisement elements are optionally present if both dot11MeshActivated and dot11MCCAActivated are true.

Table 8-20—Beacon frame body *(continued)*

Order	Information	Notes
55	Mesh Channel Switch Parameters	The Mesh Channel Switch Parameters element is present when dot11MeshActivated is true and either Channel Switch Announcement element or Extended Channel Switch Announcement element is present.
Last	Vendor Specific	One or more vendor-specific elements are optionally present. These elements follow all other elements.

8.3.3.3 ATIM frame format

The frame body of a management frame of subtype ATIM is null.

8.3.3.4 Disassociation frame format

The frame body of a management frame of subtype Disassociation contains the information shown in Table 8-21.

Table 8-21—Disassociation frame body

Order	Information
1	Reason code
2 – (Last – 1)	One or more vendor-specific elements are optionally present.
Last	The Management MIC element (MME) is present when management frame protection is enabled at the AP and the frame is a group addressed frame.
NOTE—The MME appears after all fields that it protects. Therefore, it appears last in the frame body to protect the frames as specified in 11.4.4.	

8.3.3.5 Association Request frame format

The frame body of a management frame of subtype Association Request contains the information shown in Table 8-22.

Table 8-22—Association Request frame body

Order	Information	Notes
1	Capability	
2	Listen Interval	
3	SSID	
4	Supported rates	
5	Extended Supported Rates	The Extended Supported Rates element is present if there are more than eight supported rates, and it is optional otherwise.

Table 8-22—Association Request frame body *(continued)*

Order	Information	Notes
6	Power Capability	The Power Capability element is present if dot11SpectrumManagementRequired is true or dot11RadioMeasurementActivated is true.
7	Supported Channels	The Supported Channels element is present if dot11SpectrumManagementRequired is true and dot11ExtendedChannelSwitchActivated is false.
8	RSN	The RSNE is present if dot11RSNAActivated is true.
9	QoS Capability	The QoS Capability element is present if dot11QosOption-Implemented is true.
10	RM Enabled Capabilities	RM Enabled Capabilities element is present if dot11RadioMeasurementActivated is true.
11	Mobility Domain	The MDE is present in an Association Request frame if dot11FastBSSTransitionActivated is true and if the frame is being sent to an AP that advertised its FT capability in the MDE in its Beacon or Probe Response frame (i.e., AP also has dot11FastBSSTransitionActivated equal to true).
12	Supported Operating Classes	The Supported Operating Classes element is present if dot11ExtendedChannelSwitchActivated is true.
13	HT Capabilities	The HT Capabilities element is present when dot11HighThroughputOptionImplemented attribute is true.
14	20/40 BSS Coexistence	The 20/40 BSS Coexistence element is optionally present when the dot112040BSSCoexistenceManagementSupport attribute is true.
15	Extended Capabilities	The Extended Capabilities element is optionally present if any of the fields in this element are nonzero.
16	QoS Traffic Capability	The QoS Traffic Capability element is present if dot11MgmtOptionQoSTrafficCapabilityActivated is true.
17	TIM Broadcast Request	The TIM Broadcast Request element is present if dot11MgmtOptionTIMBroadcastActivated is true.
18	Interworking	The Interworking element is present if dot11InterworkingServiceActivated is true and the non-AP STA is requesting unauthenticated access to emergency services (see 10.3.5).
Last	Vendor Specific	One or more vendor-specific elements are optionally present. These elements follow all other elements.

8.3.3.6 Association Response frame format

The frame body of a management frame of subtype Association Response contains the information shown in Table 8-23.

Table 8-23—Association Response frame body

Order	Information	Notes
1	Capability	
2	Status code	
3	AID	
4	Supported rates	
5	Extended Supported Rates	The Extended Supported Rates element is present if there are more than eight supported rates, and is optionally present otherwise.
6	EDCA Parameter Set	
7	RCPI	The RCPI element is present if dot11RMRCPIMeasurementActivated is true.
8	RSNI	The RSNI element is present if dot11RMRSNIMeasurementActivated is true.
9	RM Enabled Capabilities	RM Enabled Capabilities element is present if dot11RadioMeasurementActivated is true.
10	Mobility Domain	An MDE is present in an Association Response frame when dot11FastBSSTransitionActivated is true and this frame is a response to an Association Request frame that contained an MDE (i.e., an FT initial mobility domain association exchange).
11	Fast BSS Transition	A Fast BSS Transition element (FTE) is present in an Association Response frame when dot11FastBSSTransitionActivated is true, dot11RSNAActivated is true, and this frame is a response to an Association Request frame that contained an MDE (i.e., an FT initial mobility domain association exchange in an RSN).
12	DSE registered location	The DSE Registered Location element is present if dot11LCIDSERequired is true.
13	Timeout Interval (Association Comeback time)	A Timeout Interval element (TIE) containing the Association Comeback time is present when dot11RSNAActivated is true, dot11RSNAProtectedManagementFramesActivated is true, and the association request is rejected with a status code 30.
14	HT Capabilities	The HT Capabilities element is present when dot11HighThroughputOptionImplemented attribute is true.
15	HT Operation	The HT Operation element is included by an AP when dot11HighThroughputOptionImplemented attribute is true.
16	20/40 BSS Coexistence	The 20/40 BSS Coexistence element is optionally present when the dot112040BSSCoexistenceManagementSupport attribute is true.
17	Overlapping BSS Scan Parameters	The Overlapping BSS Scan Parameters element is optionally present if the dot11FortyMHzOptionImplemented attribute is true.
18	Extended Capabilities	The Extended Capabilities element is optionally present if any of the fields in this element are nonzero.

Table 8-23—Association Response frame body *(continued)*

Order	Information	Notes
19	BSS Max Idle Period	The BSS Max Idle Period element is present if dot11WirelessManagementImplemented is true.
20	TIM Broadcast Response	The TIM Broadcast Response element is present if dot11MgmtOptionTIMBroadcastActivated is true and the TIM Broadcast Request element is present in the Association Request that elicited this Association Response frame.
21	QoS Map	QoS Map is present if dot11QosMapActivated is true and the QoS Map field in the Extended Capabilities element of the corresponding Association Request frame is 1.
Last	Vendor Specific	One or more vendor-specific elements are optionally present. These elements follow all other elements.

8.3.3.7 Reassociation Request frame format

The frame body of a management frame of subtype Reassociation Request contains the information shown in Table 8-24.

Table 8-24—Reassociation Request frame body

Order	Information	Notes
1	Capability	
2	Listen Interval	
3	Current AP address	
4	SSID	
5	Supported rates	
6	Extended Supported Rates	The Extended Supported Rates element is present if there are more than eight supported rates, and it is optional otherwise.
7	Power Capability	The Power Capability element is present if dot11SpectrumManagementRequired is true or dot11RadioMeasurementActivated is true.
8	Supported Channels	The Supported Channels element is present if dot11SpectrumManagementRequired is true and dot11ExtendedChannelSwitchActivated is false.
9	RSN	The RSNE is present only if dot11RSNAActivated is true.
10	QoS Capability	The QoS Capability element is present if dot11QosOptionImplemented is true.
11	RM Enabled Capabilities	RM Enabled Capabilities element is present if dot11RadioMeasurementActivated is true.
12	Mobility Domain	The MDE is present in a Reassociation Request frame if dot11FastBSSTransitionActivated is true and the frame is being sent to an AP that advertised its FT Capability in the MDE in its Beacon or Probe Response frame (i.e., AP also has dot11FastBSSTransitionActivated is true).

Table 8-24—Reassociation Request frame body *(continued)*

Order	Information	Notes
13	Fast BSS Transition	An FTE is present in a Reassociation Request frame if dot11FastBSSTransitionActivated is true and dot11RSNAAuthenticationSuiteSelected is 00-0F-AC:3, 00-0F-AC:4, or 00-0F-AC:9 (i.e., part of a fast BSS transition in an RSN).
14	Resource information container (RIC)	The set of elements that formulate a RIC-Request is optionally present in a Reassociation Request frame if — dot11FastBSSTransitionActivated is true, — The FT Resource Request Protocol is not used, — The frame is being sent to an AP that advertised its FT capability in the MDE in its Beacon or Probe Response frame (i.e., AP also has dot11FastBSSTransitionActivated is true), and — Either dot11RSNAAuthenticationSuiteSelected is 00-0F-AC:3 or 00-0F-AC:4 (i.e., part of a fast BSS transition in an RSN) or dot11RSNAActivated is false (i.e., not in an RSN).
15	Supported Operating Classes	The Supported Operating Classes element is present if dot11ExtendedChannelSwitchActivated is true.
16	HT Capabilities	The HT Capabilities element is present when dot11HighThroughputOptionImplemented attribute is true.
17	20/40 BSS Coexistence	The 20/40 BSS Coexistence element is optionally present when the dot112040BSSCoexistenceManagementSupport attribute is true.
18	Extended Capabilities	The Extended Capabilities element is optionally present if any of the fields in this element are nonzero.
19	QoS Traffic Capability	The QoS Traffic Capability element is present if dot11MgmtOptionQoSTrafficCapabilityActivated is true.
20	TIM Broadcast Request	The TIM Broadcast Request element is present if dot11MgmtOptionTIMBroadcastActivated is true.
21	FMS Request	The FMS Request element may be present if dot11MgmtOptionFMSActivated is true.
22	DMS Request	The DMS Request element may be present if dot11MgmtOptionDMSActivated is true.
23	Interworking	The Interworking element is present if dot11InterworkingServiceActivated is true and the non-AP STA is requesting unauthenticated access to emergency services (see 10.3.5).
Last	Vendor Specific	One or more vendor-specific elements are optionally present. These elements follow all other elements.

8.3.3.8 Reassociation Response frame format

The frame body of a management frame of subtype Reassociation Response contains the information shown in Table 8-25.

Table 8-25—Reassociation Response frame body

Order	Information	Notes
1	Capability	
2	Status code	
3	AID	
4	Supported rates	
5	Extended Supported Rates	The Extended Supported Rates element is present if there are more than eight supported rates, and it is optional otherwise.
6	EDCA Parameter Set	
7	RCPI	The RCPI element is present if dot11RMRCPIMeasurementActivated is true.
8	RSNI	The RSNI element is present if dot11RMRSNIMeasurementActivated is true.
9	RM Enabled Capabilities	RM Enabled Capabilities element is present if dot11RadioMeasurementActivated is true.
10	RSN	An RSNE is present in a Reassociation Response frame if dot11FastBSSTransitionActivated is true, dot11RSNAActivated is true, and this frame is a response to a Reassociation Request frame that contained an FTE (i.e., part of a fast BSS transition in an RSN).
11	Mobility Domain	An MDE is present in a Reassociation Response frame if dot11FastBSSTransitionActivated is true and this frame is a response to a Reassociation Request frame that contained an MDE (i.e., either an FT initial mobility domain association exchange or part of a fast BSS transition).
12	Fast BSS Transition	An FTE is present in a Reassociation Response frame if dot11FastBSSTransitionActivated is true, dot11RSNAActivated is true, and this frame is a response to a Reassociation Request frame that contained an MDE (i.e., either an FT initial mobility domain association exchange or part of a fast BSS transition in an RSN).
13	RIC	The set of elements that formulate a RIC-Response is present in a Reassociation Response frame if dot11FastBSSTransitionActivated is true and this frame is a response to a Reassociation Request frame that contained a RIC-Request.
14	DSE registered location	The DSE Registered Location element is present if dot11LCIDSERequired is true.
15	Timeout Interval (Association Comeback time)	A TIE containing the Association Comeback time is present when dot11RSNAActivated is true, dot11RSNAProtectedManagementFramesActivated is true, and the reassociation is rejected with status code 30.
16	HT Capabilities	The HT Capabilities element is present when dot11HighThroughputOptionImplemented attribute is true.

Table 8-25—Reassociation Response frame body *(continued)*

Order	Information	Notes
17	HT Operation	The HT Operation element is included by an AP when dot11HighThroughputOptionImplemented attribute is true.
18	20/40 BSS Coexistence	The 20/40 BSS Coexistence element is optionally present when the dot112040BSSCoexistenceManagementSupport attribute is true.
19	Overlapping BSS Scan Parameters	The Overlapping BSS Scan Parameters element is optionally present if the dot11FortyMHzOptionImplemented attribute is true.
20	Extended Capabilities	The Extended Capabilities element is optionally present if any of the fields in this element are nonzero.
21	BSS Max Idle Period	The BSS Max Idle Period element is present if dot11WirelessManagementImplemented is true.
22	TIM Broadcast Response	The TIM Broadcast Response element is present if dot11MgmtOptionTIMBroadcastActivated is true and the TIM Broadcast Request element is present in the Reassociation Request frame that elicited this Reassociation Response frame.
23	FMS Response	The FMS Response element is present if dot11MgmtOptionFMSActivated is true and the FMS Request element is present in the Reassociation Request frame that elicited this Reassociation Response frame.
24	DMS Response	The DMS Response element is present if dot11MgmtOptionDMSActivated is true and the DMS Request element is present in the Reassociation Request frame that elicited this Reassociation Response frame.
25	QoS Map	QoS Map is present if dot11QosMapActivated is true and the QoS Map field in the Extended Capabilities element of the corresponding Reassociation Request frame is 1.
Last	Vendor Specific	One or more vendor-specific elements are optionally present. These elements follow all other elements.

8.3.3.9 Probe Request frame format

The frame body of a management frame of subtype Probe Request contains the information shown in Table 8-26.

Table 8-26—Probe Request frame body

Order	Information	Notes
1	SSID	If dot11MeshActivated is true, the SSID element is the wildcard value as described in 8.4.2.2.
2	Supported rates	
3	Request information	The Request element is optionally present if dot11MultiDomainCapabilityActivated is true.
4	Extended Supported Rates	The Extended Supported Rates element is present if there are more than eight supported rates, and is optionally present otherwise.

Table 8-26—Probe Request frame body *(continued)*

Order	Information	Notes
5	DSSS Parameter Set	The DSSS Parameter Set element is present within Probe Request frames generated by STAs using Clause 16, Clause 17, or Clause 19 PHYs if dot11RadioMeasurementActivated is true. The DSSS Parameter Set element is present within Probe Request frames generated by STAs using a Clause 20 PHY in the 2.4 GHz band if dot11RadioMeasurementActivated is true. The DSSS Parameter Set element is optionally present within Probe Request frames generated by STAs using Clause 16, Clause 17, or Clause 19 PHYs if dot11RadioMeasurementActivated is false. The DSSS Parameter Set element is optionally present within Probe Request frames generated by STAs using a Clause 20 PHY in the 2.4 GHz band if dot11RadioMeasurementActivated is false.
6	Supported Operating Classes	The Supported Operating Classes element is present if dot11ExtendedChannelSwitchActivated is true.
7	HT Capabilities	The HT Capabilities element is present when dot11HighThroughputOptionImplemented attribute is true.
8	20/40 BSS Coexistence	The 20/40 BSS Coexistence element is optionally present when the dot112040BSSCoexistenceManagementSupport attribute is true.
9	Extended Capabilities	The Extended Capabilities element is optionally present if any of the fields in this element are nonzero.
10	SSID List	The SSID List element is optionally present if dot11MgmtOptionSSIDListActivated is true.
11	Channel Usage	The Channel Usage element is optionally present if dot11MgmtOptionChannelUsageActivated is true.
12	Interworking	The Interworking element is present if dot11InterworkingServiceActivated is true.
13	Mesh ID	The Mesh ID element is present if dot11MeshActivated is true.
Last	Vendor Specific	One or more vendor-specific elements are optionally present. These elements follow all other elements.

8.3.3.10 Probe Response frame format

The frame body of a management frame of subtype Probe Response contains the information shown in Table 8-27. See additional details and procedures in 9.18.3 and 10.1.4.

Table 8-27—Probe Response frame body

Order	Information	Notes
1	Timestamp	
2	Beacon interval	
3	Capability	

Table 8-27—Probe Response frame body *(continued)*

Order	Information	Notes
4	SSID	If dot11MeshActivated is true, the SSID element is the wildcard value as described in 8.4.2.2.
5	Supported rates	
6	FH Parameter Set	The FH Parameter Set element is present within Probe Response frames generated by STAs using FH PHYs.
7	DSSS Parameter Set	The DSSS Parameter Set element is present within Probe Response frames generated by STAs using Clause 16, Clause 17, and Clause 19 PHYs. The DSSS Parameter Set element is present within Probe Response frames generated by STAs using a Clause 20 PHY in the 2.4 GHz band.
8	CF Parameter Set	The CF Parameter Set element is present only within Probe Response frames generated by APs supporting a PCF.
9	IBSS Parameter Set	The IBSS Parameter Set element is present only within Probe Response frames generated by STAs in an IBSS.
10	Country	The Country element is present if dot11MultiDomainCapabilityActivated is true or dot11SpectrumManagementRequired is true or dot11RadioMeasurementActivated is true.
11	FH Parameters	The FH Parameters element, as specified in 8.4.2.11, is optionally present if dot11MultiDomainCapabilityActivated is true.
12	FH Pattern Table	The FH Pattern Table element, as specified in 8.4.2.12, is optionally present if dot11MultiDomainCapabilityActivated is true.
13	Power Constraint	The Power Constraint element is present if dot11SpectrumManagementRequired is true and is optionally present if dot11RadioMeasurementActivated is true.
14	Channel Switch Announcement	The Channel Switch Announcement element is optionally present if dot11SpectrumManagementRequired is true.
15	Quiet	The Quiet element is optionally present if dot11SpectrumManagementRequired is true or if dot11RadioMeasurementActivated is true.
16	IBSS DFS	The IBSS DFS element is present if dot11SpectrumManagementRequired is true in an IBSS.
17	TPC Report	The TPC Report element is present if dot11SpectrumManagementRequired is true or dot11RadioMeasurementActivated is true.
18	ERP	The ERP element is present within Probe Response frames generated by STAs using ERPs and is optionally present otherwise.
19	Extended Supported Rates	The Extended Supported Rates element is present if there are more than eight supported rates, and it is optionally present otherwise.
20	RSN	The RSNE is present only if dot11RSNAActivated is true.
21	BSS Load	The BSS Load element is present if dot11QosOption-Implemented and dot11QBSSLoadImplemented are both true.
22	EDCA Parameter Set	The EDCA Parameter Set element is present if dot11QosOptionImplemented is true and dot11MeshActivated is false.

Table 8-27—Probe Response frame body *(continued)*

Order	Information	Notes
23	Measurement Pilot Transmission	The Measurement Pilot Transmission element is present if dot11RMMeasurementPilotActivated is between 2 and 7.
24	Multiple BSSID	One or more Multiple BSSID elements are present if dot11RMMeasurementPilotActivated is between 2 and 7 and the AP is a member of a Multiple BSSID Set (see 10.11.14) with two or more members, or if dot11MgmtOptionMultiBSSIDActivated is true, or if dot11InterworkingServiceActivated is true and the AP is a member of a Multiple BSSID Set with two or more members and at least one dot11GASAdvertisementID MIB attribute exists.
25	RM Enabled Capabilities	The RM Enabled Capabilities element is present if dot11RadioMeasurementActivated is true.
26	AP Channel Report	If dot11RMAPChannelReportActivated is true, one AP Channel Report element is optionally present for each operating class that has at least 1 channel to report.
27	BSS Average Access Delay	The BSS Average Access Delay element is optionally present if dot11RMBSSAverageAccessDelayActivated is true and the value of the AP Average Access Delay field is not equal to 255 (measurement not available).
28	Antenna	The Antenna element is optionally present if dot11RMAntennaInformationActivated is true and the value of the Antenna ID field is not equal to 0 (unknown antenna).
29	BSS Available Admission Capacity	The BSS Available Admission Capacity element is optionally present if dot11RMBSSAvailableAdmissionCapacityActivated is true with the following exceptions: 1) when Available Admission Capacity Bitmask equals 0 (Available Admission Capacity List contains no entries), or 2) when the BSS Load element is present and the Available Capacity Bitmask equals 256 (Available Admission Capacity List contains only the AC_VO entry).
30	BSS AC Access Delay	The BSS AC Access Delay element is optionally present if dot11RMBSSAverageAccessDelayActivated is true and at least one field of the element is not equal to 255 (measurement not available).
31	Mobility Domain	The MDE is present if dot11FastBSSTransitionActivated is true.
32	DSE registered location	The DSE Registered Location element is present if dot11LCIDSERequired is true.
33	Extended Channel Switch Announcement	The Extended Channel Switch Announcement element is optionally present if dot11ExendedChannelSwitchActivated is true.
34	Supported Operating Classes	The Supported Operating Classes element is present if dot11ExtendedChannelSwitchActivated is true.
35	HT Capabilities	The HT Capabilities element is present when dot11HighThroughputOptionImplemented attribute is true.
36	HT Operation	The HT Operation element is included by an AP and a mesh STA when dot11HighThroughputOptionImplemented attribute is true.
37	20/40 BSS Coexistence	The 20/40 BSS Coexistence element is optionally present when the dot112040BSSCoexistenceManagementSupport attribute is true.
38	Overlapping BSS Scan Parameters	The Overlapping BSS Scan Parameters element is optionally present if the dot11FortyMHzOptionImplemented attribute is true.
39	Extended Capabilities	The Extended Capabilities element is optionally present if any of the fields in this element are nonzero.

Table 8-27—Probe Response frame body *(continued)*

Order	Information	Notes
40	QoS Traffic Capability	The QoS Traffic Capability element is optionally present if dot11MgmtOptionACStationCountActivated is true.
41	Channel Usage	The Channel Usage element is present if the Channel Usage element is present in the Probe Request frame and dot11MgmtOptionChannelUsageActivated is true.
42	Time Advertisement	The Time Advertisement element is present if dot11MgmtOptionUTCTSFOffsetActivated is true.
43	Time Zone	The Time Zone element is present if dot11MgmtOptionUTCTSFOffsetActivated is true.
44	Interworking	The Interworking element is present if dot11InterworkingServiceActivated is true.
45	Advertisement Protocol	Advertisement Protocol element is present if dot11InterworkingServiceActivated is true and at least one dot11GASAdvertisementID MIB attribute exists.
46	Roaming Consortium	The Roaming Consortium element is present if dot11InterworkingServiceActivated is true and the dot11RoamingConsortiumTable has at least one entry.
47	Emergency Alert Identifier	One or more Emergency Alert Identifier elements are present if dot11EASActivated is true and there are one or more EAS message(s) active in the network.
48	Mesh ID	The Mesh ID element is present if dot11MeshActivated is true.
49	Mesh Configuration	The Mesh Configuration element is present if dot11MeshActivated is true.
50	Mesh Awake Window	The Mesh Awake Window element is optionally present if dot11MeshActivated is true.
51	Beacon Timing	The Beacon Timing element is optionally present if both dot11MeshActivated and dot11MBCAActivated are true.
52	MCCAOP Advertisement Overview	The MCCAOP Advertisement Overview element is optionally present if both dot11MeshActivated and dot11MCCAActivated are true.
53	MCCAOP Advertisement	One or more MCCAOP Advertisement elements are optionally present if both dot11MeshActivated and dot11MCCAActivated are true.
54	Mesh Channel Switch Parameters	The Mesh Channel Switch Parameters element is present if dot11MeshActivated is true and either Channel Switch Announcement element or Extended Channel Switch Announcement element is present.
Last–*1*	Vendor Specific	One or more vendor-specific elements are optionally present. These elements follow all other elements, except the Requested elements.
Last–*n*	Requested elements	Elements requested by the Request element of the Probe Request frame are present if dot11MultiDomainCapabilityActivated is true. See 10.1.4.3.2.

8.3.3.11 Authentication frame format

The frame body of a management frame of subtype Authentication contains the information shown in Table 8-28. FT authentication is used when FT support is advertised by the AP and dot11FastBSSTransitionActivated is true in the STA. SAE authentication is used when dot11MeshActiveAuthenticationProtocol is sae (1).

Table 8-28—Authentication frame body

Order	Information	Notes
1	Authentication algorithm number	
2	Authentication transaction sequence number	
3	Status code	The status code information is reserved in certain Authentication frames as defined in Table 8-29.
4	Challenge text	The challenge text element is present only in certain Authentication frames as defined in Table 8-29.
5	RSN	The RSNE is present in the FT Authentication frames as defined in Table 8-29.
6	Mobility Domain	The MDE is present in the FT Authentication frames as defined in Table 8-29.
7	Fast BSS Transition	An FTE is present in the FT Authentication frames as defined in Table 8-29.
8	Timeout Interval (reassociation deadline)	A TIE containing the reassociation deadline interval is present in the FT Authentication frames as defined in Table 8-29.
9	RIC	A resource information container, containing a variable number of elements, is present in the FT Authentication frames as defined in Table 8-29.
10	Finite Cyclic Group	An unsigned integer indicating a finite cyclic group as described in 11.3.4. This is present in SAE authentication frames as defined in Table 8-29.
11	Anti-Clogging Token	A random bit-string used for anti-clogging purposes as described in 11.3.6. This is present in SAE authentication frames as defined in Table 8-29.
12	Send-Confirm	A binary encoding of an integer used for anti-replay purposes as described in 11.3.7.5. This is present in SAE authentication frames as defined in Table 8-29.
13	Scalar	An unsigned integer encoded as described in 11.3.7.4. This is present in SAE authentication frames as defined in Table 8-29.
14	Element	A field element from a finite field encoded as described in 11.3.7.4. This is present in SAE authentication frames as defined in Table 8-29.
15	Confirm	An unsigned integer encoded as described in 11.3.7.5. This is present in SAE authentication frames as defined in Table 8-29.
Last	Vendor Specific	One or more vendor-specific elements are optionally present. These elements follow all other elements.

Table 8-29—Presence of fields and elements in Authentication frames

Authentication algorithm	Authentication transaction sequence no.	Status code	Presence of fields 4-15
Open System	1	Reserved	Not present
Open System	2	Status	Not present
Shared Key	1	Reserved	Not present
Shared Key	2	Status	The Challenge text element is present
Shared Key	3	Reserved	The Challenge text element is present
Shared Key	4	Status	Not present
FT	1	Reserved	The Mobility Domain element is present. The Fast BSS Transition and RSNEs are present if dot11RSNAActivated is true.
FT	2	Status	The Mobility Domain element is present if Status is 0. The Fast BSS Transition and RSNEs are present if Status is 0 and dot11RSNAActivated is true.
FT	3	Reserved	The Mobility Domain element is present. The Fast BSS Transition and RSNEs are present if dot11RSNAActivated is true. The RIC element is optionally present.
FT	4	Status	The Mobility Domain element is present if Status is 0. The Fast BSS Transition and RSNEs are present if dot11RSNAActivated is true. The RIC element is optionally present if Status is 0. The TIE (reassociation deadline) is present if a RIC element is present.
SAE	1	Status	Scalar is present if Status is zero. Element is present if Status is zero. Anti-Clogging Token is present if status is 76 or if frame is in response to a previous rejection with Status 76. Finite Cyclic Group is present if Status is zero or 76.
SAE	2	Status	Send-Confirm is present. Confirm is present.

8.3.3.12 Deauthentication

The frame body of a management frame of subtype Deauthentication contains the information shown in Table 8-30.

Table 8-30—Deauthentication frame body

Order	Information
1	Reason code
2 – (Last – 1)	One or more vendor-specific elements are optionally present.
Last	The Management MIC element (MME) is present when management frame protection is enabled at the AP and the frame is a group addressed frame.
NOTE—The MME appears after all fields that it protects. Therefore, it appears last in the frame body to protect the frames as specified in 11.4.4.	

8.3.3.13 Action frame format

The frame body of a management frame of subtype Action contains the information shown in Table 8-31.

Table 8-31—Action frame body

Order	Information
1	Action
2 – (Last – 1)	One or more vendor-specific elements are optionally present. These elements are absent when the Category subfield of the Action field is Vendor-Specific, Vendor-Specific Protected, or Self-protected.
Last	The Management MIC element (MME) is present when management frame protection is enabled at the AP, the frame is a group addressed robust Action frame, and the category of the action frame does not receive privacy as indicated by Table 8-38.
NOTE—The MME appears after any fields that it protects. Therefore, it appears last in the frame body to protect the frames as specified in 11.4.4.	

8.3.3.14 Action No Ack frame format

The frame body of a management frame of subtype Action No Ack contains the information shown in Table 8-32.

Table 8-32—Action No Ack frame body

Order	Information
1	Action
Last	One or more vendor-specific elements may appear in this frame. This element follows all other elements.

NOTE—The selection of Action or Action No Ack is made per frame that uses these formats.

Unless specified as allowing the use of the Action No Ack management frame subtype, a frame described as an "Action frame" uses only the Action subtype.

8.3.3.15 Timing Advertisement frame format

The frame body of a management frame of subtype Timing Advertisement contains the information shown in Table 8-33.

Table 8-33—Timing Advertisement frame body

Order	Information	Notes
1	Timestamp	See 8.4.1.10 for Timestamp format.
2	Capability	
3	Country	The Country element is present if dot11MultidomainCapabilityActivated is true or dot11SpectrumManagementRequired is true.
4	Power Constraint	The Power Contraint element is optionally present only if the Country element is present.
5	Time Advertisement	The Time Advertisement element is optionally present. See 8.4.2.63.
6	Extended Capabilities	The Extended Capabilities element is optionally present.
Last	Vendor specific	One or more vendor-specific elements are optionally present. These elements follow all other elements.

8.4 Management frame body components

8.4.1 Fields that are not information elements

8.4.1.1 Authentication Algorithm Number field

The Authentication Algorithm Number field indicates a single authentication algorithm. The length of the Authentication Algorithm Number field is 2 octets. The Authentication Algorithm Number field is illustrated in Figure 8-35. The following values are defined for authentication algorithm number:

Authentication algorithm number = 0: Open System
Authentication algorithm number = 1: Shared Key
Authentication algorithm number = 2: Fast BSS Transition
Authentication algorithm number = 3: simultaneous authentication of equals (SAE)
Authentication algorithm number = 65 535: Vendor specific use

NOTE—The use of this value implies that a Vendor Specific element is included with more information.

All other values of authentication algorithm number are reserved.

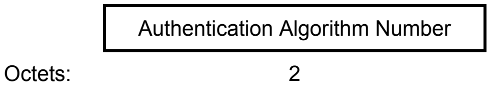


Figure 8-35—Authentication Algorithm Number field

8.4.1.2 Authentication Transaction Sequence Number field

The Authentication Transaction Sequence Number field indicates the current state of progress through a multistep transaction. The length of the Authentication Transaction Sequence Number field is 2 octets. The Authentication Transaction Sequence Number field is illustrated in Figure 8-36.

	Authentication Transaction Sequence Number
Octets:	2

Figure 8-36—Authentication Transaction Sequence Number field

8.4.1.3 Beacon Interval field

The Beacon Interval field represents the number of time units (TUs) between target beacon transmission times (TBTTs). The length of the Beacon Interval field is 2 octets. The Beacon Interval field is illustrated in Figure 8-37.

	Beacon Interval
Octets:	2

Figure 8-37—Beacon Interval field

8.4.1.4 Capability Information field

The Capability Information field contains a number of subfields that are used to indicate requested or advertised optional capabilities.

The length of the Capability Information field is 2 octets. The format of the Capability Information field is defined in Figure 8-38. No subfield is supplied for ERP as a STA supports ERP operation if it includes all of the Clause 19 mandatory rates in its supported rate set.

B0	B1	B2	B3	B4	B5	B6	B7
ESS	IBSS	CF Pollable	CF-Poll Request	Privacy	Short Preamble	PBCC	Channel Agility

B8	B9	B10	B11	B12	B13	B14	B15
Spectrum Mgmt	QoS	Short Slot Time	APSD	Radio Measurement	DSSS-OFDM	Delayed Block Ack	Immediate Block Ack

Figure 8-38—Capability Information field

Each Capability Information subfield is interpreted according to the management frame subtype, as defined in this subclause.

An AP sets the ESS subfield to 1 and the IBSS subfield to 0 within transmitted Beacon or Probe Response management frames. A STA within an IBSS sets the ESS subfield to 0 and the IBSS subfield to 1 in transmitted Beacon or Probe Response management frames. A mesh STA sets the ESS and IBSS subfields to 0 in transmitted Beacon or Probe Response management frames.

A non-AP STA sets the QoS, CF-Pollable, and CF-Poll Request subfields in Association and Reassociation Request management frames according to Table 8-34. A mesh STA sets the CF-Pollable and CF-Poll Request subfields to 0.

Table 8-34—Non-AP STA usage of QoS, CF-Pollable, and CF-Poll Request

QoS	CF-Pollable	CF-Poll request	Meaning
0	0	0	STA is not CF-Pollable
0	0	1	STA is CF-Pollable, not requesting to be placed on the CF-Polling list
0	1	0	STA is CF-Pollable, requesting to be placed on the CF-Polling list
0	1	1	STA is CF-Pollable, requesting never to be polled
1	0	0	QoS STA requesting association in a QoS BSS
1	0	1	Reserved
1	1	0	Reserved
1	1	1	Reserved

An AP sets the CF-Pollable and CF-Poll Request subfields in Beacon and Probe Response management frames according to Table 8-35. A non-QoS AP sets the CF-Pollable and CF-Poll Request subfield values in Association Response and Reassociation Response management frames equal to the values in the last Beacon or Probe Response frame that it transmitted.

Table 8-35—AP usage of QoS, CF-Pollable, and CF-Poll Request

QoS	CF-Pollable	CF-Poll Request	Meaning
0	0	0	No PC at non-QoS AP
0	0	1	PC at non-QoS AP for delivery only (no polling)
0	1	0	PC at non-QoS AP for delivery and polling
0	1	1	Reserved
1	0	0	QoS AP (HC) does not use CFP for delivery of individually addressed data frames
1	0	1	QoS AP (HC) uses CFP for delivery, but does not send CF-Polls to non-QoS STAs
1	1	0	QoS AP (HC) uses CFP for delivery, and sends CF-Polls to non-QoS STAs
1	1	1	Reserved

An AP sets the Privacy subfield to 1 within transmitted Beacon, Probe Response, Association Response, and Reassociation Response management frames if data confidentiality is required for all data frames exchanged within the BSS. If data confidentiality is not required, the Privacy subfield is set to 0.

In an RSNA, a non-AP STA in an ESS sets the Privacy subfield to 0 within transmitted Association and Reassociation Request management frames. An AP ignores the Privacy subfield within received Association and Reassociation Request management frames.

A STA within an ESS sets the Privacy subfield to 1 in DLS Request and DLS Response frames if encryption is required for all data frames exchanged. If encryption is not required, the Privacy subfield is set to 0.

A STA within an IBSS sets the Privacy subfield to 1 in transmitted Beacon or Probe Response management frames if data confidentiality is required for all data frames exchanged within the IBSS. If data confidentiality is not required, A STA in an IBSS sets the Privacy subfield to 0 within these management frames.

A mesh STA sets the Privacy subfield to 1 in transmitted Beacon or Probe Response management frames if data confidentiality is required for all data frames exchanged within the MBSS. If data confidentiality is not required, a mesh STA sets the Privacy subfield to 0 within these management frames.

A STA that includes the RSNE in Beacon and Probe Response frames sets the Privacy subfield to 1 in any frame that includes the RSNE.

An AP sets the Short Preamble subfield to 1 in transmitted Beacon, Probe Response, Association Response, and Reassociation Response MMPDUs to indicate that the use of the short preamble, as described in 17.2.2.3, is allowed within this BSS; a STA in an IBSS sets the Short Preamble subfield to 1 in transmitted Beacon when dot11ShortPreambleOptionImplemented is true. Otherwise an AP or a STA in an IBSS sets the Short Preamble subfield to 0.

A mesh STA sets the Short Preamble subfield to 1 when dot11ShortPreambleOptionImplemented is true. Otherwise, a mesh STA sets the Short Preamble subfield to 0.

An ERP STA sets dot11ShortPreambleOptionImplemented to true as all ERP devices support both long and short preamble formats.

A STA sets the Short Preamble subfield to 1 in transmitted Association Request and Reassociation Request management frames and in DLS Request and DLS Response frames when dot11ShortPreambleOptionImplemented is true. Otherwise, a STA sets the Short Preamble subfield to 0.

An AP sets the PBCC subfield to 1 in transmitted Beacon, Probe Response, Association Response, and Reassociation Response management frames to indicate that the packet binary convolutional code (PBCC) modulation option, as described in 17.4.6.7 and 19.6, is allowed within this BSS; a STA in an IBSS sets the PBCC subfield to 1 in transmitted Beacon frames when dot11PBCCOptionImplemented is true. Otherwise an AP or a STA in an IBSS sets the PBCC subfield to 0.

A non-AP STA sets the PBCC subfield to 1 in transmitted Probe Response, Association Request, Reassociation Request, DLS Request, and DLS Response frames when dot11PBCCOptionImplemented is true. Otherwise, a STA sets the PBCC subfield to 0.

The use of the ERP-PBCC option is deprecated, and this option may be removed in a later revision of the standard.

Bit 7 of the Capabilities Information field is used to indicate Channel Agility capability by the High Rate direct sequence spread spectrum (HR/DSSS) PHY or ERP. A STA sets the Channel Agility bit to 1 when Channel Agility is in use and sets it to 0 otherwise.

A STA sets the Spectrum Management subfield in the Capability Information field to 1 if dot11SpectrumManagementRequired is true; otherwise, it is set to 0.

A STA sets the QoS subfield to 1 within the Capability Information field when dot11QosOptionImplemented is true and sets it to 0 otherwise.

A STA sets the Short Slot Time subfield to 1 in transmitted Association Request, Reassociation Request, DLS Request, and DLS Response MMPDUs when dot11ShortSlotTimeOptionImplemented and dot11ShortSlotTimeOptionActivated are true. Otherwise, the STA sets the Short Slot Time subfield to 0 in transmitted Association Request and Reassociation Request MMPDUs.

An AP sets the Short Slot Time subfield in transmitted Beacon, Probe Response, Association Response, and Reassociation Response MMPDUs to indicate the currently used slot time value within this BSS. See 10.1.3.2.

See 9.3.2.12 for the operation of aSlotTime.

For IBSS and MBSS, the Short Slot Time subfield is set to 0.

An AP sets the APSD subfield to 1 within the Capability Information field when dot11APSDOptionImplemented is true and sets it to 0 otherwise. A non-AP STA always sets this subfield to 0.

An AP sets the DSSS-OFDM subfield to 1 in transmitted Beacon, Probe Response, Association Response, and Reassociation Response MMPDUs to indicate that the use of direct sequence spread spectrum with orthogonal frequency division multiplexing (DSSS-OFDM), as described in 19.7, is allowed within this BSS. To indicate that the use of DSSS-OFDM is not allowed, the DSSS-OFDM subfield is set to 0 in Beacon, Probe Response, Association Response, and Reassociation Response MMPDUs transmitted within the BSS.

A STA sets the DSSS-OFDM subfield to 1 in transmitted Association Request, Reassociation Request, DLS Request, and DLS Response MMPDUs when dot11DSSSOFDMOptionImplemented and dot11DSSSOFDMOptionActivated are true. Otherwise, a STA sets the DSSS-OFDM subfield to 0 in transmitted Association Request and Reassociation Request MMPDUs.

A STA in an IBSS sets the DSSS-OFDM subfield to 1 in transmitted Beacon and Probe Response MMPDUs when dot11DSSSOFDMOptionImplemented and dot11DSSSOFDMOptionActivated are true. Otherwise, a STA in an IBSS sets the DSSS-OFDM subfield to 0.

The use of the DSSS-OFDM option is deprecated, and this option may be removed in a later revision of the standard.

A STA sets the Delayed Block Ack subfield to 1 within the Capability Information field when dot11DelayedBlockAckOptionImplemented is true and sets it to 0 otherwise.

A STA sets the Immediate Block Ack subfield to 1 within the Capability Information field when dot11ImmediateBlockAckOptionImplemented is true and sets it to 0 otherwise.

A STA sets the Radio Measurement subfield in the Capability Information field to 1 when dot11RadioMeasurementActivated is true and sets it to 0 otherwise.

8.4.1.5 Current AP Address field

The Current AP Address field is the MAC address of the AP with which the STA is currently associated. The length of the Current AP Address field is 6 octets. The Current AP Address field is illustrated in Figure 8-39.

	Current AP Address
Octets:	6

Figure 8-39—Current AP Address field

8.4.1.6 Listen Interval field

The Listen Interval field is used to indicate to the AP how often a STA in power save mode wakes to listen to Beacon management frames. The value of this parameter is the Listen Interval parameter of the MLME-ASSOCIATE.request or MLME-REASSOCIATE.request primitive and is expressed in units of Beacon Interval. The length of the Listen Interval field is 2 octets. The Listen Interval field is illustrated in Figure 8-40.

NOTE—The value 0 might be used by a STA that never enters power save mode.

	Listen Interval
Octets:	2

Figure 8-40—Listen Interval field

An AP may use the Listen Interval information in determining the lifetime of frames that it buffers for a STA.

8.4.1.7 Reason Code field

This Reason Code field is used to indicate the reason that an unsolicited notification management frame of type Disassociation, Deauthentication, DELTS, DELBA, DLS Teardown, or Mesh Peering Close was generated. It is contained in the Mesh Channel Switch Parameters element to indicate the reason for the channel switch. It is contained in the PERR element to indicate the reason for the path error. The length of the Reason Code field is 2 octets. The Reason Code field is illustrated in Figure 8-41.

	Reason Code
Octets:	2

Figure 8-41—Reason Code field

The reason codes are defined in Table 8-36.

Table 8-36—Reason codes

Reason code	Name	Meaning
0		Reserved
1		Unspecified reason
2		Previous authentication no longer valid
3		Deauthenticated because sending STA is leaving (or has left) IBSS or ESS
4		Disassociated due to inactivity
5		Disassociated because AP is unable to handle all currently associated STAs

Table 8-36—Reason codes *(continued)*

Reason code	Name	Meaning
6		Class 2 frame received from nonauthenticated STA
7		Class 3 frame received from nonassociated STA
8		Disassociated because sending STA is leaving (or has left) BSS
9		STA requesting (re)association is not authenticated with responding STA
10		Disassociated because the information in the Power Capability element is unacceptable
11		Disassociated because the information in the Supported Channels element is unacceptable
12		Disassociated due to BSS Transition Management
13		Invalid element, i.e., an element defined in this standard for which the content does not meet the specifications in Clause 8
14		Message integrity code (MIC) failure
15		4-Way Handshake timeout
16		Group Key Handshake timeout
17		element in 4-Way Handshake different from (Re)Association Request/Probe Response/Beacon frame
18		Invalid group cipher
19		Invalid pairwise cipher
20		Invalid AKMP
21		Unsupported RSNE version
22		Invalid RSNE capabilities
23		IEEE 802.1X authentication failed
24		Cipher suite rejected because of the security policy
25		TDLS direct-link teardown due to TDLS peer STA unreachable via the TDLS direct link
26		TDLS direct-link teardown for unspecified reason
27		Disassociated because session terminated by SSP request
28		Disassociated because of lack of SSP roaming agreement
29		Requested service rejected because of SSP cipher suite or AKM requirement
30		Requested service not authorized in this location
31	SERVICE_CHANGE_ PRECLUDES_TS	TS deleted because QoS AP lacks sufficient bandwidth for this QoS STA due to a change in BSS service characteristics or operational mode (e.g., an HT BSS change from 40 MHz channel to 20 MHz channel)
32		Disassociated for unspecified, QoS-related reason
33		Disassociated because QoS AP lacks sufficient bandwidth for this QoS STA

Table 8-36—Reason codes *(continued)*

Reason code	Name	Meaning
34		Disassociated because excessive number of frames need to be acknowledged, but are not acknowledged due to AP transmissions and/or poor channel conditions
35		Disassociated because STA is transmitting outside the limits of its TXOPs
36	STA_LEAVING	Requested from peer STA as the STA is leaving the BSS (or resetting)

37	END_TS
	END_BA
	END_DLS
		Requested from peer STA as it does not want to use the mechanism

38	UNKNOWN_TS UNKNOWN_BA	Requested from peer STA as the STA received frames using the mechanism for which a setup is required
39	TIMEOUT	Requested from peer STA due to timeout
45	PEERKEY_MISMATCH	Peer STA does not support the requested cipher suite
46	PEER_INITIATED	In a DLS Teardown frame: The teardown was initiated by the DLS peer In a Disassociation frame: Disassociated because authorized access limit reached
47	AP_INITIATED	In a DLS Teardown frame: The teardown was initiated by the AP In a Disassociation frame: Disassociated due to external service requirements
48		Invalid FT Action frame count
49		Invalid pairwise master key identifier (PMKI)
50		Invalid MDE
51		Invalid FTE
52	MESH-PEERING-CANCELLED	SME cancels the mesh peering instance with the reason other than reaching the maximum number of peer mesh STAs
53	MESH-MAX-PEERS	The mesh STA has reached the supported maximum number of peer mesh STAs
54	MESH-CONFIGURATION-POLICY-VIOLATION	The received information violates the Mesh Configuration policy configured in the mesh STA profile
55	MESH-CLOSE-RCVD	The mesh STA has received a Mesh Peering Close message requesting to close the mesh peering.
56	MESH-MAX-RETRIES	The mesh STA has resent dot11MeshMaxRetries Mesh Peering Open messages, without receiving a Mesh Peering Confirm message.
57	MESH-CONFIRM-TIMEOUT	The confirmTimer for the mesh peering instance times out.
58	MESH-INVALID-GTK	The mesh STA fails to unwrap the GTK or the values in the wrapped contents do not match
59	MESH-INCONSISTENT-PARAMETERS	The mesh STA receives inconsistent information about the mesh parameters between Mesh Peering Management frames

Table 8-36—Reason codes *(continued)*

Reason code	Name	Meaning
60	MESH-INVALID-SECURITY-CAPABILITY	The mesh STA fails the authenticated mesh peering exchange because due to failure in selecting either the pairwise ciphersuite or group ciphersuite
61	MESH-PATH-ERROR-NO-PROXY-INFORMATION	The mesh STA does not have proxy information for this external destination.
62	MESH-PATH-ERROR-NO-FORWARDING-INFORMATION	The mesh STA does not have forwarding information for this destination.
63	MESH-PATH-ERROR-DESTINATION-UNREACHABLE	The mesh STA determines that the link to the next hop of an active path in its forwarding information is no longer usable.
64	MAC-ADDRESS-ALREADY-EXISTS-IN-MBSS	The Deauthentication frame was sent because the MAC address of the STA already exists in the mesh BSS. See 10.3.6.
65	MESH-CHANNEL-SWITCH-REGULATORY-REQUIREMENTS	The mesh STA performs channel switch to meet regulatory requirements.
66	MESH-CHANNEL-SWITCH-UNSPECIFIED	The mesh STA performs channel switch with unspecified reason.
67–65 535		Reserved

8.4.1.8 AID field

In infrastructure BSS operation, the AID field is a value assigned by an AP during association that represents the 16-bit ID of a STA. In mesh BSS operation, the AID field is a value that represents the 16-bit ID of a neighbor peer mesh STA. An AID value is assigned by a mesh STA that receives and accepts a Mesh Peering Open frame to the transmitter of the Mesh Peering Open frame during the mesh peering establishment process (see 13.3.1). The length of the AID field is 2 octets. The AID field is illustrated in Figure 8-42.

	Association ID (AID)
Octets:	2

Figure 8-42—AID field

The value assigned as the AID is in the range 1–2007 and is placed in the 14 LSBs of the AID field, with the two MSBs of the AID field set to 1 (see 8.2.4.2).

8.4.1.9 Status Code field

The Status Code field is used in a response management frame to indicate the success or failure of a requested operation. The length of the Status Code field is 2 octets. The Status Code field is illustrated in Figure 8-43.

	Status Code
Octets:	2

Figure 8-43—Status Code field

If an operation is successful, then the status code is set to 0. If an operation results in failure, the status code indicates a failure cause. The failure cause codes are defined in Table 8-37.

Table 8-37—Status codes

Status code	Name	Meaning
0	SUCCESS	Successful
1	REFUSED, REFUSED_REASON_UNSPECIFIED	Unspecified failure
2		TDLS wakeup schedule rejected but alternative schedule provided
3		TDLS wakeup schedule rejected
4		Reserved
5		Security disabled
6		Unacceptable lifetime
7		Not in same BSS
8–9		Reserved
10	REFUSED_CAPABILITIES_MISMATCH	Cannot support all requested capabilities in the Capability Information field
11		Reassociation denied due to inability to confirm that association exists
12		Association denied due to reason outside the scope of this standard
13		Responding STA does not support the specified authentication algorithm
14		Received an Authentication frame with authentication transaction sequence number out of expected sequence
15		Authentication rejected because of challenge failure
16		Authentication rejected due to timeout waiting for next frame in sequence
17		Association denied because AP is unable to handle additional associated STAs
18	REFUSED_BASIC_RATES_MISMATCH	Association denied due to requesting STA not supporting all of the data rates in the BSSBasicRateSet parameter
19		Association denied due to requesting STA not supporting the short preamble option
20		Association denied due to requesting STA not supporting the PBCC modulation option
21		Association denied due to requesting STA not supporting the Channel Agility option
22		Association request rejected because Spectrum Management capability is required
23		Association request rejected because the information in the Power Capability element is unacceptable
24		Association request rejected because the information in the Supported Channels element is unacceptable

Table 8-37—Status codes *(continued)*

Status code	Name	Meaning
25		Association denied due to requesting STA not supporting the Short Slot Time option
26		Association denied due to requesting STA not supporting the DSSS-OFDM option
27		Association denied because the requesting STA does not support HT features
28		R0KH unreachable
29		Association denied because the requesting STA does not support the phased coexistence operation (PCO) transition time required by the AP
30	REFUSED_TEMPORARILY	Association request rejected temporarily; try again later
31		Robust management frame policy violation
32		Unspecified, QoS-related failure
33		Association denied because QoS AP has insufficient bandwidth to handle another QoS STA
34		Association denied due to excessive frame loss rates and/or poor conditions on current operating channel
35		Association (with QoS BSS) denied because the requesting STA does not support the QoS facility
36		Reserved
37		The request has been declined
38	INVALID_PARAMETERS	The request has not been successful as one or more parameters have invalid values
39	REJECTED_WITH_SUGGESTED_CHANGES	The TS has not been created because the request cannot be honored; however, a suggested TSPEC is provided so that the initiating STA may attempt to set another TS with the suggested changes to the TSPEC
40		Invalid element, i.e., an element defined in this standard for which the content does not meet the specifications in Clause 8
41		Invalid group cipher
42		Invalid pairwise cipher
43		Invalid AKMP
44		Unsupported RSNE version
45		Invalid RSNE capabilities
46		Cipher suite rejected because of security policy
47	REJECTED_FOR_DELAY_PERIOD	The TS has not been created; however, the HC may be capable of creating a TS, in response to a request, after the time indicated in the TS Delay element
48	DLS_NOT_ALLOWED	Direct link is not allowed in the BSS by policy
49	NOT_PRESENT	The Destination STA is not present within this BSS
50	NOT_QOS_STA	The Destination STA is not a QoS STA

Table 8-37—Status codes *(continued)*

Status code	Name	Meaning
51		Association denied because the Listen Interval is too large
52		Invalid FT Action frame count
53		Invalid pairwise master key identifier (PMKID)
54		Invalid MDE
55		Invalid FTE
56		Requested TCLAS processing is not supported by the AP.
57		The AP has insufficient TCLAS processing resources to satisfy the request.
58		The TS has not been created because the request cannot be honored; however, the HC suggests the STA transitions to other BSSs to setup the TS.
59	GAS_ADVERTISEMENT_ PROTOCOL_NOT_SUPPORTED	GAS Advertisement Protocol not supported
60	NO_OUTSTANDING_GAS_ REQUEST	No outstanding GAS request
61	GAS_RESPONSE_NOT_ RECEIVED_FROM _SERVER	GAS Response not received from the Advertisement Server
62	GAS_QUERY_TIMEOUT	STA timed out waiting for GAS Query Response
63	GAS_QUERY_RESPONSE_ TOO_ LARGE	GAS Response is larger than query response length limit
64	REJECTED_HOME_WITH_ SUGGESTED_CHANGES	Request refused because home network does not support request
65	SERVER_UNREACHABLE	Advertisement Server in the network is not currently reachable
66		Reserved
67	REJECTED_FOR_SSP_ PERMISSIONS	Request refused due to permissions received via SSPN interface
68		Request refused because AP does not support unauthenticated access
69-71		Reserved
72		Invalid contents of RSNE
73		U-APSD Coexistence is not supported.
74		Requested U-APSD Coexistence mode is not supported.
75		Requested Interval/Duration value cannot be supported with U-APSD Coexistence.
76		Authentication is rejected because an Anti-Clogging Token is required.
77		Authentication is rejected because the offered finite cyclic group is not supported.

Table 8-37—Status codes *(continued)*

Status code	Name	Meaning
78	CANNOT_FIND_ALTERNATIVE_ TBTT	The TBTT adjustment request has not been successful because the STA could not find an alternative TBTT.
79	TRANSMISSION_FAILURE	Transmission failure
80	REQUESTED_TCLAS_NOT_ SUPPORTED	Requested TCLAS Not Supported.
81	TCLAS_RESOURCES_EXHAUSTED	TCLAS Resources Exhausted.
82	REJECTED_WITH_SUGGESTED_ BSS_TRANSITION	Rejected with Suggested BSS Transition.
83		Reserved
92	REFUSED_EXTERNAL_REASON	(Re)association refused for some external reason
93	REFUSED_AP_OUT_OF_MEMORY	(Re)association refused because of memory limits at the AP
94	REJECTED_EMERGENCY_ SERVICES_NOT_SUPPORTED	(Re)association refused because emergency services are not supported at the AP.
95	QUERY_RESPONSE_ OUTSTANDING	GAS query response not yet received.
96-99		Reserved
100	MCCAOP_RESERVATION_ CONFLICT	The request failed due to a reservation conflict
101	MAF_LIMIT_EXCEEDED	The request failed due to exceeded MAF limit
102	MCCA_TRACK_LIMIT_EXCEEDED	The request failed due to exceeded MCCA track limit
103–65 535		Reserved

8.4.1.10 Timestamp field

This field represents the value of the timing synchronization function (TSF) timer (see 10.1 and 10.21) of a frame's source. The length of the Timestamp field is 8 octets. The Timestamp field is illustrated in Figure 8-44.

	Timestamp
Octets:	8

Figure 8-44—Timestamp field

8.4.1.11 Action field

The Action field provides a mechanism for specifying extended management actions. The format of the Action field is shown in Figure 8-45.

	Category	Action Details
Octets:	1	variable

Figure 8-45—Action field

The Category field is set to one of the nonreserved values shown in Table 8-38. Action frames of a given category are referred to as *<category name> Action frames*. For example, frames in the QoS category are called *QoS Action frames*.

The Action Details field contains the details of the action. The details of the actions allowed in each category are described in the appropriate subclause referenced in Table 8-38.

Table 8-38—Category values

Code	Meaning	See subclause	Robust	Group addressed privacy
0	Spectrum management	8.5.2	Yes	No
1	QoS	8.5.3	Yes	No
2	DLS	8.5.4	Yes	No
3	Block Ack	8.5.5	Yes	No
4	Public	8.5.8	No	No
5	Radio measurement	8.5.7	Yes	No
6	Fast BSS Transition	8.5.9	Yes	No
7	HT	8.5.12	No	No
8	SA Query	8.5.10	Yes	No
9	Protected Dual of Public Action	8.5	Yes	No
10	WNM	8.5.14	Yes	No
11	Unprotected WNM	8.5.15	No	No
12	TDLS	8.5.13	— See NOTE	No
13	Mesh	8.5.17	Yes	Yes
14	Multihop	8.5.18	Yes	Yes
15	Self-protected	8.5.16	No	No
16	Reserved	—	—	—
17	Reserved (used by WFA)	—	—	—
18–125	Reserved	—	—	—
126	Vendor-specific Protected	8.5.6	Yes	No
127	Vendor-specific	8.5.6	No	No
128–255	Error	—	—	—
NOTE—TDLS Action fields are always transported encapsulated within a data frame (see 10.22.1), so the question of whether these frame are Robust is not applicable.				

8.4.1.12 Dialog Token field

The Dialog Token field is used for matching action responses with action requests when there are multiple, concurrent action requests. The length of the Dialog Token field is 1 octet. The Dialog Token field is illustrated in Figure 8-46. See 9.24.5.

	Dialog Token
Octets:	1

Figure 8-46—Dialog Token fixed field

8.4.1.13 DLS Timeout Value field

The DLS Timeout Value field is used in the DLS Request frame to indicate the timeout value for the direct link. The length of the DLS Timeout Value field is 2 octets. The DLS Timeout Value field is illustrated in Figure 8-47. See 10.7.2.4.

	DLS Timeout Value
Octets:	2

Figure 8-47—DLS Timeout Value fixed field

8.4.1.14 Block Ack Parameter Set field

The Block Ack Parameter Set field is used in ADDBA frames to signal the parameters for setting up a Block Ack. The length of the Block Ack Parameter Set field is 2 octets. The Block Ack Parameter Set field is illustrated in Figure 8-48.

	B0	B1	B2 B5	B6 B15
	A-MSDU Supported	Block Ack Policy	TID	Buffer Size
Bits:	1	1	4	10

Figure 8-48—Block Ack Parameter Set fixed field

The A-MSDU Supported subfield determines whether an A-MSDU may be carried in a QoS data MPDU sent under this Block Ack agreement. When equal to 1, use of A-MSDU is permitted. When equal to 0, use of A-MSDU is not permitted.

The Block Ack Policy subfield is set to 1 for immediate Block Ack and 0 for delayed Block Ack.

The TID subfield contains the value of the TC or TS for which the BlockAck is being requested.

The Buffer Size subfield indicates the number of buffers available for this particular TID.[21] When the A-MSDU Supported field is equal to 0 as indicated by the STA transmitting the Block Ack Parameter Set field, each buffer is capable of holding a number of octets equal to the maximum size of an MSDU. When the

[21] For buffer size, the recipient of data advertises a single scalar number that is the number of fragment buffers of the maximum MSDU or A-MSDU size (indicated by the A-MSDU Supported field) available for the Block Ack agreement that is being negotiated. Every buffered MPDU that is associated with this Block Ack agreement consumes one of these buffers regardless of whether the frame contains a whole MSDU (or a fragment thereof) or an A-MSDU. For example, ten maximum-size unfragmented MSDUs consumes the same amount of buffer space at the recipient as ten smaller fragments of a single MSDU of maximum size.

A-MSDU Supported field is equal to 1 as indicated by the STA, each buffer is capable of holding a number of octets equal to the maximum size of an A-MSDU that is supported by the STA.

In an ADDBA Request frame, the Buffer Size subfield is intended to provide guidance for the frame receiver to decide its reordering buffer size and is advisory only. If the Buffer Size subfield is equal to 0, it implies that the originator of the Block Ack has no information to specify its value.

In an ADDBA Response frame, when the Status Code field is equal to 0, the Buffer Size subfield is set to a value of at least 1.

8.4.1.15 Block Ack Timeout Value field

The Block Ack Timeout Value field is used in the ADDBA Request and Response frames to indicate the timeout value for Block Ack. The length of the Block Ack Timeout Value field is 2 octets. The Block Ack Timeout Value field is illustrated in Figure 8-49.

	Block Ack Timeout Value
Octets:	2

Figure 8-49—Block Ack Timeout Value fixed field

The Block Ack Timeout Value field contains the duration, in TUs, after which the Block Ack setup is terminated, if there are no frame exchanges (see 10.5.4) within this duration using this Block Ack agreement. A value of 0 disables the timeout.

8.4.1.16 DELBA Parameter Set field

The DELBA Parameter Set field is used in a DELBA frame to terminate an already setup Block Ack. The length of the DELBA Parameters field is 2 octets. The DELBA Parameters field is illustrated in Figure 8-50.

	B0 B10	B11	B12 B15
	Reserved	Initiator	TID
Bits:	11	1	4

Figure 8-50—DELBA Parameters fixed field

The Initiator subfield indicates if the originator or the recipient of the data is sending this frame. It is set to 1 to indicate the originator and is set to 0 to indicate the recipient. The TID subfield indicates the TSID or the UP for which the Block Ack has been originally set up.

8.4.1.17 QoS Info field

The QoS Info field is 1 octet in length and contains capability information bits. The contents of the field are dependent on whether the STA is contained within an AP.

The format of the QoS Info field, when sent by the AP, is defined in Figure 8-51.

	B0 B3	B4	B5	B6	B7
	EDCA Parameter Set Update Count	Q-Ack	Queue Request	TXOP Request	Reserved
Bits:	4	1	1	1	1

Figure 8-51—QoS Info field when sent by an AP

The EDCA Parameter Set Update Count subfield is described in 9.2.4.2.

APs set the Q-Ack subfield to 1 when dot11QAckOptionImplemented is true and set it to 0 otherwise.

APs set the Queue Request subfield to 1 if they can process a nonzero Queue Size subfield in the QoS Control field in QoS data frames and set it to 0 otherwise.

APs set the TXOP Request subfield to 1 if they can process a nonzero TXOP Duration Requested subfield in the QoS Control field in QoS data frames and set it to 0 otherwise.

The format of the QoS Info field, when sent by the non-AP STA, is defined in Figure 8-52.

	B0	B1	B2	B3	B4	B5 B6	B7
	AC_VO U-APSD Flag	AC_VI U-APSD Flag	AC_BK U-APSD Flag	AC_BE U-APSD Flag	Q-Ack	Max SP Length	More Data Ack
Bits:	1	1	1	1	1	1	1

Figure 8-52—QoS Info field when set by a non-AP STA

Each of the ACs U-APSD Flag subfields is 1 bit in length and set to 1 in (Re)Association Request frames to indicate that the corresponding AC (AC_BE, AC_BK, AC_VI, or AC_VO) is both trigger-enabled and delivery-enabled. It is set to 0 in (Re)Association Request frames to indicate that the corresponding AC is neither trigger-enabled nor delivery-enabled. A TSPEC as described in 10.2.1.5 is to be used to make a particular AC exclusively either trigger-enabled or delivery-enabled. These subfields are set to 0 when the APSD subfield in the Capability Information field is equal to 0.

Non-AP STAs set the Q-Ack subfield to 1 when dot11QAckOptionImplemented is true and set it to 0 otherwise.

The Max SP Length subfield is 2 bits in length and indicates the maximum number of total buffered MSDUs, A-MSDUs, and MMPDUs the AP may deliver to a STA during any SP triggered by the STA. This subfield is reserved when the APSD subfield in the Capability Information field is equal to 0. If the APSD subfield in the Capability Information field is equal to 1, the settings of the values in the Max SP Length subfield are defined in Table 8-39.

Table 8-39—Settings of the Max SP Length subfield

Bit 5	Bit 6	Usage
0	0	AP may deliver all buffered MSDUs, A-MSDUs, and MMPDUs.
1	0	AP may deliver a maximum of two MSDUs, A-MSDUs, and MMPDUs per SP.
0	1	AP may deliver a maximum of four MSDUs, A-MSDUs, and MMPDUs per SP.
1	1	AP may deliver a maximum of six MSDUs, A-MSDUs, and MMPDUs per SP.

Non-AP STAs set the More Data Ack subfield to 1 to indicate that they can process ACK frames with the More Data bit in the Frame Control field equal to 1 and remain in the Awake state. Non-AP STAs set the More Data Ack subfield to 0 otherwise. For APs, the More Data Ack subfield is reserved.

8.4.1.18 Measurement Pilot Interval field

The Measurement Pilot Interval field represents the number of time units (TUs) between target measurement pilot transmission times (TMPTTs). The length of the Measurement Pilot Interval field is 1 octet. The Measurement Pilot Interval field is illustrated in Figure 8-53.

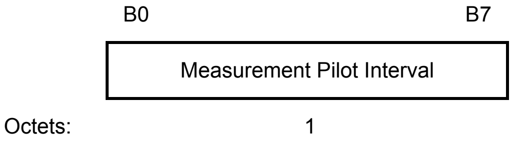


Figure 8-53—Measurement Pilot Interval fixed field

8.4.1.19 Max Transmit Power field

The Max Transmit Power field is a twos complement signed integer and is 1 octet in length, providing an upper limit, in units of dBm, on the transmit power as measured at the output of the antenna connector to be used by that AP on the current channel. See 10.11.13. The Max Transmit Power field is illustrated in Figure 8-54.

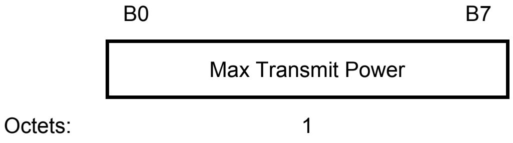


Figure 8-54—Max Transmit Power field

8.4.1.20 Transmit Power Used field

The Transmit Power Used field is a twos complement signed integer and is 1 octet in length. It is less than or equal to the Max Transmit Power and indicates the actual power used as measured at the output of the antenna connector, in units of dBm, by a STA when transmitting the frame containing the Transmit Power Used field. The Transmit Power Used value is determined anytime prior to sending the frame in which it is contained and has a tolerance of ±5 dB. The Transmit Power Used field is illustrated in Figure 8-55.

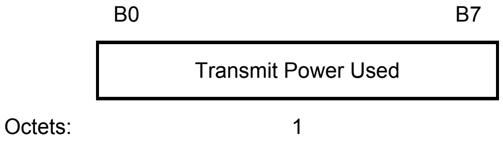


Figure 8-55—Transmit Power Used field

8.4.1.21 Channel Width field

The Channel Width field is used in a Notify Channel Width frame (see 8.5.12.2) to indicate the channel width on which the sending STA is able to receive. The length of the field is 1 octet. The Channel Width field is illustrated in Figure 8-56.

	Channel Width
Octets:	1

Figure 8-56—Channel Width fixed field

If a STA transmitting or receiving this field is operating in an operating class that includes a value of 13 or 14 in the behavior limits as specified in Annex E, then the values of the Channel Width field are defined in Table 8-40. If a STA transmitting or receiving this field is operating in an operating class that does not include a value of 13 or 14 in the behavior limits as specified in Annex E, then this field is reserved.

Table 8-40—Settings of the Channel Width field

Value	Meaning
0	20 MHz channel width
1	Any channel width in the STA's Supported Channel Width Set subfield
2–255	Reserved

8.4.1.22 SM Power Control field

The SM Power Control field is used in an SM Power Save frame (see 8.5.12.3) by a STA to communicate changes in its SM power-saving state. The field is 1 octet in length and is illustrated in Figure 8-57.

	B0	B1	B2 B7
	SM Power Save Enabled	SM Mode	Reserved
Bits:	1	1	6

Figure 8-57—SM Power Control fixed field

The SM Power Save Enabled subfield indicates whether SM power saving is enabled at the STA. A value of 1 indicates enabled, and a value of 0 indicates disabled.

The SM Mode subfield indicates the mode of operation. A value of 1 indicates dynamic SM power save mode, a value of 0 indicates static SM power save mode. The modes are described in 10.2.4.

8.4.1.23 PCO Phase Control field

The PCO Phase Control field is used in a Set PCO Phase frame (see 8.5.12.5) to indicate the phase of PCO operation (see 10.16). The length of the field is 1 octet. The PCO Phase Control field is illustrated in Figure 8-58.

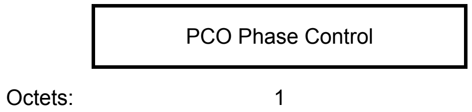


Figure 8-58—PCO Phase Control fixed field

The PCO Phase Control field indicates the current PCO phase. The values of the PCO Phase Control field are defined in Table 8-41.

Table 8-41—Settings of the PCO Phase Control field

Value	Meaning
0	20 MHz phase
1	40 MHz phase
2–255	Reserved

8.4.1.24 PSMP Parameter Set field

The PSMP Parameter Set field is used in a PSMP frame (see 8.5.12.4) to define the number of PSMP STA Info fields held in the PSMP frame, to indicate whether the PSMP sequence is to be followed by another PSMP sequence, and to indicate the duration of the PSMP sequence.

The PSMP Parameter Set field is 2 octets in length. The structure of the PSMP Parameter Set field is defined in Figure 8-59.

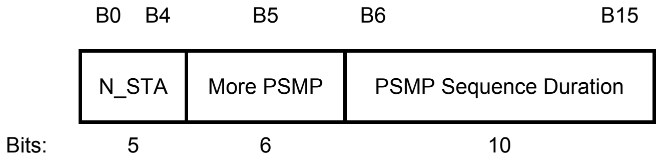


Figure 8-59—PSMP Parameter Set fixed field

The N_STA subfield indicates the number of STA Info fields present in the PSMP frame that contains the PSMP Parameter Set field.

The More PSMP subfield, when set to 1, indicates that the current PSMP sequence will be followed by another PSMP sequence. A value of 0 indicates that there will be no PSMP sequence following the current PSMP sequence.

The PSMP Sequence Duration subfield indicates the duration of the current PSMP sequence that is described by the PSMP frame, in units of 8 µs, relative to the end of the PSMP frame. Therefore, this field can describe a PSMP sequence with a duration of up to 8.184 ms. The next PSMP sequence within the current PSMP burst starts a SIFS interval after the indicated duration.

8.4.1.25 PSMP STA Info field

The PSMP STA Info field is used by the PSMP frame (see 8.5.12.4). The PSMP STA Info field defines the allocation of time to the downlink (PSMP-DTT) and/or uplink (PSMP-UTT) associated with a single RA.

There are two variants of the structure for the individually addressed and group addressed cases. The length of the PSMP STA Info field is 8 octets.

The structure of the STA Info field is defined in Figure 8-60 and Figure 8-61.

	B0 B1	B2 B12	B13 B20	B21 B63
	STA_INFO Type (=1)	PSMP-DTT Start Offset	PSMP-DTT Duration	PSMP Group Address ID
Bits:	2	11	8	43

Figure 8-60—PSMP STA Info fixed field (group addressed)

	B0 B1	B2 B12	B13 B20	B21 B36	B37 B47	B48 B57	B58 B63
	STA_INFO Type (=2)	PSMP-DTT Start Offset	PSMP-DTT Duration	STA_ID	PSMP-UTT Start Offset	PSMP-UTT Duration	Reserved
Bits:	2	11	8	16	11	10	6

Figure 8-61—PSMP STA Info fixed field (individually addressed)

The STA_INFO Type subfield indicates the format of the remainder of the structure. When the STA_INFO Type subfield is equal to 1, the PSMP STA Info field is structured as defined in Figure 8-60 and supports the transmission of group addressed data by the AP. When the STA_INFO Type subfield is equal to 2, the PSMP STA Info field is structured as defined in Figure 8-61 and supports the exchange of data with a single STA. STA_INFO Type subfield values 0 and 3 are reserved.

The PSMP-DTT Start Offset subfield indicates the start of the PSMP-DTT for the destination identified by the PSMP STA Info field, relative to the end of the PSMP frame, in units of 4 µs. This subfield locates the start of the first PPDU containing downlink data for this destination.

The PSMP-DTT Duration subfield indicates the duration of the PSMP-DTT for the destination identified by the PSMP STA Info field, in units of 16 µs. This subfield locates the end of the last PPDU containing downlink data for this destination relative to the PDMP-DTT start offset.

If no PSMP-DTT is scheduled for a STA, but a PSMP-UTT is scheduled for that STA, the PSMP-DTT Duration subfield is set to 0, and the PSMP-DTT Start Offset subfield is reserved.

The PSMP Group Address ID (B21 to B63) subfield contains the 43 least significant bits (LSBs) of a 48 bit MAC address. Use of this subfield is described in 9.26.1.8. B63 contains the LSB of the group address (considering the Individual/Group bit to be the most significant bit (MSB)).

The STA_ID subfield contains the AID of the STA to which the PSMP STA Info field is directed.

The PSMP-UTT Start Offset subfield indicates the start of the PSMP-UTT. The offset is specified relative to the end of the PSMP frame. It is specified in units of 4 µs. The first PSMP-UTT is scheduled to begin after a SIFS interval from the end of the last PSMP-DTT described in the PSMP.

The PSMP-UTT Duration subfield indicates the maximum length of a PSMP-UTT for a STA. PSMP-UTT duration is specified in units of 4 µs. All transmissions by the STA within the current PSMP sequence lie within the indicated PSMP-UTT.

If no PSMP-UTT is scheduled for a STA, but a PSMP-DTT is scheduled for that STA, the PSMP-UTT Start Offset and PSMP-UTT Duration subfields are both set to 0.

8.4.1.26 MIMO Control field

The MIMO Control field is used to manage the exchange of MIMO channel state or transmit beamforming feedback information. It is used in the CSI (see 8.5.12.6), Noncompressed Beamforming (see 8.5.12.7), and Compressed Beamforming (see 8.5.12.8) frames.

The MIMO Control field is 6 octets in length and is defined in Figure 8-62.

	B0B1	B2B3	B4	B5B6	B7B8	B9B10	B11B13	B14B15	B16B48
	Nc Index	Nr Index	MIMO Control Channel Width	Grouping (*Ng*)	Coefficient Size	Codebook Information	Remaining Matrix Segment	Reserved	Sounding Timestamp
Bits:	2	2	1	2	2	2	3	2	32

Figure 8-62—MIMO Control field

The subfields of the MIMO Control field are defined in Table 8-42.

Table 8-42—Subfields of the MIMO Control field

Subfield	Description
Nc Index	Indicates the number of columns in a matrix minus one: Set to 0 for *Nc* = 1 Set to 1 for *Nc* = 2 Set to 2 for *Nc* = 3 Set to 3 for *Nc* = 4
Nr Index	Indicates the number of rows in a matrix minus one: Set to 0 for *Nr* = 1 Set to 1 for *Nr* = 2 Set to 2 for *Nr* = 3 Set to 3 for *Nr* = 4
MIMO Control Channel Width	Indicates the width of the channel in which a measurement was made: Set to 0 for 20 MHz Set to 1 for 40 MHz
Grouping (*Ng*)	Number of carriers grouped into one: Set to 0 for *Ng* = 1 (No grouping) Set to 1 for *Ng* = 2 Set to 2 for *Ng* = 4 The value 3 is reserved

Table 8-42—Subfields of the MIMO Control field *(continued)*

Subfield	Description
Coefficient Size	Indicates the number of bits in the representation of the real and imaginary parts of each element in the matrix. For CSI feedback: Set to 0 for *Nb* = 4 Set to 1 for *Nb* = 5 Set to 2 for *Nb* = 6 Set to 3 for *Nb* = 8 For noncompressed beamforming feedback: Set 0 for *Nb* = 4 Set 1 for *Nb* = 2 Set 2 for *Nb* = 6 Set 3 for *Nb* = 8
Codebook Information	Indicates the size of codebook entries: Set to 0 for 1 bit for ψ , 3 bits for ϕ Set to 1 for 2 bits for ψ , 4 bits for ϕ Set to 2 for 3 bits for ψ , 5 bits for ϕ Set to 3 for 4 bits for ψ , 6 bits for ϕ
Remaining Matrix Segment	Contains the remaining segment number for the associated measurement report. Valid range: 0 to 7. Set to 0 for the last segment of a segmented report or the only segment of an unsegmented report.
Sounding Timestamp	Contains the lower 4 octets of the TSF timer value sampled at the instant that the MAC received the PHY-CCA.indication(IDLE) primitive that corresponds to the end of the reception of the sounding packet that was used to generate feedback information contained in the frame.

8.4.1.27 CSI Report field

The CSI Report field is used by the CSI frame (see 8.5.12.6) to carry explicit channel state information to a transmit beamformer, as described in 9.29.3.

The CSI Matrix subfields in the CSI Report field shown in Table 8-43 and Table 8-44 are matrices whose elements are taken from the CHAN_MAT parameter of RXVECTOR (see Table 20-1).

Table 8-43—CSI Report field (20 MHz)

Field	Size (bits)	Meaning
SNR in receive chain 1	8	Signal-to-noise ratio in the first receive chain of the STA sending the report.
...		
SNR in receive chain *Nr*	8	Signal-to-noise ratio in the *Nr*'th receive chain of the STA sending the report.
CSI Matrix for carrier –28	3+2× $Nb \times Nc \times Nr$	CSI matrix (see Figure 8-63)
...		

Table 8-43—CSI Report field (20 MHz) *(continued)*

Field	Size (bits)	Meaning
CSI Matrix for carrier –1	3+2× *Nb*×*Nc*×*Nr*	CSI matrix
CSI Matrix for carrier 1	3+2× *Nb*×*Nc*×*Nr*	CSI matrix
...		
CSI Matrix for carrier 28	3+2× *Nb*×*Nc*×*Nr*	CSI matrix

The structure of the field depends on the value of the MIMO Control Channel Width subfield. The CSI Report field for 20 MHz has the structure defined in Table 8-43 where

Nb is the number of bits determined by the Coefficients Size field of the MIMO Control field

Nc is the number of columns in a CSI matrix determined by the Nc Index field of the MIMO Control field

Nr is the number of rows in a CSI matrix determined by the Nr Index field of the MIMO Control field

The CSI Report field for 40 MHz has the structure defined in Table 8-44.

Table 8-44—CSI Report field (40 MHz)

Field	Size (bits)	Meaning
SNR in receive chain 1	8	Signal-to-noise ratio in the first receive chain of the STA sending the report.
...		
SNR in receive chain *Nr*	8	Signal-to-noise ratio in the *Nr*'th receive chain of the STA sending the report.
CSI Matrix for carrier –58	3+2× *Nb*×*Nc*×*Nr*	CSI matrix (see Figure 8-63)
...		
CSI Matrix for carrier –2	3+2× *Nb*×*Nc*×*Nr*	CSI matrix
CSI Matrix for carrier 2	3+2× *Nb*×*Nc*×*Nr*	CSI matrix
...		
CSI Matrix for carrier 58	3+2× *Nb*×*Nc*×*Nr*	CSI matrix

The signal-to-noise ratio (SNR) values in Table 8-43 and Table 8-44 are encoded as an 8-bit twos complement value of 4 × (SNR_average – 22), where SNR_average is the decibel representation of linearly averaged values over the tones represented. This encoding covers the SNR range from –10 dB to 53.75 dB in 0.25 dB steps.

Grouping is a method that reduces the size of the CSI Report field by reporting a single value for each group of *Ng* adjacent subcarriers. With grouping, the size of the CSI Report field is *Nr*×8+*Ns*×(3+2×*Nb*×*Nc*×*Nr)* bits, where the number of subcarriers sent, *Ns*, is a function of *Ng* and whether matrices for 40 MHz or 20 MHz are sent. The value of *Ns* and the specific carriers for which matrices are sent are shown in Table 8-45. If the size of the CSI Report field is not an integral multiple of 8 bits, up to 7 zeros are appended to the end of the report to make its size an integral multiple of 8 bits.

Table 8-45—Number of matrices and carrier grouping

BW	Grouping *Ng*	*Ns*	Carriers for which matrices are sent
20 MHz	1	56	All data and pilot carriers: –28, –27,…–2, –1, 1, 2,…27, 28
	2	30	–28,–26,–24,–22,–20,–18,–16,–14,–12,–10,–8,–6,–4,–2,–1, 1,3,5,7,9,11,13,15,17,19,21,23,25,27,28
	4	16	–28,–24,–20,–16,–12,–8,–4,–1,1,5,9,13,17,21,25,28
40 MHz	1	114	All data and pilot carriers: –58, –57, …, –3, –2, 2, 3,…, 57, 58
	2	58	–58,–56,–54,–52,–50,–48,–46,–44,–42,–40,–38,–36,–34,–32,–30, –28,–26,–24,–22,–20,–18,–16,–14,–12,–10,–8,–6,–4,–2, 2,4,6,8,10,12,14,16,18,20,22,24,26,28, 30,32,34,36,38,40,42,44,46,48,50,52,54,56,58
	4	30	–58,–54,–50,–46,–42,–38,–34,–30,–26,–22,–18,–14,–10,–6, –2, 2,6,10,14,18,22,26,30,34,38,42,46,50,54,58

The CSI matrix H_{eff} for a single carrier has the structure defined in Figure 8-63. The encoding rules for the elements of the H_{eff} matrix are given in 20.3.12.3.2.

```
For each subcarrier include
{
    Carrier Matrix Amplitude of 3 bits
    For each of Nr rows in each CSI matrix in order: (1, …, Nr)
    {
        Include Nc complex coefficients of CSI matrix $H_{eff}$ in order: (1, …, Nc );
        each element of $H_{eff}$ includes the real part of the element (Nb bits) and
        imaginary part of the element (Nb bits) in that order
    }

}
```

Figure 8-63—CSI matrix coding

When operating with a 40 MHz channel width, CSI feedback with a bandwidth of 20 MHz corresponds to the tones in the primary 20 MHz channel.

8.4.1.28 Noncompressed Beamforming Report field

The Noncompressed Beamforming Report field is used by the Noncompressed Beamforming frame to carry explicit feedback in the form of noncompressed beamforming feedback matrices *V* for use by a transmit beamformer to determine steering matrices Q, as described in 9.29.3 and 20.3.12.3.

The structure of the field is dependent on the value of the MIMO Control Channel Width subfield. The Noncompressed Beamforming Report field for 20 MHz has the structure defined in Table 8-46 where

Nb is the number of bits determined by the Coefficients Size field of the MIMO Control field

Nc is the number of columns in a beamforming feedback matrix determined by the Nc Index field of the MIMO Control field

Nr is the number of rows in a beamforming feedback matrix determined by the Nr Index field of the MIMO Control field

Table 8-46—Noncompressed Beamforming Report field (20 MHz)

Field	Size (bits)	Meaning
SNR for space-time stream 1	8	Average signal-to-noise ratio in the STA sending the report for space-time stream 1.
...		
SNR for space-time stream *Nc*	8	Average signal-to-noise ratio in the STA sending the report for space-time stream *Nc*.
Beamforming Feedback Matrix for carrier –28	2× *Nb*×*Nc*×*Nr*	Beamforming feedback matrix *V* (see Figure 8-64)
...		
Beamforming Feedback Matrix for carrier –1	2× *Nb*×*Nc*×*Nr*	Beamforming feedback matrix *V*
Beamforming Feedback Matrix for carrier 1	2× *Nb*×*Nc*×*Nr*	Beamforming feedback matrix *V*
...		
Beamforming Feedback Matrix for carrier 28	2× *Nb*×*Nc*×*Nr*	Beamforming feedback matrix *V*

The Noncompressed Beamforming Report field for 40 MHz has the structure defined in Table 8-47.

Table 8-47—Noncompressed Beamforming Report field (40 MHz)

Field	Size (bits)	Meaning
SNR for space-time stream 1	8	Average signal-to-noise ratio in the STA sending the report for space-time stream 1.
...		
SNR for space-time stream *Nc*	8	Average signal-to-noise ratio in the STA sending the report for space-time stream *Nc*.
Beamforming Feedback Matrix for carrier –58	2× *Nb*×*Nc*×*Nr*	Beamforming feedback matrix *V* (see Figure 8-64)
...		
Beamforming Feedback Matrix for carrier –2	2× *Nb*×*Nc*×*Nr*	Beamforming feedback matrix *V*
Beamforming Feedback Matrix for carrier 2	2× *Nb*×*Nc*×*Nr*	Beamforming feedback matrix *V*
...		
Beamforming Feedback Matrix for carrier 58	+2× *Nb*×*Nc*×*Nr*	Beamforming feedback matrix *V*

The SNR values in Table 8-46 and Table 8-47 are encoded as an 8-bit twos complement value of 4 × (SNR_average – 22), where SNR_average is the sum of the values of SNR per tone (in decibels) divided by the number of tones represented. This encoding covers the SNR range from –10 dB to 53.75 dB in 0.25 dB steps. The SNR in space-time stream *i* corresponds to the SNR associated with the column *i* of the beamforming feedback matrix *V*. Each SNR corresponds to the predicted SNR at beamformee when the beamformer applies the matrix *V*.

Grouping is a method that reduces the size of the Noncompressed Beamforming Report field by reporting a single value for each group of *Ng* adjacent subcarriers. With grouping, the size of the Noncompressed Beamforming Report field is $Nc\times 8+Ns\times(2\times Nb\times Nc\times Nr)$ bits. The number of subcarriers sent, *Ns*, is a function of *Ng* and whether matrices for 40 MHz or 20 MHz are sent. The value of *Ns* and the specific carriers for which matrices are sent is shown in Table 8-45. If the size of the Noncompressed Beamforming Report field is not an integral multiple of 8 bits, up to 7 zeros are appended to the end of the report to make its size an integral multiple of 8 bits.

A beamforming feedback matrix *V* for a single carrier has the structure defined in Figure 8-64.

```
For each subcarrier include
{
For each of Nr rows in the
        Noncompressed beamforming feedback matrix in order: (1, …, Nr)
    {
        Include Nc complex coefficients of the Noncompressed beamforming feedback
        matrix V in order: (1, …, Nc  );  each element of V includes the real
        part of the element (Nb bits) and imaginary part of the element (Nb bits)
        in that order
    }
}
```

Figure 8-64—*V* matrix coding (noncompressed beamforming)

Encoding rules for elements of the *V* matrix are given in 20.3.12.3.5.

When operating with a 40 MHz channel width, noncompressed feedback with a bandwidth of 20 MHz corresponds to the tones in the primary 20 MHz channel.

8.4.1.29 Compressed Beamforming Report field

The Compressed Beamforming Report field is used by the Compressed Beamforming frame (see 8.5.12.8) to carry explicit feedback information in the form of angles representing compressed beamforming feedback matrices *V* for use by a transmit beamformer to determine steering matrices Q, as described in 9.29.3 and 20.3.12.3.

The size of the Compressed Beamforming Report field depends on the values in the MIMO Control field.

The Compressed Beamforming Report field contains the channel matrix elements indexed, first, by matrix angles in the order shown in Table 8-48 and, second, by data subcarrier index from lowest frequency to highest frequency. The explanation on how these angles are generated from the beamforming feedback matrix *V* is given in 20.3.12.3.6.

Table 8-48—Order of angles in the Compressed Beamforming Report field

Size of *V* (*Nr* × *Nc*)	Number of angles (*Na*)	The order of angles in the Quantized Beamforming Feedback Matrices Information field
2×1	2	ϕ11, ψ21
2×2	2	ϕ11, ψ21
3×1	4	ϕ11, ϕ21, ψ21, ψ31

Table 8-48—Order of angles in the Compressed Beamforming Report field *(continued)*

Size of *V* (*Nr* × *Nc*)	Number of angles (*Na*)	The order of angles in the Quantized Beamforming Feedback Matrices Information field
3×2	6	ϕ11, ϕ21, ψ21, ψ31, ϕ22, ψ32
3×3	6	ϕ11, ϕ21, ψ21, ψ31, ϕ22, ψ32
4×1	6	ϕ11, ϕ21, ϕ31, ψ21, ψ31, ψ41
4×2	10	ϕ11, ϕ21, ϕ31, ψ21, ψ31, ψ41, ϕ22, ϕ32, ψ32, ψ42
4×3	12	ϕ11, ϕ21, ϕ31, ψ21, ψ31, ψ41, ϕ22, ϕ32, ψ32, ψ42, ϕ33, ψ43
4×4	12	ϕ11, ϕ21, ϕ31, ψ21, ψ31, ψ41, ϕ22, ϕ32, ψ32, ψ42, ϕ33, ψ43

The angles are quantized as defined in Table 8-49. All angles are transmitted LSB to MSB.

Table 8-49—Quantization of angles

Quantized ψ	Quantized ϕ
$\psi = \frac{k\pi}{2^{b_\psi + 1}} + \frac{\pi}{2^{b_\psi + 2}}$ radians where $k = 0, 1, \ldots, 2^{b_\psi} - 1$ b_ψ is the number of bits used to quantize ψ (defined by the Codebook Information field of the MIMO Control field; see 8.4.1.26);	$\phi = \frac{k\pi}{2^{b_\phi - 1}} + \frac{\pi}{2^{b_\phi}}$ radians where $k = 0, 1, \ldots, 2^{b_\phi} - 1$ b_ϕ is the number of bits used to quantize ϕ (defined by the Codebook Information field of the MIMO Control field; see 8.4.1.26)

The Compressed Beamforming Report field for 20 MHz has the structure defined in Table 8-50, where *Na* is the number of angles used for beamforming feedback matrix *V* (see Table 8-48).

Table 8-50—Compressed Beamforming Report field (20 MHz)

Field	Size (bits)	Meaning
SNR in space-time stream 1	8	Average signal-to-noise ratio in the STA sending the report for space-time stream 1
...		
SNR in space-time stream *Nc*	8	Average signal-to-noise ratio in the STA sending the report for space-time stream *Nc*
Beamforming Feedback Matrix *V* for carrier –28	$Na \times (b_\psi + b_\phi)/2$	Beamforming feedback matrix *V*
...		
Beamforming Feedback Matrix *V* for carrier –1	$Na \times (b_\psi + b_\phi)/2$	Beamforming feedback matrix *V*

Table 8-50—Compressed Beamforming Report field (20 MHz) *(continued)*

Field	Size (bits)	Meaning
Beamforming Feedback Matrix *V* for carrier 1	$Na\times(b_\psi+b_\phi)/2$	Beamforming feedback matrix *V*
...		
Beamforming Feedback Matrix *V* for carrier 28	$Na\times(b_\psi+b_\phi)/2$	Beamforming feedback matrix *V*

The Compressed Beamforming Report field for 40 MHz has the structure defined in Table 8-51, where *Na* is the number of angles used for beamforming feedback matrix *V* (see Table 8-48).

Table 8-51—Compressed Beamforming Report field (40 MHz)

Field	Size (bit)	Meaning
SNR in space-time stream 1	8	Average signal-to-noise ratio in the STA sending the report for space-time stream 1
...		
SNR in space-time stream *Nc*	8	Average signal-to-noise ratio in the STA sending the report for space-time stream *Nc*
Beamforming Feedback Matrix *V* for carrier –58	$Na\times(b_\psi+b_\phi)/2$	Beamforming feedback matrix *V*
Beamforming Feedback Matrix *V* for carrier –58 + *Ng*	$Na\times(b_\psi+b_\phi)/2$	Beamforming feedback matrix *V*
...		
Beamforming Feedback Matrix *V* for carrier –2	$Na\times(b_\psi+b_\phi)/2$	Beamforming feedback matrix *V*
Beamforming Feedback Matrix *V* for carrier 2	$Na\times(b_\psi+b_\phi)/2$	Beamforming feedback matrix *V*
Beamforming Feedback Matrix *V* for carrier 2 + *Ng*	$Na\times(b_\psi+b_\phi)/2$	Beamforming feedback matrix *V*
...		
Beamforming Feedback Matrix *V* for carrier 58	$Na\times(b_\psi+b_\phi)/2$	Beamforming feedback matrix *V*

The SNR values in Table 8-50 and Table 8-51 are encoded as an 8-bit twos complement value of 4 × (SNR_average – 22), where SNR_average is the sum of the values of SNR per tone (in decibels) divided by the number of tones represented. This encoding covers the SNR range from –10 dB to 53.75 dB in 0.25 dB steps. Each SNR value per tone in stream *i* (before being averaged) corresponds to the SNR associated with the column *i* of the beamforming feedback matrix *V* determined at the beamformee. Each SNR corresponds to the predicted SNR at the beamformee when the beamformer applies the matrix *V*.

Grouping is a method that reduces the size of the Compressed Beamforming Report field by reporting a single value for each group of *Ng* adjacent subcarriers. With grouping, the size of the Compressed Beamforming Report field is $Nc\times 8+Ns\times(Na\times(b_\psi+b_\phi)/2)$ bits, where the number of subcarriers sent, *Ns*, is a function of *Ng* and whether matrices for 40 MHz or 20 MHz are sent. The value of *Ns* and the specific carriers for which matrices are sent is defined in Table 8-45. If the size of the Compressed Beamforming Report field is not an integral multiple of 8 bits, up to 7 zeros are appended to the end of the report to make its size an integral multiple of 8 bits.

See Figure 8-65 and Figure 8-66 for examples of this encoding.

Bits	b1..b5	b6..b8	b9..b13	b14..b16	…	b441..b445	b446..b448
Data	$\phi_{11}(-28)$	$\psi_{21}(-28)$	$\phi_{11}(-27)$	$\psi_{21}(-27)$	…	$\phi_{11}(28)$	$\psi_{21}(28)$

Conditions:

- 2 x 2 V
- $b_\psi = 3$, $b_\phi = 5$
- No grouping
- 20 MHz width
- The matrix *V* is encoded using 8 bits per tone.

Figure 8-65—First example of Compressed Beamforming Report field encoding

Bits	b1..b4	b5..b8	…	b27..b28	b29..b30	b31..b34	…	b59..b60	…	b871..b874	…	b899..b900
Data	$\phi_{11}(-58)$	$\phi_{21}(-58)$	…	$\psi_{32}(-58)$	$\psi_{42}(-58)$	$\phi_{11}(-54)$	…	$\psi_{42}(-54)$	…	$\phi_{11}(58)$	…	$\psi_{42}(58)$

Conditions:

- 4 x 2 V
- $b_\psi = 2$, $b_\phi = 4$
- 4 tone grouping
- 40 MHz width
- The matrix *V* is encoded using 30 bits per tone.

Figure 8-66—Second example of Compressed Beamforming Report field encoding

When operating with a 40 MHz channel width, compressed feedback with a bandwidth of 20 MHz corresponds to the tones in the primary 20 MHz channel.

8.4.1.30 Antenna Selection Indices field

The Antenna Selection Indices field is used within the Antenna Selection Indices Feedback frame to carry ASEL feedback, as described in 9.30.

The Antenna Selection Indices field is 1 octet in length and illustrated in Figure 8-67.

	Antenna Selection Indices
Octets:	1

Figure 8-67—Antenna Selection Indices fixed field

Bits 0 to 7 in the Antenna Selection Indices field correspond to antennas with indices 0 to 7, respectively. A value of 1 in a bit indicates the corresponding antenna is selected, and the value of 0 indicates the corresponding antenna is not selected.

8.4.1.31 Organization Identifier field

The Organization Identifier field shall contain a public organizationally unique identifier assigned by the IEEE. The order of the Organization Identifier field is described in 8.2.2. The IEEE has assigned public organizationally unique identifiers both of 24-bit length (OUI) and longer length. In the latter case specific OUI values are shared over multiple organizations, e.g., using 36-bit length identifiers (OUI-36 and IAB) (see IEEE Registration Authority [B19]). The length of the Organization Identifier field (*j*) shall be the minimum number of octets required to contain the entire organizationally unique identifier (see Figure 8-68), and the first 3 octets shall contain the OUI portion of the identifier. Thus, the Organization Identifier field is 3 octets in length if the organizationally unique identifier is an OUI, or 5 octets in length if the organizationally unique identifier is 36 bits in length. The IEEE assigns 36-bit organizationally unique identifiers such that the OUI portion indicates that total length of the identifier is 36 bits.

	Organization Identifier
Octets:	j

Figure 8-68—Organization Identifier field

If the length of the organizationally unique identifier is not an integral number of octets, the least significant bits of the last octet are specified by the organization identified.

NOTE—For example, for the organizationally unique identifier 0x0050C24A4, the Organization Identifier field would contain 0x0050C24A4*y* where *y* represents the four least significant bits of the fifth octet of the field. The value of *y* is specified by the organization whose identifier is 0x0050C24A4.

8.4.1.32 Rate Identification field

The Rate Identification field is 4 octets in length and contains the rate identification information for a frame that is not the current frame transmitted or received by a STA. This information allows services to exchange frame rate information prior to use of the frames that use the rate specified by the Rate Identification field. The contents of the field is defined in Figure 8-69.

	Mask	MCS Index	Rate
Octets:	1	1	2

Figure 8-69—Identification field format

The Mask field specifies which other fields in the Rate Identification field are used by a STA. The format of the Mask field is shown in Figure 8-70.

	B0 B2	B3 B4	B5 B7
	MCS Selector	Rate Type	Reserved
Bits:	3	2	3

Figure 8-70—Mask field format

The MCS Selector field set to 0 indicates the MCS Index field is reserved. The MCS Selector field set to 1 indicates the MCS Index field specifies an index value that is taken from Table 20-30 through Table 20-33 and Table 20-39 through Table 20-41 in 20.6. The MCS Selector field set to 2 indicates the MCS Index field specifies an index value that is taken from Table 20-34 through Table 20-38 and Table 20-43 through Table 20-44 in 20.6. The MCS Selector field values 3 to 7 are reserved.

The Rate Type field set to 0 indicates the Rate field is reserved. The Rate Type field set to 1 indicates the Rate field specifies a data rate that is in the basic rate set. The Rate Type field set to 2 indicates the Rate field specifies a data rate that is not in the basic rate set.

The MCS Index field is a 1 octet unsigned integer that specifies the row index for one of the MCS parameter tables in 20.6.

The Rate field contains a 2-octet unsigned integer that specifies the PHY rate in 0.5 Mb/s units.

8.4.1.33 GAS Query Response Fragment ID field

A GAS Query Fragment Response ID field is used by the STA to indicate when a GAS Query Response spans multiple MMPDUs. STAs responding to GAS request use this field to inform the requesting STA of the GAS fragment number of the transmitted frames as well as identifying the last GAS fragment of the Query Response. Requesting STAs use this field to determine if any fragments of the Query Response are missing. The maximum value permitted in the GAS Query Response Fragment ID is 127. The More GAS Fragments field is set to 1 in GAS Query Response fragments of GAS Comeback Response frames that have another GAS fragment of the current query response to follow; otherwise, it is set to 0. The format of GAS Query Response Fragment ID is shown in Figure 8-71.

	B0 B6	B7
	GAS Query Response Fragment ID	More GAS Fragments
Bits:	7	1

Figure 8-71—GAS Query Response Fragment ID field

8.4.1.34 Venue Info field

The Venue Info field is a 2-octet field. It contains Venue Group and Venue Type subfields. The format of Venue Info subfield is shown in Figure 8-72.

	Venue Group	Venue Type
Octets:	1	1

Figure 8-72—Venue Info field format

The Venue Group and Venue Type subfields are both 1-octet values selected from Table 8-52 and Table 8-53, respectively. The entries in Table 8-52 and Table 8-53 are drawn from the International Building Code's Use and Occupancy Classifications [B43].

Table 8-52—Venue group codes and descriptions

Venue group code	Venue group description
0	Unspecified
1	Assembly
2	Business
3	Educational
4	Factory and Industrial
5	Institutional
6	Mercantile
7	Residential
8	Storage
9	Utility and Miscellaneous
10	Vehicular
11	Outdoor
12–255	Reserved

Table 8-53—Venue type assignments

Venue group code	Venue type code	Venue description
0	0	Unspecified
0	1–255	Reserved
1	0	Unspecified Assembly
1	1	Arena
1	2	Stadium
1	3	Passenger Terminal (e.g., airport, bus, ferry, train station)
1	4	Amphitheater
1	5	Amusement Park
1	6	Place of Worship
1	7	Convention Center
1	8	Library
1	9	Museum
1	10	Restaurant
1	11	Theater
1	12	Bar
1	13	Coffee Shop
1	14	Zoo or Aquarium

Table 8-53—Venue type assignments *(continued)*

Venue group code	Venue type code	Venue description
1	15	Emergency Coordination Center
1	16–255	Reserved
2	0	Unspecified Business
2	1	Doctor or Dentist office
2	2	Bank
2	3	Fire Station
2	4	Police Station
2	6	Post Office
2	7	Professional Office
2	8	Research and Development Facility
2	9	Attorney Office
2	10–255	Reserved
3	0	Unspecified Educational
3	1	School, Primary
3	2	School, Secondary
3	3	University or College
3	4–255	Reserved
4	0	Unspecified Factory and Industrial
4	1	Factory
4	2–255	Reserved
5	0	Unspecified Institutional
5	1	Hospital
5	2	Long-Term Care Facility (e.g., Nursing home, Hospice, etc.)
5	3	Alcohol and Drug Rehabilitation Center
5	4	Group Home
5	5	Prison or Jail
5	6–255	Reserved
6	0	Unspecified Mercantile
6	1	Retail Store
6	2	Grocery Market
6	3	Automotive Service Station
6	4	Shopping Mall
6	5	Gas Station
6	6–255	Reserved
7	0	Unspecified Residential
7	1	Private Residence
7	2	Hotel or Motel

Table 8-53—Venue type assignments *(continued)*

Venue group code	Venue type code	Venue description
7	3	Dormitory
7	4	Boarding House
7	5–255	Reserved
8	0	Unspecified Storage
8	1–255	Reserved
9	0	Unspecified Utility and Miscellaneous
9	1–255	Reserved
10	0	Unspecified Vehicular
10	1	Automobile or Truck
10	2	Airplane
10	3	Bus
10	4	Ferry
10	5	Ship or Boat
10	6	Train
10	7	Motor Bike
10	8–255	Reserved
11	0	Unspecified Outdoor
11	1	Muni-mesh Network
11	2	City Park
11	3	Rest Area
11	4	Traffic Control
11	5	Bus Stop
11	6	Kiosk
11	7–255	Reserved

8.4.1.35 Target Channel

The Target Channel field specifies the channel number of the target channel. The length of the Target Channel field is 1 octet. The Target Channel Field is illustrated in Figure 8-73.

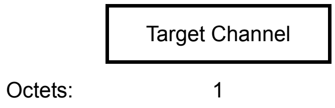


Figure 8-73—Target Channel field format

8.4.1.36 Operating Class

The Operating Class field specifies the operating class for the channel field included in the same frame. The length of the Operating Class field is 1 octet. Operating classes are defined in Annex E. The Operating Class field is illustrated in Figure 8-74.

	Operating Class
Octets:	1

Figure 8-74—Operating Channel field format

8.4.1.37 Send-Confirm field

The Send-Confirm field is used with SAE authentication as an anti-replay counter as specified in 11.3. See Figure 8-75.

	Send-Confirm
Octets:	2

Figure 8-75—Send-Confirm field

8.4.1.38 Anti-Clogging Token field

The Anti-Clogging Token field is used with SAE authentication for denial-of-service protection as specified in 11.3. See Figure 8-76.

	Anti-Clogging Token
Octets:	variable

Figure 8-76—Anti-Clogging Token field

8.4.1.39 Scalar field

The Scalar field is used with SAE authentication to communicate cryptographic material as specified in 11.3. See Figure 8-77.

	Scalar
Octets:	variable

Figure 8-77—Scalar field

8.4.1.40 Element field

The Element field is used with SAE authentication to communicate an element in a finite field as specified in 11.3. See Figure 8-78.

	Element
Octets:	variable

Figure 8-78—Element field

8.4.1.41 Confirm field

The Confirm field is used with SAE authentication to authenticate and prove possession of a cryptographic key as specified in 11.3. See Figure 8-79.

	Confirm
Octets:	variable

Figure 8-79—Confirm field

8.4.1.42 Finite Cyclic Group field

The Finite Cyclic Group is used in SAE to indicate which cryptographic group to use in the SAE exchange as specified in 11.3. See Figure 8-80.

	Finite Cyclic Group
Octets:	2

Figure 8-80—Finite Cyclic Group field

8.4.2 Information elements

8.4.2.1 General

Elements are defined to have a common general format consisting of a 1 octet Element ID field, a 1 octet Length field, and a variable-length element-specific Information field. Each element is assigned a unique Element ID as defined in this standard. The Length field specifies the number of octets in the Information field. See Figure 8-81.

	Element ID	Length	Information
Octets:	1	1	variable

Figure 8-81—Element format

The set of valid elements is defined in Table 8-54.

Table 8-54—Element IDs

Element	Element ID	Length of indicated element (in octets)	Extensible
SSID (see 8.4.2.2)	0	2 to 34	
Supported rates (see 8.4.2.3)	1	3 to 10	
FH Parameter Set (see 8.4.2.4)	2	7	
DSSS Parameter Set (see 8.4.2.5)	3	3	
CF Parameter Set (see 8.4.2.6)	4	8	
TIM (see 8.4.2.7)	5	6 to 256	
IBSS Parameter Set (see 8.4.2.8)	6	4	
Country (see 8.4.2.10)	7	8 to 256	
Hopping Pattern Parameters (see 8.4.2.11)	8	4	
Hopping Pattern Table (see 8.4.2.12)	9	6 to 256	
Request (see 8.4.2.13)	10	2 to 256	
BSS Load (see 8.4.2.30)	11	7	
EDCA Parameter Set (see 8.4.2.31)	12	20	
TSPEC (see 8.4.2.32)	13	57	
TCLAS (see 8.4.2.33)	14	2 to 257	
Schedule (see 8.4.2.36)	15	16	
Challenge text (see 8.4.2.9)	16	3 to 255	
Reserved	17–31		
Power Constraint (see 8.4.2.16)	32	3	
Power Capability (see 8.4.2.17)	33	4	
TPC Request (see 8.4.2.18)	34	2	
TPC Report (see 8.4.2.19)	35	4	
Supported Channels (see 8.4.2.20)	36	4 to 256	

Table 8-54—Element IDs *(continued)*

Element	Element ID	Length of indicated element (in octets)	Extensible
Channel Switch Announcement (see 8.4.2.21)	37	5	
Measurement Request (see 8.4.2.23)	38	5 to 257	Subelements, for formats using 8.4.2.23.4 to 8.4.2.23.12.
Measurement Report (see 8.4.2.24)	39	5 to 257	Subelements, for formats using 8.4.2.24.4 to 8.4.2.24.11.
Quiet (see 8.4.2.25)	40	8	
IBSS DFS (see 8.4.2.26)	41	10 to 255	
ERP (see 8.4.2.14)	42	3	
TS Delay (see 8.4.2.34)	43	6	
TCLAS Processing (see 8.4.2.35)	44	3	
HT Capabilities (see 8.4.2.58)	45	28	Yes
QoS Capability (see 8.4.2.37)	46	3	
Reserved	47		
RSN (see 8.4.2.27)	48	36 to 256	
Reserved	49		
Extended Supported Rates (see 8.4.2.15)	50	3 to 257	
AP Channel Report (see 8.4.2.38)	51	3 to 257	
Neighbor Report (see 8.4.2.39)	52	15 to 257	Subelements
RCPI (see 8.4.2.40)	53	3	Yes
Mobility Domain (MDE) (see 8.4.2.49)	54	5	
Fast BSS Transition (FTE) (see 8.4.2.50)	55	84 to 257	
Timeout Interval (see 8.4.2.51)	56	7	
RIC Data (RDE) (see 8.4.2.52)	57	6	
DSE Registered Location (see 8.4.2.54)	58	22	
Supported Operating Classes (see 8.4.2.56)	59	4 to 255	
Extended Channel Switch Announcement (see 8.4.2.55)	60	6	
HT Operation (see 8.4.2.59)	61	24	Yes
Secondary Channel Offset (see 8.4.2.22)	62	3	
BSS Average Access Delay (see 8.4.2.41)	63	3	Yes
Antenna (see 8.4.2.42)	64	3	Yes
RSNI (see 8.4.2.43)	65	3	Yes
Measurement Pilot Transmission (see 8.4.2.44)	66	3 to 257	Subelements

Table 8-54—Element IDs *(continued)*

Element	Element ID	Length of indicated element (in octets)	Extensible
BSS Available Admission Capacity (see 8.4.2.45)	67	4 to 28	Yes
BSS AC Access Delay (see 8.4.2.46)	68	6	Yes
Time Advertisement (see 8.4.2.63)	69	3 to 18	Yes
RM Enabled Capabilities (see 8.4.2.47)	70	7	Yes
Multiple BSSID (see 8.4.2.48)	71	3 to 257	Subelements
20/40 BSS Coexistence (see 8.4.2.62)	72	3	Yes
20/40 BSS Intolerant Channel Report (see 8.4.2.60)	73	3 to 257	
Overlapping BSS Scan Parameters (see 8.4.2.61)	74	16	
RIC Descriptor (see 8.4.2.53)	75	3 to 257	
Management MIC (see 8.4.2.57)	76	18	
Event Request (see 8.4.2.69)	78	5 to 257	Subelements
Event Report (see 8.4.2.70)	79	5 to 257	
Diagnostic Request (see 8.4.2.71)	80	6 to 257	Subelements
Diagnostic Report (see 8.4.2.72)	81	5 to 257	Subelements
Location Parameters (see 8.4.2.73)	82	2 to 257	Subelements
Nontransmitted BSSID Capability (see 8.4.2.74)	83	4	
SSID List (see 8.4.2.75)	84	2 to 257	
Multiple BSSID-Index (see 8.4.2.76)	85	3 to 5	
FMS Descriptor (see 8.4.2.77)	86	3 to 257	
FMS Request (see 8.4.2.78)	87	3 to 257	Subelements
FMS Response (see 8.4.2.79)	88	18 to 257	Subelements
QoS Traffic Capability (see 8.4.2.80)	89	3 to 5	Yes
BSS Max Idle Period (see 8.4.2.81)	90	5	Yes
TFS Request (see 8.4.2.82)	91	6 to 257	Subelements
TFS Response (see 8.4.2.83)	92	6 to 256	Subelements
WNM-Sleep Mode (see 8.4.2.84)	93	6	Yes
TIM Broadcast Request (see 8.4.2.85)	94	3	Yes
TIM Broadcast Response (see 8.4.2.86)	95	3 or 12	Yes
Collocated Interference Report (see 8.4.2.87)	96	23	Yes
Channel Usage (see 8.4.2.88)	97	3 to 257	Subelements
Time Zone (see 8.4.2.89)	98	3 to 257	Yes
DMS Request (see 8.4.2.90)	99	3 to 257	Subelements
DMS Response (see 8.4.2.91)	100	3 to 257	Subelements
Link Identifier (see 8.4.2.64)	101	20	Yes

Table 8-54—Element IDs *(continued)*

Element	Element ID	Length of indicated element (in octets)	Extensible
Wakeup Schedule (see 8.4.2.65)	102	20	Yes
Channel Switch Timing (see 8.4.2.66)	104	6	Yes
PTI Control (see 8.4.2.67)	105	5	Yes
TPU Buffer Status (see 8.4.2.68)	106	3	Yes
Interworking (see 8.4.2.94)	107	3, 5, 9, 11	
Advertisement Protocol (see 8.4.2.95)	108	variable	
Expedited Bandwidth Request (see 8.4.2.96)	109	3	
QoS Map Set (see 8.4.2.97)	110	18 to 60	Yes
Roaming Consortium (see 8.4.2.98)	111	variable	Yes
Emergency Alert Identifier (see 8.4.2.99)	112	10	
Mesh Configuration (see 8.4.2.100	113	9	Yes
Mesh ID (see 8.4.2.101	114	2 to 34	
Mesh Link Metric Report (see 8.4.2.102)	115	3 to 257	
Congestion Notification (see 8.4.2.103)	116	16	Yes
Mesh Peering Management (see 8.4.2.104)	117	5, 7, 9, 21, 23, or 25	Yes
Mesh Channel Switch Parameters (see 8.4.2.105)	118	8	Yes
Mesh Awake Window (see 8.4.2.106)	119	4	Yes
Beacon Timing (see 8.4.2.107)	120	3 to 255	
MCCAOP Setup Request (see 8.4.2.108)	121	8	Yes
MCCAOP Setup Reply (see 8.4.2.109)	122	4 or 9	
MCCAOP Advertisement (see 8.4.2.111)	123	4 to 257	Yes
MCCAOP Teardown (see 8.4.2.112)	124	3 or 9	Yes
GANN (see 8.4.2.113)	125	17	Yes
RANN (see 8.4.2.114)	126	23	Yes
Extended Capabilities (see 8.4.2.29)	127	3 to 8	Yes
Reserved	128–129		
PREQ (see 8.4.2.115)	130	39 to 254	Yes
PREP (see 8.4.2.116)	131	33 or 39	Yes
PERR (see 8.4.2.117)	132	17 to 251	Yes
Reserved	133–136		
PXU (see 8.4.2.118)	137	21 to 257	Yes
PXUC (see 8.4.2.119)	138	9	Yes
Authenticated Mesh Peering Exchange (see 8.4.2.120)	139	86 to 257	
MIC (see 8.4.2.121)	140	18	
Destination URI (see 8.4.2.92)	141	3 to 257	Yes

Table 8-54—Element IDs *(continued)*

Element	Element ID	Length of indicated element (in octets)	Extensible
U-APSD Coexistence (see 8.4.2.93)	142	14 to 257	Subelements
Reserved	143–173		
MCCAOP Advertisement Overview (see 8.4.2.110)	174	8	Yes
Reserved	175–220		
Vendor Specific (see 8.4.2.28)	221	3 to 257	
Reserved	222–255		

The frame body components specified for many management subtypes result in elements ordered by ascending Element ID, with the exception of the MIC Management element (8.4.2.57). If present, the MIC Management element appears at the end of the robust management frame body. See 9.24.6 on the parsing of elements.

A "Yes" in the Extensible column of an element listed in Table 8-54 indicates that the Length of the element might be extended in future revisions or amendments of this standard. See 9.24.8. When the Extensible column of an element is set to "Subelements," then the element might be extended in future revisions or amendments of this standard by defining additional subelements. See 9.24.9.

8.4.2.2 SSID element

The SSID element indicates the identity of an ESS or IBSS. See Figure 8-82.

	Element ID	Length	SSID
Octets:	1	1	0–32

Figure 8-82—SSID element format

The length of the SSID field is between 0 and 32 octets. A SSID field of length 0 is used within Probe Request management frames to indicate the wildcard SSID. The wildcard SSID is also used in Beacon and Probe Response frames transmitted by mesh STAs.

When the UTF-8 SSID subfield of the Extended Capabilities element is equal to 1 in the frame that includes the SSID element, the SSID is interpreted using UTF-8 encoding.

NOTE—This is true for Beacon and Probe Response frames when the MLME-START.request primitive was issued with the SSIDEncoding parameter equal to UTF8.

8.4.2.3 Supported Rates element

The Supported Rates element specifies up to eight rates in the OperationalRateSet parameter, as described in the MLME-JOIN.request and MLME-START.request primitives, and zero or more BSS membership selectors. The Information field is encoded as 1 to 8 octets, where each octet describes a single Supported Rate or BSS membership selector (see Figure 8-83).

	Element ID	Length	Supported Rates
Octets:	1	1	1–8

Figure 8-83—Supported rates element format

Within Beacon, Probe Response, Association Response, Reassociation Response, Mesh Peering Open, and Mesh Peering Confirm management frames, each Supported Rate contained in the BSSBasicRateSet parameter is encoded as an octet with the MSB (bit 7) set to 1, and bits 6 to 0 are set to the data rate, if necessary rounded up to the next 500kb/s, in units of 500 kb/s. For example, a 2.25 Mb/s rate contained in the BSSBasicRateSet parameter is encoded as X'85'. Rates not contained in the BSSBasicRateSet parameter are encoded with the MSB set to 0, and bits 6 to 0 are set to the appropriate value from the valid range column of the DATA_RATE row of the table in 6.5.5.2 (e.g., a 2 Mb/s rate not contained in the BSSBasicRateSet parameter is encoded as X'04'). The MSB of each Supported Rate octet in other management frame types is ignored by receiving STAs.

Within Beacon, Probe Response, Association Response, Reassociation Response, Mesh Peering Open, and Mesh Peering Confirm management frames, each BSS membership selector contained in the BSSMembershipSelectorSet parameter is encoded as an octet with the MSB (bit 7) set to 1, and bits 6 to 0 are set to the encoded value for the selector as found in Table 8-55 (e.g., an HT PHY BSS membership selector contained in the BSSMembershipSelectorSet parameter is encoded as X'FF'). A BSS membership selector that has the MSB (bit 7) set to 1 in the Supported Rates element is defined to be basic. The MSB of each Supported Rate octet in other management frame types is ignored by receiving STAs.

The valid values for BSS membership selectors and their associated features are shown in Table 8-55.

NOTE—Because the BSS membership selector and supported rates are carried in the same field, the BSS membership selector value cannot match the value corresponding to any valid supported rate. This allows any value in the supported rates set to be determined as either a supported rate or a BSS membership selector.

Table 8-55—BSS membership selector value encoding

Value	Feature	Interpretation
127	HT PHY	Support for the mandatory features of Clause 20 is required in order to join the BSS that was the source of the Supported Rates element or Extended Supported Rates element containing this value.

See 10.1.4.6.

8.4.2.4 FH Parameter Set element

The FH Parameter Set element contains the set of parameters necessary to allow synchronization for STAs using an FH PHY. The Information field contains Dwell Time, Hop Set, Hop Pattern, and Hop Index parameters. The total length of the Information field is 5 octets. See Figure 8-84.

	Element ID	Length	Dwell Time (TU)	Hop Set	Hop Pattern	Hop Index
Octets:	1	1	2	1	1	1

Figure 8-84—FH Parameter Set element format

The Dwell Time field is 2 octets in length and contains the dwell time (dot11CurrentDwellTime) in TU.

The Hop Set field identifies the current set (dot11CurrentSet) of hop patterns and is a single octet.

The Hop Pattern field identifies the current pattern (dot11CurrentPattern) within a set of hop patterns and is a single octet.

The Hop Index field selects the current index (dot11CurrentIndex) within a pattern and is a single octet.

The description of the attributes used in this subclause is in 14.9.2.

8.4.2.5 DSSS Parameter Set element

The DSSS Parameter Set element contains information to allow channel number identification for STAs. The Information field contains a single parameter containing the dot11CurrentChannel (see 16.4.6.3, 17.4.6.3, 18.3.8.4.2 and 20.3.15 for values). The length of the dot11CurrentChannel parameter is 1 octet. See Figure 8-85.

	Element ID	Length	Current Channel
Octets:	1	1	1

Figure 8-85—DSSS Parameter Set element format

8.4.2.6 CF Parameter Set element

The CF Parameter Set element contains the set of parameters necessary to support the PCF. The Information field contains the CFPCount, CFPPeriod, CFPMaxDuration, and CFPDurRemaining fields. The total length of the Information field is 6 octets. See Figure 8-86.

	Element ID	Length	CFP Count	CFP Period	CFP MaxDuration (TU)	CFP DurRemaining (TU)
Octets:	1	1	1	1	2	2

Figure 8-86—CF Parameter Set element format

CFPCount indicates how many delivery traffic indication maps (DTIMs) (including the current frame) appear before the next CFP start. A CFPCount of 0 indicates that the current DTIM marks the start of the CFP.

CFPPeriod indicates the number of DTIM intervals between the start of CFPs. The value is an integral number of DTIM intervals.

CFPMaxDuration indicates the maximum duration, in TU, of the CFP that may be generated by this PCF. This value is used by STAs to set their NAV at the TBTT of Beacon frames that begin CFPs.

CFPDurRemaining indicates the maximum time, in TU, remaining in the present CFP, and is set to 0 in CFP Parameter elements of Beacon frames transmitted during the CP. The value of CFPDurRemaining is referenced to the immediately previous TBTT. This value is used by all STAs to update their NAVs during CFPs.

8.4.2.7 TIM element

The TIM element contains four fields: DTIM Count, DTIM Period, Bitmap Control, and Partial Virtual Bitmap. See Figure 8-87.

	Element ID	Length	DTIM Count	DTIM Period	Bitmap Control	Partial Virtual Bitmap
Octets:	1	1	1	1	1	1–251

Figure 8-87—TIM element format

The Length field for this element indicates the length of the Information field, which is constrained as described below.

The DTIM Count field indicates how many Beacon frames (including the current frame) appear before the next DTIM. A DTIM Count of 0 indicates that the current TIM is a DTIM. The DTIM count field is a single octet.

The DTIM Period field indicates the number of beacon intervals between successive DTIMs. If all TIMs are DTIMs, the DTIM Period field has the value 1. The DTIM Period value 0 is reserved. The DTIM period field is a single octet.

The Bitmap Control field is a single octet. Bit 0 of the field contains the Traffic Indicator bit associated with AID 0. This bit is set to 1 in TIM elements with a value of 0 in the DTIM Count field when one or more group addressed MSDUs/MMPDUs are buffered at the AP or the mesh STA. The remaining 7 bits of the field form the Bitmap Offset.

The traffic-indication virtual bitmap, maintained by the AP or the mesh STA that generates a TIM, consists of 2008 bits, and is organized into 251 octets such that bit number N ($0 \leq N \leq 2007$) in the bitmap corresponds to bit number ($N \bmod 8$) in octet number $\lfloor N / 8 \rfloor$ where the low-order bit of each octet is bit number 0, and the high order bit is bit number 7. Each bit in the traffic-indication virtual bitmap corresponds to traffic buffered for a specific neighbor peer mesh STA within the MBSS that the mesh STA is prepared to deliver or STA within the BSS that the AP is prepared to deliver at the time the Beacon frame is transmitted. Bit number N is 0 if there are no individually addressed MSDUs/MMPDUs buffered for the STA whose AID is N. If any individually addressed MSDUs/MMPDUs for that STA are buffered and the AP or the mesh STA is prepared to deliver them, bit number N in the traffic-indication virtual bitmap is 1. A PC might decline to set bits in the TIM for CF-Pollable STAs it does not intend to poll (see 10.2.1.7).

When dot11MgmtOptionMultiBSSIDActivated is false, the Partial Virtual Bitmap field consists of octets numbered $N1$ to $N2$ of the traffic indication virtual bitmap, where $N1$ is the largest even number such that bits numbered 1 to $(N1 \times 8) - 1$ in the bitmap are all 0 and $N2$ is the smallest number such that bits numbered $(N2 + 1) \times 8$ to 2007 in the bitmap are all 0. In this case, the Bitmap Offset subfield value contains the number $N1/2$, and the Length field is set to $(N2 - N1) + 4$.

In the event that all bits other than bit 0 in the virtual bitmap are 0, the Partial Virtual Bitmap field is encoded as a single octet equal to 0, the Bitmap Offset subfield is 0, and the Length field is 4.

When dot11MgmtOptionMultiBSSIDActivated is true, the Partial Virtual Bitmap field of the TIM element is constructed as follows, where the maximum possible number of BSSIDs is an integer power of 2, n = log2 (maximum possible number of BSSIDs), k is the number of actually supported nontransmitted BSSIDs, and $k \leq (2^n - 1)$.

- The bits 1 to k of the bitmap are used to indicate that one or more group addressed frames are buffered for each AP corresponding to a nontransmitted BSSID. The AIDs from 1 to k are not allocated to a STA. The AIDs from $(k + 1)$ to $(2^n - 1)$ are reserved and set to 0. The remaining AIDs are shared by the BSSs corresponding to the transmitted BSSID and all nontransmitted BSSIDs.
- When the DTIM Count field is 0 for a BSS that has a nontransmitted BSSID, and one or more group addressed frames are buffered at the AP for this BSS, the corresponding bits from bit 1 to bit k is set to 1.

- Each bit starting from bit 2^n in the traffic-indication virtual bitmap corresponds to individually addressed traffic buffered for a specific STA within any BSS corresponding to a transmitted or nontransmitted BSSID at the time the Beacon frame is transmitted. The correspondence is based on the AID of the STA.
- Based upon its knowledge of the capability of associated stations to support the multiple BSSID capability, as indicated by the corresponding field in the Extended Capabilities element and the content of the traffic indication virtual bitmap, an AP shall encode the Partial Virtual Bitmap and the Bitmap Control field of the TIM element using one of the two following methods. Specifically, an AP uses Method B when it determines that the bit for each associated non-AP STA in the virtual bitmap that is reconstructed by each non-AP STA from the received TIM element encoded using Method B is set correctly. Otherwise, an AP uses Method A.

Method A and Method B are described as follows:

a) Method A: The Partial Virtual Bitmap field consists of octets numbered 0 to $N2$ of the traffic indication virtual bitmap, where $N2$ is the smallest number such that bits numbered $(N2 + 1) \times 8$ to 2007 in the bitmap are all 0. If such a value $N2$ does not exist, that is, when not all bits in the last octet of the traffic indication virtual bitmap are equal to 0, $N2 = 250$. When using this method, the Bitmap Offset subfield value always contains the number 0, and the Length field is $N2 + 4$.
b) Method B: The Partial Virtual Bitmap field consists of a concatenation of octets numbered 0 to $N0 - 1$ and octets numbered $N1$ to $N2$ of the traffic indication virtual bitmap, where $N0$ is the smallest positive integer such that $N0 \times 8 - 2^n < 8$. If $N0$ is an odd number, then $N1$ is the largest odd number such that $N0 < N1$and each of the bits $N0 \times 8$ to $(N1 \times 8 - 1)$ is equal to 0. When $N0$ is an even number, $N1$ is the largest even number such that $N0 < N1$ and each of the bits $N0 \times 8$ to $(N1 \times 8 - 1)$ is equal to 0. If such a value $N1 > N0$ does not exist, $N1 = N0$. Additionally, $N2$ is the smallest integer value for which the values for bit $(N2+1) \times 8$ to 2007 in the bitmap are all 0. If such a value $N2$ does not exist, that is, when not all bits in the last octet of the traffic indication virtual bitmap are equal to 0, $N2 = 250$. When using this method, the Bitmap Offset subfield contains the value of $(N1 - N0)/2$, and the Length field is $N0 + N2 - N1 + 4$.

NOTE—When $N1 = N0$, Method B reduces to Method A.

For both Method A and Method B, when there are no frames buffered for any BSS corresponding to a transmitted or nontransmitted BSSID supported, the Partial Virtual Bitmap field is encoded as a single octet equal to 0, the Bitmap Offset subfield is 0, and the Length field is 4. When there are no buffered individually addressed frames for any BSS corresponding to a transmitted or nontransmitted BSSID, but there are buffered group addressed frames for one or more of the BSSs, the Partial Virtual Bitmap field consists of the octets number 0 to $N0 - 1$ where $N0$ is the smallest positive integer such that $(N0 \times 8 - 2^n < 8)$. In this case, the Bitmap Offset subfield value contains the number 0, and the Length field is $N0$+3.

8.4.2.8 IBSS Parameter Set element

The IBSS Parameter Set element contains the set of parameters necessary to support an IBSS. The Information field contains the ATIM Window parameter. See Figure 8-88.

	Element ID	Length	ATIM Window
Octets:	1	1	2

Figure 8-88—IBSS Parameter Set element format

The ATIM Window field is 2 octets in length and contains the ATIM Window length in TU.

8.4.2.9 Challenge Text element

The Challenge Text element contains the challenge text within Authentication exchanges. The length of the Information field is dependent upon the authentication algorithm and the transaction sequence number as specified in 11.2.3.2. See Figure 8-89.

	Element ID	Length	Challenge Text
Octets:	1	1	1–253

Figure 8-89—Challenge Text element format

8.4.2.10 Country element

The Country element contains the information required to allow a STA to identify the regulatory domain in which the STA is located and to configure its PHY for operation in that regulatory domain. The format of this element is as shown in Figure 8-90.

These three fields are repeated, as determined by the Length field

	Element ID	Length	Country String	First Channel Number / Operating Extension Identifier	Number of Channels / Operating Class	Maximum Transmit Power Level / Coverage Class	Pad (if needed)
Octets:	1	1	3	1	1	1	0 or 1

Figure 8-90—Country element format

The element ID for this element is set to the value for Country, specified in Table 8-54. The length of the element is variable, as the element may contain more than one triplet comprising the First Channel Number, Number of Channels, and Maximum Transmit Power Level fields and referred to as subband triplets. Alternatively, where dot11OperatingClassesRequired is true and the First Channel Number/Operating Extension Identifier octet has a positive integer value of 201 or greater, then that triplet comprises the Operating Extension Identifier, Operating Class, and Coverage Class fields. Together they are referred to as an operating triplet. The minimum length of the element is 8 octets.

The Country String field of the element is 3 octets in length. The AP and mesh STA set this field to the value contained in the dot11CountryString attribute before transmission in a Beacon or Probe Response frame. Upon reception of this element, a STA sets the value of the dot11CountryString to the value contained in this field.

The First Channel Number/Operating Extension Identifier field is 1 octet in length. If the field has a positive integer value less than 201, then it contains a positive integer value that indicates the lowest channel number in the subband described in this element. The group of channels described by each pair of the First Channel Number and the Number of Channels fields do not have overlapping channel identifiers. [For example, the pairs (2,4) and (5,2) overlap and are not used together.] The First Channel Numbers are monotonically increasing where dot11OperatingClassesRequired is not true.

Where dot11OperatingClassesRequired is true, consecutive subband triplets following an operating triplet have monotonically increasing First Channel Number fields.

The Number of Channels field of the subelement is 1 octet in length.

The Maximum Transmit Power Level field is a signed number and is 1 octet in length. It indicates the maximum power, in dBm, allowed to be transmitted. As the method of measurement for maximum transmit power level differs by regulatory domain, the value in this field is interpreted according to the regulations applicable for the domain identified by the Country String.

An operating class is an index into a set of values for radio equipment sets of rules. The Operating Class field is 1 octet in length.

A coverage class is an index into a set of values for aAirPropagationTime. The Coverage Class field is 1 octet in length.

The Coverage Class field of the operating triplet specifies the aAirPropagationTime characteristic used in BSS operation, as shown in Table 8-56. The characteristic aAirPropagationTime describes variations in actual propagation time that are accounted for in a BSS and, together with maximum transmit power level, allow control of BSS diameter.

Table 8-56—Coverage Class field parameters

Coverage class value	aAirPropagationTime (µs)
0	≤ 1
1	3
2	6
3	9
4	12
5	15
6	18
7	21
8	24
9	27
10	30
11	33
12	36
13	39
14	42
15	45
16	48
17	51
18	54
19	57
20	60
21	63
22	66

Table 8-56—Coverage Class field parameters *(continued)*

Coverage class value	aAirPropagationTime (µs)
23	69
24	72
25	75
26	78
27	81
28	84
29	87
30	90
31	93
32–255	—

The Pad field is 0 or 1 octet in length. The length of the Country element is evenly divisible by 2. The Pad is used to add a single octet to the element if the length is not evenly divisible by 2. The value of the Pad field is 0.

8.4.2.11 Hopping Pattern Parameters element

The mechanisms described in this subclause are obsolete. Consequently, this subclause may be removed in a later revision of the standard.

The Hopping Pattern Parameters element contains the information necessary to allow a STA to calculate the code family using the hyperbolic congruence code (HCC) and extended HCC (EHCC) algorithms. See 9.18.3 for a description of the HCC and EHCC algorithms. The format of this element is as shown in Figure 8-91.

	Element ID	Length	Prime Radix	Number of Channels
Octets:	1	1	1	1

Figure 8-91—Hopping Pattern Parameters element

The Element ID of this element is 8. The length of this element is 4 octets.

The Prime Radix field of this element indicates the value to be used as the prime radix (*N*) in the HCC and EHCC algorithms. The value of this field is a positive integer. The size of this field is 1 octet.

The Number of Channels field of this element indicates the value to be used as the maximum for the family index (*a*) in the HCC and EHCC algorithms. The value of this field is a positive integer and is not less than the prime radix minus 3 (*N*–3). The size of this field is 1 octet.

8.4.2.12 Hopping Pattern Table element

The Hopping Pattern Table element contains the information necessary for an FH implementation to be able to create the hopping sequences necessary to operate in the regulatory domain in which the element was received. The format of the element is as shown in Figure 8-92.

	Element ID	Length	Flag	Number of Sets	Modulus	Offset	Random Table
Octets:	1	1	1	1	1	1	variable

Figure 8-92—Hopping Pattern Table element

The Element ID of this element is set to the value for Hopping Pattern Table, specified in Table 8-54. The element is variable in length. The length of the element is indicated by the Length field.

The Flag field indicates that a Random Table is present when the value is 1. When the flag value is 0, it indicates that a Random Table is not present and that the hop index method is to be used to determine the hopping sequence. The size of this field is 1 octet.

The Number of Sets field indicates the total number of sets within the hopping patterns. The size of this field is 1 octet.

The Modulus and Offset fields indicate the values to be used in the equations to create a hopping sequence from the Random Table information. The size of these fields are each 1 octet.

The Random Table field is a variable-length field. It is a vector of single octet values that indicate the random sequence to be followed during a hopping sequence. The size of the Random Table field is found by subtracting 4 from the value of the Length field of this element.

See 9.18.4.

8.4.2.13 Request element

This element is placed in a Probe Request frame to request that the responding STA include the requested information in the Probe Response frame. The format of the element is as shown in Figure 8-93.

	Element ID	Length	Requested Element IDs
Octets:	1	1	variable

Figure 8-93—Request element

The Element ID of this element is set to the value for Request, specified in Table 8-54. The element is variable in length. The length of the element is indicated in the Length field.

The Requested Element IDs are the list of elements that are requested to be included in the Probe Response frame. The Requested Element IDs are listed in order of increasing element ID.

See 10.1.4.3.2 for additional requirements.

8.4.2.14 ERP element

The ERP element contains information on the presence of Clause 16 or Clause 17 STAs in the BSS that are not capable of Clause 19 (ERP-OFDM) data rates. It also contains the requirement of the ERP element sender (AP in an infrastructure BSS, STA in an IBSS, or mesh STA in an MBSS) as to the use of protection mechanisms to optimize BSS performance and as to the use of long or short Barker preambles. See Figure 8-94 for a definition of the frame element.

The ERP element has the form shown in Figure 8-94. Note that the length of this element is flexible and may be expanded in the future.

	Element ID	Length (1)	ERP Parameters
Octets:	1	1	1

Figure 8-94—ERP element

The ERP Parameters field is defined in Figure 8-95.

	B0	B1	B2	B3 B7
	NonERP_Present	Use_Protection	Barker_Preamble_Mode	Reserved
Bits:	1	1	1	5

Figure 8-95—ERP Parameters field

Recommended behavior for setting the Use_Protection bit is contained in 9.23.

8.4.2.15 Extended Supported Rates element

The Extended Supported Rates element specifies the rates in the OperationalRateSet parameter, as described in the MLME_JOIN.request and MLME_START.request primitives, and zero or more BSS membership selector values that are not carried in the Supported Rates element. The Information field is encoded as 1 to 255 octets where each octet describes a single supported rate or BSS membership selector.

Within Beacon, Probe Response, Association Response, Reassociation Response, Mesh Peering Open, and Mesh Peering Confirm management frames, each supported rate contained in the BSSBasicRateSet parameter, as defined in 6.3.11.2, is encoded as an octet with the MSB (bit 7) set to 1 and bits 6 to 0 are set to the appropriate value from the valid range column of the DATA_RATE row of the table in 6.5.5.2 (e.g., a 1 Mb/s rate contained in the BSSBasicRateSet parameter is encoded as X'82'). Rates not contained in the BSSBasicRateSet parameter are encoded with the MSB set to 0, and bits 6 to 0 are set to the appropriate value from the valid range column of the DATA_RATE row of the table in 6.5.5.2 (e.g., a 2 Mb/s rate not contained in the BSSBasicRateSet parameter is encoded as X'04'). The MSB of each octet in the Extended Supported Rate element in other management frame types is ignored by receiving STAs.

Within Beacon, Probe Response, Association Response, Reassociation Response, Mesh Peering Open, and Mesh Peering Confirm management frames, each BSS membership selector contained in the BSSMembershipSelectorSet parameter is encoded as an octet with the MSB (bit 7) set to 1, and bits 6 to 0 are set to the encoded value for the selector as found in Table 8-55 (e.g., an HT PHY BSS membership selector contained in the BSSMembershipSelectorSet parameter is encoded as X'FF').

The Extended Supported Rates element has the format shown in Figure 8-96.

	Element ID	Length	Extended Supported Rates
Octets:	1	1	1–255

Figure 8-96—Extended Supported Rates element format

See 10.1.4.6

8.4.2.16 Power Constraint element

The Power Constraint element contains the information necessary to allow a STA to determine the local maximum transmit power in the current channel. The format of the Power Constraint element is shown in Figure 8-97.

	Element ID	Length	Local Power Constraint
Octets:	1	1	1

Figure 8-97—Power Constraint element format

The Length field is set to 1.

The field is coded as an unsigned integer in units of decibels. The local maximum transmit power for a channel is thus defined as the maximum transmit power level specified for the channel in the Country element minus the local power constraint specified for the channel (from the MIB) in the Power Constraint element.

The Power Constraint element is included in Beacon frames, as described in 8.3.3.2, and Probe Response frames, as described in 8.3.3.10. The use of Power Constraint elements is described in 10.8.4.

8.4.2.17 Power Capability element

The Power Capability element specifies the minimum and maximum transmit powers with which a STA is capable of transmitting in the current channel. The format of the Power Capability element is shown in Figure 8-98.

	Element ID	Length	Minimum Transmit Power Capability	Maximum Transmit Power Capability
Octets:	1	1	1	1

Figure 8-98—Power Capability element format

The Length field is set to 2.

The Minimum Transmit Power Capability field is set to the nominal minimum transmit power with which the STA is capable of transmitting in the current channel, with a tolerance ± 5 dB. The field is coded as a signed integer in units of decibels relative to 1 mW.

The Maximum Transmit Power Capability field is set to the nominal maximum transmit power with which the STA is capable of transmitting in the current channel, with a tolerance ± 5 dB. The field is coded as a signed integer in units of decibels relative to 1 mW.

The Power Capability element is included in Association Request frames, as described in 8.3.3.5; Reassociation Request frames, as described in 8.3.3.7; and Mesh Peering Open frame, as described in 8.5.16.2.2. The use of Power Capability elements is described in 10.8.2.

8.4.2.18 TPC Request element

The TPC Request element contains a request for a STA to report transmit power and link margin information using a TPC Report element. The format of the TPC Request element is shown in Figure 8-99.

	Element ID	Length
Octets:	1	1

Figure 8-99—TPC Request element format

The Length field is set to 0.

The TPC Request element is included in TPC Request frames, as described in 8.5.2.4. The use of TPC Request elements and frames is described in 10.8.6.

8.4.2.19 TPC Report element

The TPC Report element contains transmit power and link margin information sent in response to a TPC Request element or a Link Measurement Request frame. A TPC Report element is included in a Beacon frame or Probe Response frame without a corresponding request. The format of the TPC Report element is shown in Figure 8-100.

	Element ID	Length	Transmit Power	Link Margin
Octets:	1	1	1	1

Figure 8-100—TPC Report element format

The Length field is set to 2.

The Transmit Power field is set to the transmit power used to transmit the frame containing the TPC Report element. The field is coded as a twos complement signed integer in units of decibels relative to 1 mW. The maximum tolerance for the transmit power value reported in the TPC Response element is ± 5 dB. This tolerance is defined as the difference, in decibels, between the reported power value and the actual EIRP of the STA (when transmitting 1500 octet frames or maximum MPDU sized-frames, whichever is smaller).

The Link Margin field contains the link margin for the receive time and for the receive rate of the frame containing the TPC Request element or the Link Measurement Request frame. The field is coded as a twos complement signed integer in units of decibels. The Link Margin field is reserved when a TPC Report element is included in a Beacon frame or Probe Response frame. The measurement method of Link Margin is beyond the scope of this standard.

The TPC Report element is included in TPC Report frames, as described in 8.5.2.5; Link Measurement Report frames, as described in 8.5.7.5; Beacon frames, as described in 8.3.3.2; and Probe Response frames, as described in 8.3.3.10. The use of TPC Report elements and frames is described in 10.8.6.

8.4.2.20 Supported Channels element

The Supported Channels element contains a list of channel subbands (from those channels defined in 18.3.8.4.3) in which a STA is capable of operating. The format of the Supported Channels element is shown in Figure 8-101.

			One (first channel, number of channels) tuple for each subband	
	Element ID	Length	First Channel Number	Number of Channels
Octets:	1	1	1	1

Figure 8-101—Supported Channels element format

The Length field is variable and depends on the number of subbands, defined by a First Channel Number–Number of Channels pair, that are included in the element.

The First Channel Number field is set to the first channel (as defined in 18.3.8.4.3) in a subband of supported channels.

The Number of Channels field is set to the number of channels in a subband of supported channels.

The Supported Channels element is included in Association Request frames, as described in 8.3.3.5; Reassociation Request frames, as described in 8.3.3.7; and Mesh Peering Open frame, as described in 8.5.16.2.2. The use of the Supported Channels element is described in 10.9.2 and 10.9.8.

8.4.2.21 Channel Switch Announcement element

The Channel Switch Announcement element is used by an AP in a BSS, a STA in an IBSS, or a mesh STA in an MBSS to advertise when it is changing to a new channel and the channel number of the new channel. The format of the Channel Switch Announcement element is shown in Figure 8-102.

	Element ID	Length	Channel Switch Mode	New Channel Number	Channel Switch Count
Octets:	1	1	1	1	1

Figure 8-102—Channel Switch Announcement element format

The Length field is set to 3.

The Channel Switch Mode field indicates any restrictions on transmission until a channel switch. An AP in a BSS or a STA in an IBSS sets the Channel Switch Mode field to either 0 or 1 on transmission. In an MBSS, the Channel Switch Mode Field is reserved. See 10.9.9.

The New Channel Number field is set to the number of the channel to which the STA is moving (as defined in 18.3.8.4.3).

For nonmesh STAs, the Channel Switch Count field either is set to the number of TBTTs until the STA sending the Channel Switch Announcement element switches to the new channel or is set to 0. A value of 1 indicates that the switch occurs immediately before the next TBTT. A value of 0 indicates that the switch occurs at any time after the frame containing the element is transmitted.

For mesh STAs, the Channel Switch Count field is encoded as an octet with bits 6 to 0 set to the time, in units of 2 TU when the MSB (bit 7) is 0, or in units of 100 TU when the MSB (bit 7) is 1, until the mesh STA sending the Channel Switch Announcement element switches to the new channel. A value of 0 for bits 6 to 0 indicates that the switch occurs at any time after the frame containing the element is transmitted. For example, a 200 TU channel switch time is encoded as X'82' and a 10 TU channel switch time is encoded as X'05'.

The Channel Switch Announcement element is included in Channel Switch Announcement frames, as described in 8.5.2.6, and may be included in Beacon frames, as described in 8.3.3.2, and Probe Response frames, as described in 8.3.3.10. The use of Channel Switch Announcement elements and frames is described in 10.9.8.

8.4.2.22 Secondary Channel Offset element

The Secondary Channel Offset element is used by an AP in a BSS, a STA in an IBSS, or a mesh STA in an MBSS together with the Channel Switch Announcement element when changing to a new 40 MHz channel. The format of the Secondary Channel Offset element is shown in Figure 8-103.

The Secondary Channel Offset element is included in Channel Switch Announcement frames, as described in 8.5.2.6.

	Element ID	Length (=1)	Secondary Channel Offset
Octets:	1	1	1

Figure 8-103—Secondary Channel Offset element format

The Secondary Channel Offset field of the Secondary Channel Offset element represents the position of the secondary channel relative to the primary channel. The values of the Secondary Channel Offset field are defined in Table 8-57.

Table 8-57—Values of the Secondary Channel Offset field

Value	Name	Description
0	SCN - no secondary channel	Indicates that no secondary channel is present.
1	SCA - secondary channel above	Indicates that the secondary channel is above the primary channel.
2		Reserved.
3	SCB - secondary channel below	Indicates that the secondary channel is below the primary channel.
4–255		Reserved.

8.4.2.23 Measurement Request element

8.4.2.23.1 General

The Measurement Request element contains a request that the receiving STA undertake the specified measurement action. The Measurement Request element is included in Spectrum Management Measurement Request frames, as described in 8.5.2.2, or Radio Measurement Request frames, as described in 8.5.7.2. Measurement Types 0, 1, and 2 are defined for spectrum management and are included only in Spectrum Management Measurement Request frames. The use of Measurement Request elements for spectrum management is described in 10.9.7. Measurement Types 3 to 9 and 255 are defined for radio measurement and are included only in Radio Measurement Request frames. The use of Measurement Request elements for radio measurement is described in 10.11.

The format of the Measurement Request element is shown in Figure 8-104.

	Element ID	Length	Measurement Token	Measurement Request Mode (see Figure 8-105)	Measurement Type	Measurement Request
Octets:	1	1	1	1	1	variable

Figure 8-104—Measurement Request element format

	B0	B1	B2	B3	B4	B5 B7
	Parallel	Enable	Request	Report	Duration Mandatory	Reserved
Bits:	1	1	1	1	1	3

Figure 8-105—Measurement Request Mode field

The value of the Length field in octets is variable and depends on the length of the Measurement Request field. The minimum value of the Length field is 3 (based on a minimum length for the Measurement Request field of 0 octets).

The Measurement Token is set to a nonzero number that is unique among the Measurement Request elements in a particular Measurement Request frame.

The Measurement Request Mode field (shown in Figure 8-105) is a bit field with the following bits defined:

- The Parallel bit (bit 0) is used to request that more than one measurement is to be started in parallel. Parallel is set to 1 to request that the measurement is to start at the same time as the measurement described by the next Measurement Request element in the same Measurement Request frame. Parallel is set to 0 if the measurements are to be performed in sequence. The Parallel bit is reserved when Enable is 1, in the last or only measurement request element in the frame, or when the value of the Measurement Type field is 0, 1, or 2 (spectrum management measurements). See 10.11.6.
- The Enable bit (bit 1) is used to differentiate between a request to make a measurement and a request to control the measurement requests and triggered or autonomous reports generated by the destination STA. The Enable bit is further described in Table 8-58.
- Request bit (bit 2) is described in Table 8-58.
- Report bit (bit 3) is described in Table 8-58.
- The Duration Mandatory bit (bit 4) indicates whether the measurement duration contained within the Measurement Request is interpreted as mandatory by the STA receiving the request. A value of 0 indicates that the duration requested is a maximum duration, and the requesting STA accepts measurement results taken over any shorter duration. A value of 1 indicates that the duration requested is a mandatory duration. The Duration Mandatory bit is reserved when the Enable bit is 1, or when the value of the Measurement Type field is 0, 1, 2, 8, or 255. See 10.11.4.
- All other bits are reserved.

The use of the Enable, Request, and Report bits is summarized in Table 8-58. See 10.9.7 and 10.11.6 for the description of how a STA handles requests to enable or disable measurement requests and autonomous reports. See 10.11.8 for a description of the use of the Enable and Report bits in triggered reporting.

Table 8-58—Summary of use of Enable, Request, and Report bits

Bits			Meaning of bits
Enable	Request	Report	
0	Reserved	Reserved	The transmitting STA is requesting that the destination STA make a Measurement of type indicated in the Measurement Type field. When Enable bit is 0, Request and Report bits are reserved.
1	0	0	The transmitting STA is requesting that the destination STA not send any measurement requests or reports of the type indicated in the Measurement Type field.
1	1	0	The transmitting STA is indicating to the destination STA that it can accept measurement requests and is requesting the destination STA not send autonomous or triggered measurement reports of the type indicated in the Measurement Type field. NOTE—This setting corresponds to the default STA behavior.
1	0	1	The transmitting STA is requesting that the destination STA not send measurement requests and indicating it accepts autonomous or triggered measurement reports of the type indicated in the Measurement Type field.
1	1	1	The transmitting STA is indicating to the destination STA that it can accept measurement requests and can accept autonomous or triggered measurement reports of the type indicated in the Measurement Type field.

The Measurement Type field is set to a number that identifies a type of measurement request or measurement report. The Measurement Types that have been allocated for measurement requests are shown in Table 8-59 and measurement reports are shown in Table 8-81 (in 8.4.2.24).

Table 8-59—Measurement Type definitions for measurement requests

Name	Measurement Type	Measurement Use
Basic request	0	Spectrum Management
Clear channel assessment (CCA) request	1	
Receive power indication (RPI) histogram request	2	
Channel load request	3	Radio Measurement
Noise histogram request	4	
Beacon request	5	
Frame request	6	
STA statistics request	7	Radio Measurement and WNM
LCI request	8	Radio Measurement and WNM
Transmit stream/category measurement request	9	Radio Measurement
Multicast diagnostics request	10	WNM
Location Civic request	11	Radio Measurement and WNM
Location Identifier request	12	Radio Measurement and WNM
Reserved	13–254	N/A
Measurement pause request	255	Radio Measurement

When the Enable bit is 0, the Measurement Request field contains the specification of a single measurement request corresponding to the Measurement Type, as described in 8.4.2.23.2 to 8.4.2.23.12. When the Enable bit is 1, the Measurement Request field is present only when requesting a triggered measurement.

8.4.2.23.2 Basic request

A Measurement Type in the Measurement Request element may indicate a basic request. The Measurement Request field corresponding to a basic request is shown in Figure 8-106. See 10.9.7.

	Channel Number	Measurement Start Time	Measurement Duration
Octets:	1	8	2

Figure 8-106—Measurement Request field format for a basic request

The Channel Number field is set to the channel number for which the measurement request applies (as defined in 18.3.8.4.3).

The Measurement Start Time field is set to the TSF timer at the time (± 32 µs) at which the requested basic request measurement starts. A value of 0 indicates it starts immediately.

The Measurement Duration field is set to the duration of the requested measurement, expressed in TUs.

8.4.2.23.3 CCA request

A Measurement Type in the Measurement Request element may indicate a CCA request. A response to a CCA request is a CCA report. It is optional for a STA to generate a CCA report in response to a CCA Request. The Measurement Request field corresponding to a CCA request is shown in Figure 8-107.

	Channel Number	Measurement Start Time	Measurement Duration
Octets:	1	8	2

Figure 8-107—Measurement Request field format for a CCA request

The Channel Number field is set to the channel number for which the measurement request applies (as defined in 18.3.8.4.3).

The Measurement Start Time field is set to the TSF at the time (± 32 µs) at which the requested CCA request measurement starts. A value of 0 indicates it starts immediately.

The Measurement Duration field is set to the duration of the requested measurement, expressed in TUs.

8.4.2.23.4 RPI histogram request

A Measurement Type in the Measurement Request element may indicate an RPI histogram request. A response to an RPI histogram request is an RPI histogram report. It is optional for a STA to generate a RPI histogram report in response to a RPI histogram request. The Measurement Request field corresponding to an RPI histogram request is shown in Figure 8-108.

	Channel Number	Measurement Start Time	Measurement Duration
Octets:	1	8	2

Figure 8-108—Measurement Request field format for a RPI histogram request

The Channel Number field is set to the channel number for which the measurement request applies (as defined in 18.3.8.4.3).

The Measurement Start Time field is set to the TSF at the time (± 32 µs) at which the requested RPI histogram request measurement starts. A value of 0 indicates it starts immediately.

The Measurement Duration field is set to the duration of the requested measurement, expressed in TUs.

8.4.2.23.5 Channel Load Request

The Measurement Request field corresponding to a Channel Load Request is shown in Figure 8-109.

	Operating Class	Channel Number	Randomization Interval	Measurement Duration	Optional Subelements
Octets:	1	1	2	2	variable

Figure 8-109—Measurement Request field format for Channel Load Request

Operating Class indicates the channel set for which the measurement request applies. Country, Operating Class, and Channel Number together specify the channel frequency and spacing for which the measurement request applies. Valid values of Operating Class are shown in Annex E.

Channel Number indicates the channel number for which the measurement request applies. Channel Number is defined within an Operating Class as shown in Annex E.

Randomization Interval specifies the upper bound of the random delay to be used prior to making the measurement, expressed in units of TUs. See 10.11.3.

The Measurement Duration field is set to the preferred or mandatory duration of the requested measurement, expressed in units of TUs. See 10.11.4.

The Optional Subelements field format contains zero or more subelements, each consisting of a 1-octet Subelement ID field, a 1-octet Length field, and a variable-length Data field, as shown in Figure 8-402. Any optional subelements are ordered by nondecreasing subelement ID.

The Subelement ID field values for the defined optional subelements are shown in Table 8-60. A Yes in the Extensible column of a subelement listed in Table 8-60 indicates that the Length of the subelement might be extended in future revisions or amendments of this standard. When the Extensible column of an element is equal to Subelements, then the subelement might be extended in future revisions or amendments of this standard by defining additional subelements within the subelement. See 9.24.9.

Table 8-60—Optional subelement IDs for Channel Load Request

Subelement ID	Name	Length field (octets)	Extensible
0	Reserved		
1	Channel Load Reporting Information	2	Yes
2–220	Reserved		
221	Vendor Specific	1 to 244	
222–255	Reserved		

The Channel Load Reporting Information subelement indicates the condition for issuing a Channel Load Report. Channel Load Reporting Information subelement data field format is shown in Figure 8-110 and contains a 1-octet Reporting Condition subfield and a 1-octet Channel Load Reference Value subfield. The Reporting Condition is described in Table 8-61. The Channel Load Reference value is a Channel Load value as defined in 10.11.9.3 and is the reference value for the indicated Reporting Condition.

	Reporting Condition	Channel Load Reference Value
Octets:	1	1

Figure 8-110—Channel Load Reporting Information data field format

Table 8-61—Reporting Condition for Channel Load Report

Condition for report to be issued	Reporting Condition
Report to be issued after each measurement (default, used when Channel Load Reporting Information subelement is not included in Channel Load Request).	0
Report to be issued when measured Channel Load is equal to or greater than the reference value.	1
Report to be issued when measured Channel Load is equal to or less than the reference value.	2
Reserved	3–255

The Vendor Specific subelements have the same format as their corresponding elements (see 8.4.2.28). Multiple Vendor Specific subelements might be included in the list of optional subelements.

8.4.2.23.6 Noise Histogram Request

The Measurement Request field corresponding to a Noise Histogram Request is shown in Figure 8-111.

	Operating Class	Channel Number	Randomization Interval	Measurement Duration	Optional Subelements
Octets:	1	1	2	2	variable

Figure 8-111—Measurement Request field format for Noise Histogram Request

Operating Class indicates the channel set for which the measurement request applies. Country, Operating Class, and Channel Number together specify the channel frequency and spacing for which the measurement request applies. Valid values of Operating Class are shown in Annex E.

Channel Number indicates the channel number for which the measurement request applies. Channel Number is defined within an Operating Class as shown in Annex E.

Randomization Interval specifies the upper bound of the random delay to be used prior to making the measurement, expressed in units of TUs. See 10.11.3.

The Measurement Duration field is set to the preferred or mandatory duration of the requested measurement, expressed in units of TUs. See 10.11.4.

The Optional Subelements field format contains zero or more subelements, each consisting of a 1-octet Subelement ID field, a 1-octet Length field, and a variable-length Data field, as shown in Figure 8-402. Any optional subelements are ordered by nondecreasing subelement ID.

The Subelement ID field values for the defined optional subelements are shown in Table 8-62. A Yes in the Extensible column of a subelement listed in Table 8-62 indicates that the Length of the subelement might be extended in future revisions or amendments of this standard. When the Extensible column of an element is equal to Subelements, then the subelement might be extended in future revisions or amendments of this standard by defining additional subelements within the subelement. See 9.24.9.

Table 8-62—Optional subelement IDs for Noise Histogram Request

Subelement ID	Name	Length field (octets)	Extensible
0	Reserved		
1	Noise Histogram Reporting Information	2	Yes
2–220	Reserved		
221	Vendor Specific	1 to 244	
222–255	Reserved		

The Noise Histogram Reporting Information subelement indicates the condition for issuing a Noise Histogram Report. The Noise Histogram Reporting Information subelement data field format is shown in Figure 8-112 and contains a 1-octet Reporting Condition subfield and a 1-octet ANPI Reference Value subfield. The Reporting Condition is described in Table 8-63. The ANPI Reference Value is an ANPI value as defined in 10.11.9.4 and is the reference value for the indicated Reporting Condition.

	Reporting Condition	ANPI Reference Value
Octets:	1	1

Figure 8-112—Noise Histogram Reporting Information data field format

The Vendor Specific subelements have the same format as their corresponding elements (see 8.4.2.28). Multiple Vendor Specific subelements might be included in the list of optional subelements.

Table 8-63—Reporting Condition for Noise Histogram Report

Condition for report to be issued	Reporting Condition
Report to be issued after each measurement (default, used when Noise Histogram Reporting Information subelement is not included in Noise Histogram Request).	0
Noise Histogram Report to be issued when measured ANPI is equal to or greater than the reference value.	1
Noise Histogram Report to be issued when measured ANPI is equal to or less than the reference value.	2
Reserved	3–255

8.4.2.23.7 Beacon Request

The Measurement Request field corresponding to a Beacon Request is shown in Figure 8-113.

	Operating Class	Channel Number	Randomization Interval	Measurement Duration
Octets:	1	1	2	2

	Measurement Mode	BSSID	Optional Subelements
Octets:	1	6	variable

Figure 8-113—Measurement Request field format for Beacon Request

Operating Class indicates the channel set for which the measurement request applies. Country, Operating Class, and Channel Number together specify the channel frequency and spacing for which the measurement request applies. Valid values of Operating Class are shown in Annex E.

Channel Number indicates the channel number for which the measurement request applies. Channel Number is defined within an Operating Class as shown in Annex E. A Channel Number of 0 indicates a request to make iterative measurements for all supported channels in the Operating Class where the measurement is permitted on the channel and the channel is valid for the current regulatory domain. A Channel Number of 255 indicates a request to make iterative measurements for all supported channels in the current Operating Class listed in the latest AP Channel Report received from the serving AP. The procedures for iterative measurements on multiple channels are described in 10.11.9.1.

Randomization Interval specifies the upper bound of the random delay to be used prior to making the measurement, expressed in units of TUs. See 10.11.3.

The Measurement Duration field is set to the preferred or mandatory duration of the requested measurement, expressed in units of TUs. See 10.11.4.

Measurement Mode indicates the mode to be used for the measurement. The valid measurement modes are listed in Table 8-64. The procedures for each mode are described in 10.11.9.1.

Table 8-64—Measurement Mode definitions for Beacon Request element

Mode	Value
Passive	0
Active	1
Beacon Table	2
Reserved	3–255

The BSSID field indicates the BSSID of the BSS(s) for which a beacon report is requested. When requesting beacon reports for all BSSs on the channel, the BSSID field contains the wildcard BSSID; otherwise the BSSID field contains a specific BSSID for a single BSS.

The Optional Subelements field format contains zero or more subelements, each consisting of a 1-octet Subelement ID field, a 1-octet Length field, and a variable-length Data field, as shown in Figure 8-402. Any optional subelements are ordered by nondecreasing subelement ID.

The Subelement ID field values for the defined optional subelements are shown in Table 8-65. A Yes in the Extensible column of a subelement listed in Table 8-65 indicates that the Length of the subelement might be extended in future revisions or amendments of this standard. When the Extensible column of an element is equal to Subelements, then the subelement might be extended in future revisions or amendments of this standard by defining additional subelements within the subelement. See 9.24.9.

Table 8-65—Optional subelement IDs for Beacon Request

Subelement ID	Name	Length field (octets)	Extensible
0	SSID	0 to 32	
1	Beacon Reporting Information	2	Yes
2	Reporting Detail	1	Yes
3–9	Reserved		
10	Request	0 to 237	
11–50	Reserved		
51	AP Channel Report	1 to 237	
52–220	Reserved		
221	Vendor Specific	1 to 237	
222–255	Reserved		

The SSID subelement indicates the ESS(s) or IBSS(s) for which a beacon report is requested. When SSID is not included in a Beacon Request, the default “wildcard SSID” is used; otherwise the SSID is included in the Beacon Request and contains a specific SSID for a single ESS or IBSS. The wildcard SSID is used to represent all possible SSIDs. The SSID element is described in 8.4.2.2.

The Beacon Reporting Information subelement indicates the condition for issuing a Beacon Report. The Beacon Reporting Information subelement may be included in a Beacon Request only for repeated measurements. The Beacon Reporting Information subelement data field format is shown in Figure 8-114 and contains a 1-octet Reporting Condition subfield and a 1-octet Threshold/Offset Reference Value subfield. The Reporting Condition is described in Table 8-66. The Threshold/Offset Reference Value provides either the threshold value or the offset value to be used for conditional reporting. For Reporting Conditions 1 and 2, the threshold value is a logarithmic function of the received signal power, as defined in the RCPI measurement subclause for the associated PHY Type. For Reporting Conditions 3 and 4, the threshold value is a logarithmic function of the signal-to-noise ratio, as described in 8.4.2.43. For Reporting Conditions 5 to 10, the offset value is an 8-bit two’s complement integer in units of 0.5 dBm. The indicated Reporting Condition applies individually to each measured Beacon, Measurement Pilot, or Probe Response. Reporting Conditions are further described in 10.11.9.1.

	Reporting Condition	Threshold/Offset Reference Value
Octets:	1	1

Figure 8-114—Beacon Reporting Information data field format

Table 8-66—Reporting Condition for Beacon Report

Condition for report to be issued in Repeated Measurement	Reporting Condition
Report to be issued after each measurement (default, used when Beacon Reporting Information subelement is not included in Beacon Request).	0
The measured RCPI level is greater than the threshold indicated in the Threshold/ Offset Reference Value.	1
The measured RCPI level is less the threshold indicated in the Threshold/Offset Reference Value.	2
The measured RSNI level is greater than the threshold indicated in the Threshold/ Offset Reference Value.	3
The measured RSNI level is less than the threshold indicated in the Threshold/Offset Reference Value.	4
The measured RCPI level is greater than a threshold defined by an offset from the serving AP’s reference RCPI, where the offset is indicated in the Threshold/Offset Reference Value.	5
The measured RCPI level is less than a threshold defined by an offset from the serving AP’s reference RCPI, where the offset is indicated in the Threshold/Offset Reference Value.	6
The measured RSNI level is greater than a threshold defined by an offset from the serving AP’s reference RSNI, where the offset is indicated in the Threshold/Offset Reference Value.	7
The measured RSNI level is less than a threshold defined by an offset from the serving AP’s reference RSNI, where the offset is indicated in the Threshold/Offset Reference Value.	8
The measured RCPI level is in a range bound by the serving AP’s reference RCPI and an offset from the serving AP’s reference RCPI, where the offset is indicated in the Threshold/Offset Reference Value.	9
The measured RSNI level is in a range bound by the serving AP’s reference RSNI and an offset from the serving AP’s reference RSNI, where the offset is indicated in the Threshold/Offset Reference Value.	10
Reserved	11–255

The Reporting Detail subelement contains a 1-octet Reporting Detail data field that defines the level of detail per AP to be reported to the requesting STA. The Reporting Detail values are defined in Table 8-67.

Table 8-67—Reporting Detail values

Level of detail requested	Reporting Detail
No fixed-length fields or elements	0
All fixed-length fields and any requested elements in the Request element if present	1
All fixed-length fields and elements (default, used when Reporting Detail subelement is not included in Beacon Request)	2
Reserved	3–255

The indicated Reporting Detail applies individually to each measured Beacon, Measurement Pilot, or Probe Response. If the Reporting Detail equals 1, a Request element is optionally present in the optional subelements field. If included, the Request element lists the Element IDs of the elements requested to be reported in the Reported Frame Body of the Beacon Report.

The Request, AP Channel Report, and Vendor Specific subelements have the same format as their corresponding elements (see 8.4.2.13, 8.4.2.38, and 8.4.2.28, respectively). Multiple AP Channel Report and Vendor Specific subelements can be included in the list of optional subelements.

If one or more AP Channel Report elements are included, they indicate that iterative measurements are requested first on the channel(s) indicated by the Operating Class and Channel Number fields included in the Beacon Request, and second on the channel(s) indicated by the Operating Class and Channel List fields of each AP Channel Report element included in the Beacon Request. The procedures for iterative measurements on multiple channels are described in 10.11.9.1.

8.4.2.23.8 Frame request

The Measurement Request field corresponding to a frame request is shown Figure 8-115.

	Operating Class	Channel Number	Randomization Interval	Measurement Duration	Frame Request Type	MAC Address	Optional subelements
Octets:	1	1	2	2	1	6	variable

Figure 8-115—Measurement Request field format for frame request

Operating Class indicates the channel set for which the measurement request applies. Country, Operating Class, and Channel Number together specify the channel frequency and spacing for which the measurement request applies. Valid values of Operating Class are shown in Annex E.

Channel Number indicates the channel number for which the measurement request applies. Channel Number is defined within an Operating Class as shown in Annex E.

Randomization Interval specifies the upper bound of the random delay to be used prior to making the measurement, expressed in units of TUs. See 10.11.3.

The Measurement Duration field is set to the preferred or mandatory duration of the requested measurement, expressed in units of TUs. See 10.11.4.

The Frame Request Type indicates which subelements are requested in the Frame Report. The value of 1 signifies that a Frame Count Report is requested. The values 0 and 2 to 255 are reserved.

If the MAC Address field is the broadcast address, then all frames are counted towards the Frame Report generated in response to this frame request. For other MAC addresses, only frames matching this MAC address as the Transmitter Address are counted towards the Frame Report generated in response to this frame request.

The Optional Subelements field format contains zero or more subelements, each consisting of a 1-octet Subelement ID field, a 1-octet Length field, and a variable-length Data field, as shown in Figure 8-402. Any optional subelements are ordered by nondecreasing subelement ID.

The Subelement ID field values for the defined optional subelements are shown in Table 8-68. A Yes in the Extensible column of a subelement listed in Table 8-68 indicates that the Length of the subelement might be extended in future revisions or amendments of this standard. When the Extensible column of an element is equal to Subelements, then the subelement might be extended in future revisions or amendments of this standard by defining additional subelements within the subelement. See 9.24.9.

The Vendor Specific subelements have the same format as their corresponding elements (see 8.4.2.28). Multiple Vendor Specific subelements are optionally present in the list of optional subelements.

Table 8-68—Optional subelement IDs for frame request

Subelement ID	Name	Length field (octets)	Extensible
0–220	Reserved		
221	Vendor Specific	1 to 237	
222–255	Reserved		

8.4.2.23.9 STA Statistics Request

The Measurement Request field corresponding to a STA Statistics Request is shown in Figure 8-116.

	Peer MAC Address	Randomization Interval	Measurement Duration	Group Identity	Optional Subelements
Octets:	6	2	2	1	variable

Figure 8-116—Measurement Request field format for STA Statistics Request

The Peer MAC Address field is the RA or TA MAC address for the frame statistics of this measurement.

Randomization Interval specifies the upper bound of the random delay to be used prior to making the measurement, expressed in units of TUs. See 10.11.3.

The Measurement Duration field is set to the duration of the requested measurement in TUs except when triggered reporting is used. When triggered reporting is used, the measurement duration is 0.

Group Identity indicates the requested statistics group according to Table 8-69.

Table 8-69—Group Identity for a STA Statistics Request

Statistics Group Name	Group Identity
STA Counters from dot11CountersTable	0
STA Counters from dot11MacStatistics group	1
QoS STA Counters for UP0 from dot11QosCountersTable	2
QoS STA Counters for UP1 from dot11QosCountersTable	3
QoS STA Counters for UP2 from dot11QosCountersTable	4
QoS STA Counters for UP3 from dot11QosCountersTable	5
QoS STA Counters for UP4 from dot11QosCountersTable	6
QoS STA Counters for UP5 from dot11QosCountersTable	7
QoS STA Counters for UP6 from dot11QosCountersTable	8
QoS STA Counters for UP7 from dot11QosCountersTable	9
BSS Average Access Delays as described in 8.4.2.41 and 8.4.2.46	10
STA Counters from dot11CountersGroup3 (A-MSDU)	11
STA Counters from dot11CountersGroup3 (A-MPDU)	12
STA Counters from dot11CountersGroup3 (BlockAckReq, Channel Width, PSMP)	13
STA Counters from dot11CountersGroup3 (RD, dual CTS, L-SIG TXOP protection)	14
STA Counters from dot11CountersGroup3 (beamforming and STBC)	15
RSNA Counters	16
Reserved	17–255

The Optional Subelements field format contains zero or more subelements, each consisting of a 1-octet Subelement ID field, a 1-octet Length field, and a variable-length Data field, as shown in Figure 8-402. Any optional subelements are ordered by nondecreasing subelement ID.

The Subelement ID field values for the defined optional subelements are shown in Table 8-70. A Yes in the Extensible column of a subelement listed in Table 8-70 indicates that the Length of the subelement might be extended in future revisions or amendments of this standard. When the Extensible column of an element is equal to Subelements, then the subelement might be extended in future revisions or amendments of this standard by defining additional subelements within the subelement. See 9.24.9.

Table 8-70—Optional subelement IDs for STA Statistics Request

Subelement ID	Name	Length field (octets)	Extensible
0	Reserved		
1	Triggered Reporting	variable	
2–220	Reserved		
221	Vendor Specific	1 to 239	
222–255	Reserved		

The Triggered Reporting subelement is used to specify trigger conditions and thresholds for triggered STA Statistics measurements. It is present when setting up triggered reporting from STA Counters, QoS STA Counters, or RSNA Counters; see 10.11.9.5.

The format of the Triggered Reporting subelement for STA Counters is shown in Figure 8-117. The fields marked as optional are only present if the appropriate bit in the STA Counter Trigger Condition field is 1.

	Measurement Count	Trigger Timeout	STA Counter Trigger Condition	dot11FailedCount Threshold (optional)	dot11FCS ErrorCount Threshold (optional)
Octets:	4	2	2	0 or 4	0 or 4

	dot11Multiple RetryCount Threshold (optional)	dot11Frame DuplicateCount Threshold (optional)	dot11RTS FailureCount Threshold (optional)	dot11ACK FailureCount Threshold (optional)	dot11RetryCount Threshold (optional)
Octets:	0 or 4	0 or 4	0 or 4	0 or 4	0 or 4

Figure 8-117—Triggered Reporting subelement for STA Counters

The value in the Measurement Count field specifies the number of MSDUs or MPDUs transmitted and/or received by the STA that are to be used to determine if one or more of the trigger conditions have been met.

The Trigger Timeout field contains a value in units of 100 TUs during which a measuring STA does not generate further triggered STA Statistics Reports after a trigger condition has been met.

The STA Counter Trigger Condition field specifies trigger values used when requesting triggered STA Statistics reporting. The format of the STA Counter Trigger Condition field is shown in Figure 8-118.

	B0	B1	B2	B3
	dot11FailedCount	dot11FCSError Count	dot11Multiple RetryCount	dot11Frame DuplicateCount
Bits	1	1	1	1

	B4	B5	B6	B7 B15
	dot11RTS FailureCount	dot11ACK FailureCount	dot11RetryCount	Reserved
Bits	1	1	1	9

Figure 8-118—STA Counter Trigger Condition field

For each bit in the STA Counter Trigger Condition field that is 1, a corresponding threshold value exists (defined in Figure 8-117) in the Triggered Reporting subelement for STA Counters. With this, the STA Statistics Request indicates that a STA Statistics Report be generated when the corresponding STA counter defined in 8-154 and 8-155 (in 8.4.2.24.9) exceeds the value of the specified threshold, within the total number of MSDUs or MPDUs indicated in the Measurement Count field. See 10.11.9.5. One or more trigger conditions are set with specified thresholds. In the triggered STA Statistics request, the value of the Group Identity field is either 0 or 1. When the Group Identity field value of the triggered STA Statistics request is 0, B2–B6 in the STA Counter Trigger Condition field are set to 0. When the group identity of the triggered STA Statistics request is 1, B0 and B1 in the STA Counter Trigger Condition field are set to 0.

The format of the Triggered Reporting subelement for QoS STA Counters is shown in Figure 8-119. The fields marked as optional are only present if the appropriate bit in the QoS STA Counter Trigger Condition is 1.

	Measurement Count	Trigger Timeout	QoS STA Counter Trigger Condition	dot11QoSFailed Count Threshold (optional)	dot11QoSRetry Count Threshold (optional)
Octets:	4	2	2	0 or 4	0 or 4

	dot11QoSMultiple RetryCount Threshold (optional)	dot11QoSFrame DuplicateCount Threshold (optional)	dot11QoSRTSC ount Failure Threshold (optional)	dot11QoSACK FailureCount Threshold (optional)	dot11QoSDisca rdedCount Threshold (optional)
Octets:	0 or 4	0 or 4	0 or 4	0 or 4	0 or 4

Figure 8-119—Triggered Reporting subelement for QoS STA Counters

The UP of the QoS STA Counters for triggered QoS Statistics measurement is determined by the group identity of the measurement request field for a STA Statistics Request frame as defined in 8-69.

The value in the Measurement Count field specifies the number of MSDUs or MPDUs transmitted and/or received by the STAs that are to be used to determine if one or more of the trigger conditions have been met.

The Trigger Timeout field contains a value in units of 100 TUs during which a measuring STA does not generate further triggered STA Statistics Reports after a trigger condition has been met.

The QoS STA Counter Trigger Condition field specifies reporting triggers when requesting triggered STA Statistics reporting. The format of the QoS STA Counter Trigger Condition field is shown in Figure 8-120.

	B0	B1	B2	B3
	dot11QoS FailedCount	dot11QoS RetryCount	dot11QoSMultiple RetryCount	dot11QoSFrame DuplicateCount
Bits:	1	1	1	1

	B4	B5	B6	B7-B15
	dot11QoSRTS FailureCount	dot11QoSACK FailureCount	dot11QoS DiscardedCount	Reserved
Bits:	1	1	1	9

Figure 8-120—QoS STA Counter Trigger Condition field

For each bit in the QoS STA Counter Trigger Condition field that is 1, a corresponding threshold value exists (defined in Figure 8-119) in the Triggered Reporting subelement for QoS STA Counters. With this, the STA Statistics Request indicates that a STA Statistics Report be generated when the corresponding QoS STA counter defined in Figure 8-156 (in 8.4.2.24.9) exceeds the value of the specified threshold, within the total number of MSDUs or MPDUs indicated in the Measurement Count field. See 10.11.9.5. One or more trigger conditions are set with specified thresholds.

The format of the Triggered Reporting subelement for RSNA Counters is shown in Figure 8-121. The fields marked as optional are only present if the appropriate bit in the RSNA Trigger Condition is 1.

	Measurement Count	Trigger Timeout	RSNA Counter Trigger Condition	dot11RSNAStats CMACICVErrors Threshold (optional)	dot11RSNA StatsCMACReplays Threshold (optional)
Octets:	4	2	2	0 or 4	0 or 4

	dot11RSNA StatsRobustMgmt CCMPReplays Threshold (optional)	dot11RSNAStats TKIPICVErrors Threshold (optional)	dot11RSNAStats TKIPReplays Threshold (optional)	dot11RSNAStats CCMPDecryptErrors Threshold (optional)	dot11RSNAStats CCMPReplays Threshold (optional)
Octets:	0 or 4	0 or 4	0 or 4	0 or 4	0 or 4

Figure 8-121—Triggered Reporting subelement for RSNA Counters

The value in the Measurement Count field specifies the number of MPDUs transmitted and/or received by the STA that are to be used to determine if one or more of the trigger conditions have been met.

The Trigger Timeout field contains a value in units of 100 TUs during which a measuring STA does not generate further triggered STA Statistics Reports after a trigger condition has been met.

The RSNA Counter Trigger Condition field specifies reporting triggers when requesting triggered STA Statistics reporting. The format of the RSNA Trigger Condition field is shown in Figure 8-122.

	B0	B1	B2	B3
	dot11RSNAStats CMACICVErrors	dot11RSNA StatsCMACReplays	dot11RSNAStats RobustMgmt CCMPReplays	dot11RSNAStats TKIPICVErrors
Bits:	1	1	1	1

	B4	B5	B6	B7 B15
	dot11RSNAStats TKIPReplays	dot11RSNAStats CCMPDecryptErrors	dot11RSNAStats CCMPReplays	Reserved
Bits:	1	1	1	9

Figure 8-122—RSNA Trigger Condition field

For each bit in the RSNA Trigger Condition field that is 1, a corresponding threshold value exists (defined in Figure 8-121) in the Triggered Reporting subelement for RSNA Counters. With this, the STA Statistics Request indicates that a STA Statistics Report be generated when the corresponding RSNA counter defined in Figure 8-158 (in 8.4.2.24.9) exceeds the value of the specified threshold, within the total number of MPDUs indicated in the Measurement Count field. See 10.11.9.5. One or more trigger conditions are set with specified thresholds.

The Vendor Specific subelements have the same format as their corresponding elements (see 8.4.2.28). Multiple Vendor Specific subelements are optionally present in the list of optional subelements.

8.4.2.23.10 Location Configuration Information Request

The Measurement Request field corresponding to an LCI request is shown in Figure 8-123.

	Location Subject	Latitude Requested Resolution	Longitude Requested Resolution	Altitude Requested Resolution	Optional Subelements
Octets:	1	1	1	1	variable

Figure 8-123—Measurement Request field format for LCI Request

The Location Subject field of a LCI request is a single octet. See Table 8-71.

Table 8-71—Location subject definition

Value	Location Subject
0	Location Subject Local
1	Location Subject Remote
2	Location Subject Third Party
3–255	Reserved

The term Local refers to the location of the requesting STA, and Remote refers to the location of the reporting STA.

NOTE—Local Measurement Request is used by requesting STA to obtain its own location, asking “Where am I?” Remote Measurement Request is used by requesting STA to obtain the location of the reporting STA, asking “Where are you?”

Latitude Requested Resolution is the number of valid most significant bits requested for the fixed-point value of Latitude in degrees. Values above 34 (decimal), the specified maximum number of bits of Latitude, are reserved.

Longitude Requested Resolution is the number of valid most significant bits requested for the fixed-point value of Longitude in degrees. Values above 34 (decimal), the specified maximum number of bits of Longitude, are reserved.

Altitude Requested Resolution is the number of valid most significant bits requested for the Altitude, which has either of two types, as described in 8.4.2.24.10. Values above 30 (decimal), the specified maximum number of bits of Altitude, are reserved.

The Optional Subelements field format contains zero or more subelements, each consisting of a 1-octet Subelement ID field, a 1-octet Length field, and a variable-length Data field, as shown in Figure 8-402. Any optional subelements are ordered by nondecreasing subelement ID.

The Subelement ID field values for the defined optional subelements are shown in Table 8-72. A Yes in the Extensible column of a subelement listed in Table 8-72 indicates that the Length of the subelement might be extended in future revisions or amendments of this standard. When the Extensible column of an element is equal to Subelements, then the subelement might be extended in future revisions or amendments of this standard by defining additional subelements within the subelement. See 9.24.9.

Table 8-72—Optional subelement IDs for LCI Request

Subelement ID	Name	Length field (octets)	Extensible
0	Reserved		
1	Azimuth Request	1	Yes
2	Originator Requesting STA MAC Address	6	No
3	Target MAC Address	6	No
4–220	Reserved		
221	Vendor Specific	1 to 246	
222–255	Reserved		

The Azimuth Request subelement is present when requesting azimuth information. The Azimuth Request subelement is as shown in Figure 8-124.

	Subelement ID	Length	Azimuth Request
Octets:	1	1	1

Figure 8-124—Azimuth Request subelement format

The value of the subelement ID is equal to the Azimuth Request value in Table 8-72.

The Length field value is set to 1.

The Azimuth Request field of an Azimuth Request subelement is shown in Figure 8-125.

	B0 B3	B4	B5 B7
	Azimuth Resolution Requested	Azimuth Type	Reserved
Bits:	4	1	3

Figure 8-125—Azimuth Request field

Azimuth Resolution Requested is the number of valid most significant bits requested for the fixed-point value of Azimuth, reported in integer degrees. Values above 9 are reserved.

Azimuth Type (bit 4) is set to 1 to request a report of the Azimuth of radio reception and is set to 0 to request a report of the Azimuth of front surface of the reporting STA.

NOTE—A geographic feature is an abstraction of a real-world phenomenon; it is a geographic feature if it is associated with a location relative to the Earth. The designation of a horizontal plane is relative to the Earth. The designation of the “front surface” of a station is arbitrary, but refers to an orientable surface (possessing a centerline) of the station. It is common to use a direction cosine matrix to convert from one coordinate system to another, i.e., body-centered coordinates to earth-centered coordinates.

The Originator Requesting STA MAC Address subelement contains the MAC address of the STA requesting the Location Information and it is present whenever the location subject definition field is set to 2. The format of the Originator Requesting STA MAC Address subelement is shown in Figure 8-126.

	Subelement ID	Length	Originator Requesting STA MAC Address
Octets:	1	1	6

Figure 8-126—Originator Requesting STA MAC Address subelement format

The Target MAC Address subelement contains the MAC address of the STA whose Location Information is requested and it is present whenever the location subject definition field is set to 2. The format of the Target MAC address subelement is shown in Figure 8-127.

	Subelement ID	Length	Originator Requesting STA MAC Address
Octets:	1	1	6

Figure 8-127—Target MAC Address subelement format

The Vendor Specific subelement has the same format as the Vendor Specific element (see 8.4.2.28). Multiple Vendor Specific subelements are optionally present in the list of optional subelements.

8.4.2.23.11 Transmit Stream/Category Measurement Request

The Transmit Stream/Category Measurement applies to TIDs for Traffic Streams associated with TSPECs and also to TIDs for Traffic Categories for QoS traffic without TSPECs. The Measurement Request field corresponding to a Transmit Stream/Category Measurement Request is shown in Figure 8-128.

	Randomization Interval	Measurement Duration	Peer STA Address	Traffic Identifier	Bin 0 Range	Optional Subelements
Octets:	2	2	6	1	1	variable

Figure 8-128—Measurement Request field format for Transmit Stream/Category Measurement Request

Randomization Interval is set to the desired maximum random delay in the measurement start time, expressed in units of TUs. The use of Randomization Interval is described in 10.11.3. When requesting a triggered Transmit Stream/Category Measurement, Randomization Interval is not used and is set to 0. See 10.11.9.8.

The Measurement Duration is set to the duration of the requested measurement, expressed in units of TUs except when setting up a triggered QoS measurement, when it is not used and is set to 0.

The Peer STA Address contains a MAC address indicating the RA in the MSDUs to be measured.

The Traffic Identifier field contains the TID subfield as shown in Figure 8-129.

	B0 B3	B4 B7
	Reserved	TID
Bits:	4	4

Figure 8-129—Traffic Identifier field

The TID subfield indicates the TC or TS for which traffic is to be measured.

Bin 0 Range indicates the delay range of the first bin (Bin 0) of the Transmit Delay Histogram, expressed in units of TUs. The Bin 0 Range value is used to calculate the delay ranges of the other 5 bins making up the histogram. The delay range for each bin increases in a binary exponential fashion as described in 8.4.2.24.11.

The Optional Subelements field format contains zero or more subelements, each consisting of a 1-octet Subelement ID field, a 1-octet Length field, and a variable-length Data field, as shown in Figure 8-402. The optional subelements are ordered by nondecreasing subelement ID.

The Subelement ID field values for the defined optional subelements are shown in Table 8-73. A Yes in the Extensible column of a subelement listed in Table 8-73 indicates that the Length of the subelement might be extended in future revisions or amendments of this standard. When the Extensible column of an element is equal to Subelements, then the subelement might be extended in future revisions or amendments of this standard by defining additional subelements within the subelement. See 9.24.9.

Table 8-73—Optional subelement IDs for Transmit Stream/Category Measurement Request

Subelement ID	Name	Length field (octets)	Extensible
0	Reserved		
1	Triggered Reporting	6	Yes
2–220	Reserved		
221	Vendor Specific	1 to 238	
222–255	Reserved		

The Triggered Reporting subelement is used to specify measurement trigger thresholds. It is present only if requesting triggered transmit stream/category measurement reporting. The Triggered Reporting subelement field format is shown in Figure 8-130.

	Subelement ID	Length	Triggered Reporting
Octets:	1	1	6

Figure 8-130—Triggered Reporting subelement format

The value of the subelement ID is equal to the Triggered Reporting value in Table 8-73.

The value of the Length field in octets is equal to 6.

The Triggered Reporting field is as shown in Figure 8-131.

	Trigger Conditions	Average Error Threshold	Consecutive Error Threshold	Delay Threshold	Measurement Count	Trigger Time-out
Octets:	1	1	1	1	1	1

Figure 8-131—Triggered Reporting field

Trigger Conditions is a bit-field that specifies reporting triggers when requesting a triggered transmit stream/category measurement. The format of the Trigger Conditions bit-field is shown in Figure 8-132.

	B0	B1	B2	B3 B7
	Average	Consecutive	Delay	Reserved
Bits:	1	1	1	5

Figure 8-132—Trigger Conditions bit-field

- Average is set to 1 to request that a Transmit Stream/Category Measurement Report be generated when the number of MSDUs for the TC or TS given by the TID that are discarded out of the number of preceding MSDUs specified in Measurement Count is greater than or equal to the value given in Average Error Threshold. MSDUs discarded due to the number of transmit attempts exceeding dot11ShortRetryLimit or dot11LongRetryLimit, or due to the MSDU lifetime having been reached, are counted.
- Consecutive is set to 1 to request that a Transmit Stream/Category Measurement Report be generated when the number of MSDUs for the TC or TS given by the TID that are discarded in succession is greater than or equal to the value given in Consecutive Error Threshold. MSDUs discarded due to the number of transmit attempts exceeding dot11ShortRetryLimit or dot11LongRetryLimit, or due to the MSDU lifetime having been reached, are counted.
- Delay is set to 1 to request that a Transmit Stream/Category Measurement Report be generated when the number of consecutive MSDUs for the TC, or TS given by the TID that experience a transmit delay greater than or equal to the value specified in the Delay Threshold subfield, is greater than or equal to the value given in Delayed MSDU Count. Delay is measured from the time the MSDU is passed to the MAC until the point at which the entire MSDU has been successfully transmitted, including receipt of the final ACK from the peer STA if the QoSAck service class is being used.

The Average Error Threshold field contains a value representing the number of discarded MSDUs to be used as the threshold value for the Average trigger condition. The field is reserved if the Average Error Threshold subfield of the Trigger Conditions bit-field is 0.

The Consecutive Error Threshold field contains a value representing the number of discarded MSDUs to be used as the threshold value for the Consecutive trigger condition. The field is reserved if the Consecutive Error Threshold subfield of the Trigger Conditions bit-field is 0.

The Delay Threshold field contains two subfields as shown in Figure 8-133. The Delay Threshold field is reserved if the Delay Threshold subfield of the Trigger Conditions bit-field is 0.

	B0 B1	B2 B7
	Delayed MSDU Range	Delayed MSDU Count
Bits:	2	6

Figure 8-133—Delay Threshold subfield

Delayed MSDU Range contains a value representing the MSDU transmit delay at or above which an MSDU is counted towards the Delayed MSDU Count threshold. Delayed MSDU Range is encoded as a value representing the lower bound of a bin in the Transmit Delay Histogram as shown in Table 8-74. The Transmit Delay Histogram is defined in 8.4.2.24.11.

Table 8-74—Delayed MSDU Range Definitions

Delayed MSDU Range	Condition
0	Transmit Delay = Lower Bound of Bin 2
1	Transmit Delay = Lower Bound of Bin 3
2	Transmit Delay = Lower Bound of Bin 4
3	Transmit Delay = Lower Bound of Bin 5

Delayed MSDU Count contains a value representing the number of MSDUs to be used as the threshold value for the Delay trigger condition.

The Measurement Count field contains a number of MSDUs. This value is used to calculate an average discard count for the Average trigger condition. It is also used in place of measurement duration in determining the scope of the reported results when a report is triggered; see 10.11.9.8.

The Trigger Timeout field contains a value, expressed in units of 100 TU, during which a measuring STA does not generate further triggered transmit stream/category measurement reports after a trigger condition has been met. See 10.11.9.8.

The Vendor Specific subelement has the same format as the Vendor Specific element (see 8.4.2.28). Multiple Vendor Specific subelements are optionally present in the list of optional subelements.

8.4.2.23.12 Measurement pause request

The Measurement Request field corresponding to a measurement pause request is shown in Figure 8-134. The measurement pause request cannot be processed in parallel with any other Measurement Request. See 10.11.9.7.

	Pause Time	Optional Subelements
Octets:	2	variable

Figure 8-134—Measurement Request field format for measurement pause request

The Pause Time field contains a number between 1 and 65 535 representing the time period for which measurements are suspended or paused. The time unit for the Pause Time is 10 TUs. The Pause Time value 0 is reserved. Measurement pause requests are used to provide time delays between the execution times of measurement request elements in a Measurement Request frame.

The Optional Subelements field format contains zero or more subelements, each consisting of a 1-octet Subelement ID field, a 1-octet Length field, and a variable-length Data field, as shown in Figure 8-402. Any optional subelements are ordered by nondecreasing subelement ID.

The Subelement ID field values for the defined optional subelements are shown in Table 8-75. A Yes in the Extensible column of a subelement listed in Table 8-75 indicates that the Length of the subelement might be extended in future revisions or amendments of this standard. When the Extensible column of an element is equal to Subelements, then the subelement might be extended in future revisions or amendments of this standard by defining additional subelements within the subelement. See 9.24.9.

Table 8-75—Optional subelement IDs for measurement pause request

Subelement ID	Name	Length field (octets)	Extensible
0–220	Reserved		
221	Vendor Specific	1 to 248	
222–255	Reserved		

The Vendor Specific subelements have the same format as their corresponding elements (see 8.4.2.28). Multiple Vendor Specific subelements are optionally present in the list of optional subelements.

8.4.2.23.13 Multicast Diagnostics Request

The Measurement Request field corresponding to a Multicast Diagnostics Request is shown in Figure 8-135. The response to a Multicast Diagnostics Request is a Multicast Diagnostics Report.

	Randomization Interval	Measurement Duration	Group MAC Address	Multicast Triggered Reporting (optional)	Optional Subelements
Octets:	2	2	6	0 or 5	variable

Figure 8-135—Measurement Request field format for a Multicast Diagnostics Request

The Randomization Interval field is the desired upper limit of random delay before the measurement begins, expressed in TUs. The use of the Randomization Interval is described in 10.12.3. When requesting a triggered multicast diagnostic report, the Randomization Interval field is reserved.

The Measurement Duration field is the duration of the requested measurement, expressed in TUs. When requesting a triggered multicast diagnostic report, the Measurement Duration field is reserved.

A Group MAC Address field with the LSB of the first octet set to 1 contains the MAC address of the group addressed frames to which the measurement request relates. A Group MAC Address field with the LSB of the first octet set to 0 indicates that all group addressed frames, apart from the broadcast MAC address, are requested.

The Multicast Triggered Reporting field is used to specify trigger conditions and thresholds. It is only present when requesting triggered multicast diagnostic reporting. The format of Multicast Triggered Reporting subelement is as shown in Figure 8-136.

	Subelement ID	Length	Multicast Trigger Condition	Inactivity Timeout	Reactivation Delay
Octets:	1	1	1	1	1

Figure 8-136—Multicast Triggered Reporting subelement format

The Multicast Trigger Condition field specifies reporting triggers for triggered management diagnostic reporting. The format of the Multicast Trigger Condition field is shown in Figure 8-137.

	B0	B1 B7
	Inactivity Timeout Request	Reserved
Bits:	1	7

Figure 8-137—Multicast Trigger Condition field

The Inactivity Timeout Request field is 1 to request that a Multicast Triggered Report be generated when no group addressed frames with the monitored group address are received in a time equal to the value given in the Inactivity Timeout field. The Inactivity Timeout Request field is 0 when a multicast reception timeout is not requested.

The Inactivity Timeout field contains a time value in units of 100 TUs to be used as the threshold value for the Inactivity Timeout trigger condition.

The Reactivation Delay field contains a value in units of 100 TUs during which a measuring STA does not generate further Multicast Triggered Reports after a trigger condition has been met.

The Optional Subelements field format contains zero or more Subelements, each consisting of a 1-octet Subelement ID field, a 1-octet Length field and a variable-length Data field, as shown in Figure 8-402. The optional subelements are ordered by nondecreasing subelement ID.

The Subelement ID field values for the defined optional subelements are shown in Table 8-76. A Yes in the Extensible column of a subelement listed in Table 8-76 indicates that the length of the subelement might be extended in future revisions or amendments of this standard. When the Extensible column of an element is Subelements, then the subelement might be extended in future revisions or amendments of this standard by defining additional subelements within the subelement. See 9.24.9.

The use of Multicast Diagnostics Request is defined in 10.11.19.

Table 8-76—Optional subelement IDs for STA Multicast Diagnostics Request

Subelement ID	Name	Length field (octets)	Extensible
0	Reserved		
1	Multicast Triggered Reporting	5	
2–220	Reserved		
221	Vendor Specific	1 to 239	
222–255	Reserved		

8.4.2.23.14 Location Civic Request

The Measurement Request field corresponding to a Location Civic Request is shown in Figure 8-138. The response to a Location Civic Request is a Location Civic Report.

	Location Subject	Civic Location Type	Location Service Interval Units	Location Service Interval	Optional Subelements
Octets:	1	1	1	2	variable

Figure 8-138—Location Civic Request field format

The Location Subject field is a single octet and is defined in Table 8-71.

The Civic Location Type field contains the format of location information in the Location Civic Report, as indicated in Table 8-77.

Table 8-77—Civic Location Type

Civic Location Type value	Description
0	IETF RFC4776-2006; includes all subsequent RFCs that define additional civic address Types
1	Vendor Specific
2–255	Reserved

When the Civic Location Type value is Vendor Specific, a Vendor Specific subelement is included in the Optional Subelements field that identifies the Organization Identifier corresponding to the Civic Location Type.

The Location Service Interval Units field contains the units for the Location Service Interval field, as indicated in Table 8-78.

Table 8-78—Location Service Interval Units

Location Service Interval Units value	Description
0	Seconds
1	Minutes
2	Hours
3–255	Reserved

The Location Service Interval field is the time interval, expressed in the units indicated in the Location Service Interval Units field, at which the STA requests to receive Location Civic Reports. A Location Service Interval of 0 indicates that only a single Location Civic Report is requested.

The Optional Subelements field format contains zero or more subelements, each consisting of a 1-octet Subelement ID field, a 1-octet Length field and a variable-length Data field, as shown in Figure 8-402. The optional subelements are ordered by nondecreasing subelement ID.

The Subelement ID field values for the defined optional subelements are shown in Table 8-79. A Yes in the Extensible column of a subelement listed in Table 8-79 indicates that the length of the subelement might be extended in future revisions or amendments of this standard. When the Extensible column of an element is Subelement, then the subelement might be extended in future revisions or amendments of this standard by defining additional subelements within the subelement. See 9.24.9.

Table 8-79—Optional subelement IDs for Location Civic Request

Subelement ID	Name	Length field (octets)	Extensible
0	Reserved		
1	Originator Requesting STA MAC Address	6	No
2	Target MAC Address	6	No
3–220	Reserved		
221	Vendor Specific	1 to 237	
222–255	Reserved		

The Originator Requesting STA MAC Address subelement contains the MAC address of the STA requesting for the Location Information and it is present whenever the location subject definition field is set to 2. The format of the Originator Requesting STA MAC Address subelement is shown in Figure 8-126.

The Target MAC Address subelement contains the MAC address of the STA whose Location Information is requested and it is present whenever the location subject definition field is set to 2. The format of the Target MAC address subelement is shown in Figure 8-127.

8.4.2.23.15 Location Identifier Request

The Measurement Request field corresponding to a Location Identifier Request is shown in Figure 8-139. The response to a Location Identifier Request is a Location Identifier Report.

	Location Subject	Location Service Interval Units	Location Service Interval	Optional Subelements
Octets:	1	1	2	variable

Figure 8-139—Location Identifier Request field format

The Location Subject field is a single octet and is defined in Table 8-71.

The Location Service Interval Units field is defined in Table 8-78.

The Location Service Interval field is the time interval, expressed in the units indicated in the Location Service Interval Units field, at which the STA requests to receive Location Identifier Reports. A Location Service Interval of 0 indicates that only a single Location Identifier Report is requested.

The Optional Subelements field format contains zero or more subelements, each consisting of a 1-octet Subelement ID field, a 1-octet Length field and a variable-length Data field, as shown in Figure 8-402. Any optional subelements are ordered by nondecreasing subelement ID.

The Subelement ID field values for the defined optional subelements are shown in Table 8-80. A Yes in the Extensible column of a subelement listed in Table 8-80 indicates that the Length of the subelement might be extended in future revisions or amendments of this standard. When the Extensible column of an element is Subelement, then the subelement might be extended in future revisions or amendments of this standard by defining additional subelements within the subelement. See 9.24.9.

Table 8-80—Optional subelement IDs for Location Identifier Request

Subelement ID	Name	Length field (octets)	Extensible
0	Reserved		
1	Originator Requesting STA MAC Address	6	No
2	Target MAC Address	6	No
3–220	Reserved		
221	Vendor Specific	1 to 237	
222–255	Reserved		

The Originator Requesting STA MAC Address subelement contains the MAC address of the STA requesting the Location Information and it is present whenever the location subject definition field is set to 2. The format of the Originator Requesting STA MAC Address subelement is shown in Figure 8-126.

The Target MAC Address subelement contains the MAC address of the STA whose Location Information is requested and it is present whenever the location subject definition field is set to 2. The format of the Target MAC address subelement is shown in Figure 8-127.

8.4.2.24 Measurement Report element

8.4.2.24.1 General

The Measurement Report element contains a measurement report. The format of the Measurement Report element is shown in Figure 8-140. The Measurement Report element is included in Spectrum Management Measurement Report frames, as described in 8.5.2.3, or Radio Measurement Report frames, as described in 8.5.7.3. Measurement Types 0, 1, and 2 are used for spectrum management and are included only in Spectrum Management Measurement Report frames. All other Measurement Types are used for radio measurement and are included only in Radio Measurement Report frames. The use of Measurement Report elements and frames for spectrum management is described in 10.9.7. The use of Measurement Report elements and frames for radio measurement is described in 10.11.

	Element ID	Length	Measurement Token	Measurement Report Mode (see Figure 8-141)	Measurement Type	Measurement Report
Octets:	1	1	1	1	1	variable

Figure 8-140—Measurement Report element format

	B0	B1	B2	B3 B7
	Late	Incapable	Refused	Reserved
Bits:	1	1	1	5

Figure 8-141—Measurement Report Mode field

The value of the Length field in octets is variable and depends on the length of the Measurement Report field. The minimum value of the Length field is 3.

The Measurement Token field is set to the Measurement Token in the corresponding Measurement Request element. If the Measurement Report element is being sent autonomously, then the Measurement Token is set to 0. If the Measurement Report element is being sent in a Location Track Notification frame then the Measurement Token is set to the same value as the Location Configuration Request frame Dialog Token field that configured the STA to send the location track notification frames.

The Measurement Report Mode field (shown in Figure 8-141) is used to indicate the reason for a failed or rejected measurement request. The Measurement Report Mode is a bit field with the following bits defined:

- Late bit (bit 0) indicates whether this STA is unable to carry out a measurement request because it received the request after the requested measurement time. The Late bit is set to 1 to indicate the request was too late. The Late bit is set to 0 to indicate the request was received in time for the measurement to be executed. The Late bit only applies to spectrum management measurement and is set to 0 in all measurement report elements for radio measurement types (see Table 8-81).
- Incapable bit (bit 1) indicates whether this STA is incapable of generating a report of the type specified in the Measurement Type field that was previously requested by the destination STA of this Measurement Report element. The Incapable bit is set to 1 to indicate the STA is incapable. The Incapable bit is set to 0 to indicate the STA is capable or the report is autonomous.
- Refused bit (bit 2) indicates whether this STA is refusing to generate a report of the type specified in the Measurement Type field that was previously requested by the destination STA of this Measurement Report element. The Refused bit is set to 1 to indicate the STA is refusing. The Refused bit is set to 0 to indicate the STA is not refusing or the report is autonomous.

— All other bits are reserved.

Table 8-81—Measurement Type definitions for measurement reports

Name	Measurement Type	Measurement Use
Basic report	0	Spectrum Management
CCA report	1	
RPI histogram report	2	
Channel load report	3	Radio Measurement
Noise histogram report	4	
Beacon report	5	
Frame report	6	
STA statistics report	7	Radio Measurement and WNM
LCI report	8	Radio Measurement, Spectrum Management, and WNM
Transmit stream/category measurement report	9	Radio Measurement
Multicast diagnostics report	10	WNM
Location Civic report	11	Radio Measurement and WNM
Location Identifier report	12	Radio Measurement and WNM
Reserved	13–255	N/A

No more than one bit is set to 1 within a Measurement Report Mode field. All bits within the Measurement Mode field are set to 0 if the results of a successful measurement request or an autonomous measurement are being reported.

The Measurement Type field is set to a number that identifies the measurement report. The Measurement Types that have been allocated are shown in Table 8-81.

The Measurement Report field is not present when the Late bit is equal to 1, the Incapable bit is equal to 1, or the Refused bit is equal to 1. Otherwise, it contains a single measurement report, as described in 8.4.2.24.2 to 8.4.2.24.11.

8.4.2.24.2 Basic report

A Measurement Type in the Measurement Report element may indicate a basic report. The format of the Measurement Report field corresponding to a basic report is shown in Figure 8-142. It is mandatory for a STA to support the generation of this report.

	Channel Number	Measurement Start Time	Measurement Duration	Map (see Figure 8-143)
Octets:	1	8	2	1

Figure 8-142—Measurement Report field format for a basic report

	BSS	Orthogonal frequency division multiplexing (OFDM) Preamble	Unidentified Signal	Radar	Unmeasured	Reserved
Bit:	0	1	2	3	4	5-7

Figure 8-143—Map field format

The Channel Number field is set to the channel number to which the basic report applies (as defined in 18.3.8.4.3).

The Measurement Start Time field is set to the TSF at the time (± 32 µs) at which the basic report measurement started.

The Measurement Duration field is set to the duration over which the basic report was measured, expressed in TUs.

The Map field is coded as a bit field, as shown in Figure 8-143, and contains the following bits:

- BSS bit, which is set to 1 when at least one valid MPDU was received in the channel during the measurement period from another BSS. Otherwise, the BSS bit is set to 0.
- OFDM preamble bit, which is set to 1 when at least one sequence of short training symbols, as defined in 18.3.3, was detected in the channel during the measurement period without a subsequent valid SIGNAL field (see 18.3.4). This may indicate the presence of an OFDM preamble, such as high-performance RLAN/2 (HIPERLAN/2). Otherwise, the OFDM preamble bit is set to 0.
- Unidentified Signal bit, which may be set to 1 when, in the channel during the measurement period, there is significant power detected that is not characterized as radar, an OFDM preamble, or a valid MPDU. Otherwise, the Unidentified Signal bit is set to 0. The definition of significant power is implementation dependent.
- Radar bit, which is set to 1 when radar was detected operating in the channel during the measurement period. The radar detection algorithm that satisfies regulatory requirements is outside the scope of this standard. Otherwise, the Radar bit is set to 0.
- Unmeasured bit, which is set to 1 when this channel has not been measured. Otherwise, the Unmeasured bit is set to 0. When the Unmeasured field is equal to 1, all the other bit fields are set to 0.

8.4.2.24.3 CCA report

A Measurement Type in the Measurement Report element may indicate a CCA report. It is optional for a STA to support the generation of this report. The format of the Measurement Report field corresponding to a CCA report is shown in Figure 8-144.

	Channel Number	Measurement Start Time	Measurement Duration	CCA Busy Fraction
Octets:	1	8	2	1

Figure 8-144—Measurement Report field format for a CCA report

The Channel Number field contains the channel number to which the CCA report applies (as defined in 18.3.8.4.3).

The Measurement Start Time field is set to the TSF at the time (± 32 µs) at which the CCA report measurement started.

The Measurement Duration field is set to the duration over which the CCA report was measured, expressed in TUs.

The CCA Busy Fraction field contains the fractional duration over which CCA indicated the channel was busy during the measurement duration. The resolution of the CCA busy measurement is in microseconds. The CCA Busy Fraction value is defined as Ceiling (255 × [Duration CCA indicated channel was busy (µs)] / (1024 × [Measurement duration (TUs)])).

8.4.2.24.4 RPI histogram report

A Measurement Type in the Measurement Report element may indicate an RPI histogram report. It is optional for a STA to support the generation of this report. The format of the Measurement Report field corresponding to an RPI histogram report is shown in Figure 8-145.

	Channel Number	Measurement Start Time	Measurement Duration
Octets:	1	8	2

	RPI 0 density	RPI 1 density	RPI 2 density	RPI 3 density	RPI 4 density	RPI 5 density	RPI 6 density	RPI 7 density
Octets:	1	1	1	1	1	1	1	1

Figure 8-145—Measurement Report field format for an RPI histogram report

The Channel Number field is set to the channel number to which the RPI histogram report applies (as defined in 18.3.8.4.3).

The Measurement Start Time field is set to the TSF at the time (± 32 µs) at which the RPI histogram report measurement started.

The Measurement Duration field is set to the duration over which the RPI histogram report was measured, expressed in TUs.

The RPI histogram report contains the RPI densities observed in the channel for the eight RPI levels defined in Table 8-82. See 10.11.12.

Table 8-82—RPI definitions for an RPI histogram report

RPI	Power observed at the antenna (dBm)
0	Power ≤ –87
1	–87 < Power ≤ –82
2	–82 < Power ≤ –77
3	–77 < Power ≤ –72
4	–72 < Power ≤ –67
5	–67 < Power ≤ –62
6	–62 < Power ≤ –57
7	–57 < Power

The RPI histogram report provides an additional mechanism for a STA to gather information on the state of a channel from other STAs. The STA may use this information to assist in the choice of new channel, to help avoid false radar detections, and to assess the general level of interference present on a channel.

8.4.2.24.5 Channel Load Report

The format of the Measurement Report field corresponding to a Channel Load Report is shown in Figure 8-146.

	Operating Class	Channel Number	Actual Measurement Start Time	Measurement Duration	Channel Load	Optional Subelements
Octets:	1	1	8	2	1	variable

Figure 8-146—Measurement Report field format for Channel Load Report

Operating Class indicates the channel set for which the measurement request applies. Country, Operating Class, and Channel Number together specify the channel frequency and spacing for which the measurement request applies. Valid values of Operating Class are shown in Annex E.

Channel Number indicates the channel number for which the measurement report applies. Channel Number is defined within an Operating Class as shown in Annex E.

Actual Measurement Start Time is set to the value of the measuring STA’s TSF timer at the time the measurement started.

Measurement Duration is set to the duration over which the Channel Load Report was measured, expressed in units of TUs.

Channel Load contains the proportion of measurement duration for which the measuring STA determined the channel to be busy. Procedure for Channel Load measurement and definition of channel load values are found in 10.11.9.3.

The Optional Subelements field format contains zero or more subelements, each consisting of a 1-octet Subelement ID field, a 1-octet Length field, and a variable-length Data field, as shown in Figure 8-402. Any optional subelements are ordered by nondecreasing subelement ID.

The Subelement ID field values for the defined optional subelements are shown in Table 8-83. A Yes in the Extensible column of a subelement listed in Table 8-83 indicates that the Length of the subelement might be extended in future revisions or amendments of this standard. When the Extensible column of an element is equal to Subelements, then the subelement might be extended in future revisions or amendments of this standard by defining additional subelements within the subelement. See 9.24.9.

Table 8-83—Optional subelement IDs for Channel Load Report

Subelement ID	Name	Length field (octets)	Extensible
0–220	Reserved		
221	Vendor Specific	1 to 237	
222–255	Reserved		

The Vendor Specific subelements have the same format as their corresponding elements (see 8.4.2.28). Multiple Vendor Specific subelements are optionally present in the list of optional subelements.

8.4.2.24.6 Noise Histogram Report

The format of the Measurement Report field of a Noise Histogram Report is shown in Figure 8-147.

	Operating Class	Channel Number	Actual Measurement Start Time	Measurement Duration	Antenna ID	ANPI
Octets:	1	1	8	2	1	1

	IPI 0 Density	IPI 1 Density	IPI 2 Density	. . .	IPI 8 Density	IPI 9 Density	IPI 10 Density	Optional subelements
Octets:	1	1	1	5	1	1	1	variable

Figure 8-147—Measurement Report field format for Noise Histogram Report

Operating Class indicates the channel set for which the measurement request applies. Country, Operating Class, and Channel Number together specify the channel frequency and spacing for which the measurement request applies. Valid values of Operating Class are shown in Annex E.

Channel Number indicates the channel number for which the measurement report applies. Channel Number is defined within an Operating Class as shown in Annex E.

Actual Measurement Start Time is set to the value of the measuring STA’s TSF timer at the time the measurement started.

Measurement Duration is set to the duration over which the Noise Histogram Report was measured, expressed in units of TUs.

Antenna ID is set to the identifying number for the antenna(s) used for this measurement. Antenna ID is defined in 8.4.2.42.

ANPI is set to the average noise plus interference power value measured during the indicated Measurement Duration while the indicated channel is idle as described in 10.11.9.4.

The Noise Histogram Report contains the IPI densities, as defined in 10.11.9.4, observed in the channel for the eleven IPI levels defined in Table 8-84.

Table 8-84—IPI Definitions for a Noise Histogram Report

IPI Level	IPI Measured Power (dBm)
0	IPI ≤ – 92
1	–92 < IPI ≤ –89
2	–89 < IPI ≤ –86
3	–86 < IPI ≤ –83
4	–83 < IPI ≤ –80
5	–80 < IPI ≤ –75
6	–75 < IPI ≤ –70
7	–70 < IPI ≤ –65
8	–65 < IPI ≤ –60
9	–60 < IPI ≤ –55
10	–55 < IPI

The Optional Subelements field format contains zero or more subelements, each consisting of a 1-octet Subelement ID field, a 1-octet Length field, and a variable-length Data field, as shown in Figure 8-402. Any optional subelements are ordered by nondecreasing subelement ID.

The Subelement ID field values for the defined optional subelements are shown in Table 8-85. A Yes in the Extensible column of a subelement listed in Table 8-85 indicates that the Length of the subelement might be extended in future revisions or amendments of this standard. When the Extensible column of an element is equal to Subelements, then the subelement might be extended in future revisions or amendments of this standard by defining additional subelements within the subelement. See 9.24.9.

Table 8-85—Optional subelement IDs for Noise Histogram Report

Subelement ID	Name	Length field (octets)	Extensible
0–220	Reserved		
221	Vendor Specific	1 to 225	
222–255	Reserved		

The Vendor Specific subelements have the same format as their corresponding elements (see 8.4.2.28). Multiple Vendor Specific subelements are optionally present in the list of optional subelements.

8.4.2.24.7 Beacon Report

The format of the Measurement Report field corresponding to a Beacon Report is shown in Figure 8-148.

	Operating Class	Channel Number	Actual Measurement Start Time	Measurement Duration	Reported Frame Information
Octets:	1	1	8	2	1

	RCPI	RSNI	BSSID	Antenna ID	Parent TSF	Optional Subelements
Octets:	1	1	6	1	4	variable

Figure 8-148—Measurement Report field format for Beacon Report

Operating Class indicates the channel set for which the measurement request applies. Country, Operating Class, and Channel Number together specify the channel frequency and spacing for which the measurement request applies. Valid values of Operating Class are shown in Annex E.

Channel Number indicates the channel number for which the measurement report applies. Channel Number is defined within an Operating Class as shown in Annex E.

Actual Measurement Start Time is set to the value of the measuring STA's TSF timer at the time the measurement started.

Measurement Duration is set to the duration over which the Beacon Report was measured, expressed in units of TUs.

The Reported Frame Information field contains two subfields as shown in Figure 8-149.

	B0 B6	B7
	Condensed PHY Type	Reported Frame Type
Bits:	7	1

Figure 8-149—Reported Frame Information field

Condensed PHY Type indicates the physical medium type on which the Beacon, Measurement Pilot, or Probe Response frame being reported was received. It has an integer value between 0 and 127 coded according to the value of dot11PHYType.

Reported Frame Type indicates the type of frame reported. A value of 0 indicates a Beacon or Probe Response frame; a value of 1 indicates a Measurement Pilot frame.

RCPI indicates the received channel power of the Beacon, Measurement Pilot, or Probe Response frame, which is a logarithmic function of the received signal power, as defined in the RCPI measurement subclause for the indicated PHY Type.

RSNI indicates the received signal to noise indication for the Beacon, Measurement Pilot, or Probe Response frame, as described in 8.4.2.43.

The BSSID field contains the BSSID from the Beacon, Measurement Pilot, or Probe Response frame being reported.

The Antenna ID field contains the identifying number for the antenna(s) used for this measurement. Antenna ID is defined in 8.4.2.42.

The Parent TSF field contains the lower 4 octets of the measuring STA's TSF timer value at the start of reception of the first octet of the timestamp field of the reported Beacon, Measurement Pilot, or Probe Response frame at the time the Beacon, Measurement Pilot, or Probe Response frame being reported was received.

The Optional Subelements field format contains zero or more subelements, each consisting of a 1-octet Subelement ID field, a 1-octet Length field, and a variable-length Data field, as shown in Figure 8-402. Any optional subelements are ordered by nondecreasing subelement ID.

The Subelement ID field values for the defined optional subelements are shown in Table 8-86. A Yes in the Extensible column of a subelement listed in Table 8-86 indicates that the Length of the subelement might be extended in future revisions or amendments of this standard. When the Extensible column of an element is equal to Subelements, then the subelement might be extended in future revisions or amendments of this standard by defining additional subelements within the subelement. See 9.24.9.

Table 8-86—Optional subelement IDs for Beacon Report

Subelement ID	Name	Length field (octets)	Extensible
0	Reserved		
1	Reported Frame Body	0 to 224	
2–220	Reserved		
221	Vendor Specific	1 to 224	
222–255	Reserved		

The Reported Frame Body subelement contains the requested fields and elements of the frame body of the reported Beacon, Measurement Pilot, or Probe Response frame. If the Reporting Detail subelement of the corresponding Beacon Request equals 0, the Reported Frame Body subelement is not included in the Beacon Report. If the Reporting Detail subelement equals 1, all fixed fields and any elements whose Element IDs are present in the Request element in the corresponding Beacon Request are included in the Reported Frame Body subelement, in the order that they appeared in the reported frame. If the Reporting Detail field equals 2, all fixed fields and elements are included in the order they appeared in the reported frame. Reported TIM elements are truncated such that only the first 4 octets of the element are reported and the element Length field is modified to indicate the truncated length of 4. Reported IBSS dynamic frequency selection (DFS) elements shall be truncated so that only the lowest and highest channel number map are reported and the element Length field is modified to indicate the truncated length of 13. Reported RSNEs shall be truncated so that only the first 4 octets of the element are reported and the element Length field is modified to indicate the truncated length of 4. If the length of the Reported Frame Body subelement would cause the Measurement Report element to exceed the maximum element size, then the Reported Frame Body subelement is truncated so that the last element in the Reported Frame Body subelement is a complete element.

The Vendor Specific subelements have the same format as their corresponding elements (see 8.4.2.28). Multiple Vendor Specific subelements are optionally present in the list of optional subelements.

8.4.2.24.8 Frame Report

The format of the Measurement Report field corresponding to a Frame Report is shown in Figure 8-150.

	Operating Class	Channel Number	Actual Measurement Start Time	Measurement Duration	Optional Subelements
Octets:	1	1	8	2	variable

Figure 8-150—Measurement Report field format for Frame Report

Operating Class indicates the channel set for which the measurement request applies. Country, Operating Class, and Channel Number together specify the channel frequency and spacing for which the measurement request applies. Valid values of Operating Class are shown in Annex E.

Channel Number indicates the channel number for which the measurement report applies. Channel Number is defined within an Operating Class as shown in Annex E.

Actual Measurement Start Time is set to the value of the measuring STA's TSF timer at the time the measurement started.

Measurement Duration is set to the duration over which the Frame Report was measured, expressed in units of TUs.

The Optional Subelements field format contains zero or more subelements, each consisting of a 1-octet Subelement ID field, a 1-octet Length field, and a variable-length Data field, as shown in Figure 8-402. The optional subelements are ordered by nondecreasing subelement ID.

The Subelement ID field values for the defined optional subelements are shown in Table 8-87. A Yes in the Extensible column of a subelement listed in Table 8-87 indicates that the Length of the subelement might be extended in future revisions or amendments of this standard. When the Extensible column of an element is equal to Subelements, then the subelement might be extended in future revisions or amendments of this standard by defining additional subelements within the subelement. See 9.24.9.

Table 8-87—Optional subelement IDs for Frame Report

Subelement ID	Name	Length field (octets)	Extensible
0	Reserved		
1	Frame Count Report	0 to 228	
2–220	Reserved		
221	Vendor Specific	1 to 238	
222–255	Reserved		

The Frame Count Report subelement is used to report information about frames sent by a transmitter. The Frame Count Report subelement is as shown in Figure 8-151.

			Zero or more entries
	Subelement ID	Length	Frame Report Entry
Octets:	1	1	n x 19

Figure 8-151—Frame Count Report subelement format

The value of the subelement ID is equal to the Frame Count Report value in Table 8-87.

The Length field value is equal to 19 times the number of Frame Count Report Entries included.

The format of the Frame Report Entry is shown in Figure 8-152.

	Transmit Address	BSSID	PHY Type	Average RCPI	Last RSNI	Last RCPI	Antenna ID	Frame Count
Octets:	6	6	1	1	1	1	1	2

Figure 8-152—Frame Report Entry field format

The Transmit Address field contains the Transmitter Address (TA) from the frames being reported.

The BSSID field contains the BSSID from the frames being reported.

PHY Type indicates the physical medium type for the frame(s) being reported. Valid entries are coded according to the value of dot11PHYType.

Average RCPI indicates the average value for the received channel power of frames received and counted in this Frame Report Entry, as described in 10.11.9.2. Average RCPI is a logarithmic function of the received signal power, as defined in the RCPI measurement subclause for the PHY Type.

Last RSNI indicates the received signal to noise indication of the most recently measured frame counted in this Frame Report Entry, as described in 8.4.2.43.

Last RCPI indicates the received channel power of the most recently measured frame counted in this Frame Report entry. Last RCPI is a logarithmic function of the received signal power, as defined in the RCPI measurement subclause for the PHY Type.

The Antenna ID field contains the identifying number for the antenna(s) used to receive the most recently measured frame counted in this Frame Report entry. Antenna ID is defined in 8.4.2.42.

Frame Count is a count of the data and management frames received with the indicated Transmit Address and BSSID during the measurement duration. The value 65 535 indicates a count of 65 535 or more.

The Vendor Specific subelement has the same format as the Vendor Specific element (see 8.4.2.28). Multiple Vendor Specific subelements are optionally present in the list of optional subelements.

8.4.2.24.9 STA Statistics Report

The format of the Measurement Report field of a STA Statistics Report is shown in Figure 8-153.

	Measurement Duration	Group Identity	Statistics Group Data	Optional Subelements
Octets:	2	1	variable	variable

Figure 8-153—Measurement Report field format for STA Statistics Report

The STA Statistics Report reports the change in the requested Statistics Group Data values measured within the Measurement Duration. When the Measurement Duration is equal to 0 the current values of the requested Statistics Group Data is reported, rather than the change.

The Measurement Duration is set to the duration over which the change in Statistics Group Data was measured and reported, expressed in units of TUs. A Measurement Duration value of 0 indicates a report of the current values of the Statistics Group Data.

Group Identity indicates the requested statistics group describing the Statistics Group Data according to Table 8-88.

Statistics Group Data contains the requested statistics from the MIB in Annex C related to the interface on which the request was received according to Table 8-88. Units used for reporting a statistic or change in statistic are the units used to define the statistic in Annex C. When Measurement Duration value is nonzero, the reported data values for statistics that are not counters are the current values of the statistics data at the end of the Measurement Duration.

Table 8-88—Group Identity for a STA Statistics Report

Group Identity Requested	Statistics Group Data field length (octets)	Statistics Returned
0	28	dot11Counters Group for the Interface on which the STA Statistics Request was received (with the exception of WEPUndecryptableCount and those counters listed in Group Identity 1): dot11TransmittedFragmentCount (Counter32), dot11GroupTransmittedFrameCount (Counter32), dot11FailedCount (Counter32), dot11ReceivedFragmentCount (Counter32), dot11GroupReceivedFrameCount (Counter32), dot11FCSErrorCount (Counter32), dot11TransmittedFrameCount (Counter32)
1	24	dot11MACStatistics Group for the Interface on which the STA Statistics Request was received: dot11RetryCount (Counter32), dot11MultipleRetryCount (Counter32), dot11FrameDuplicateCount (Counter32), dot11RTSSuccessCount (Counter32), dot11RTSFailureCount (Counter32), dot11ACKFailureCount (Counter32)

Table 8-88—Group Identity for a STA Statistics Report *(continued)*

Group Identity Requested	Statistics Group Data field length (octets)	Statistics Returned
2	52	dot11QosCounters Group for UP0 for the Interface on which the STA Statistics Request was received: dot11QosTransmittedFragmentCount (Counter32), dot11QosFailedCount (Counter32), dot11QosRetryCount (Counter32), dot11QosMultipleRetryCount (Counter32), dot11QosFrameDuplicateCount (Counter32), dot11QosRTSSuccessCount (Counter32), dot11QosRTSFailureCount (Counter32), dot11QosACKFailureCount (Counter32), dot11QosReceivedFragmentCount (Counter32), dot11QosTransmittedFrameCount (Counter32), dot11QosDiscardedFrameCount (Counter32), dot11QosMPDUsReceivedCount (Counter32), dot11QosRetriesReceivedCount (Counter32)
3	52	dot11QosCounters Group for UP1 for the Interface on which the STA Statistics Request was received: dot11QosTransmittedFragmentCount (Counter32), dot11QosFailedCount (Counter32), dot11QosRetryCount (Counter32), dot11QosMultipleRetryCount (Counter32), dot11QosFrameDuplicateCount (Counter32), dot11QosRTSSuccessCount (Counter32), dot11QosRTSFailureCount (Counter32), dot11QosACKFailureCount (Counter32), dot11QosReceivedFragmentCount (Counter32), dot11QosTransmittedFrameCount (Counter32), dot11QosDiscardedFrameCount (Counter32), dot11QosMPDUsReceivedCount (Counter32), dot11QosRetriesReceivedCount (Counter32)
4	52	dot11QosCounters Group for UP2 for the Interface on which the STA Statistics Request was received: dot11QosTransmittedFragmentCount (Counter32), dot11QosFailedCount (Counter32), dot11QosRetryCount (Counter32), dot11QosMultipleRetryCount (Counter32), dot11QosFrameDuplicateCount (Counter32), dot11QosRTSSuccessCount (Counter32), dot11QosRTSFailureCount (Counter32), dot11QosACKFailureCount (Counter32), dot11QosReceivedFragmentCount (Counter32), dot11QosTransmittedFrameCount (Counter32), dot11QosDiscardedFrameCount (Counter32), dot11QosMPDUsReceivedCount (Counter32), dot11QosRetriesReceivedCount (Counter32)

Table 8-88—Group Identity for a STA Statistics Report *(continued)*

Group Identity Requested	Statistics Group Data field length (octets)	Statistics Returned
5	52	dot11QosCounters Group for UP3 for the Interface on which the STA Statistics Request was received: dot11QosTransmittedFragmentCount (Counter32), dot11QosFailedCount (Counter32), dot11QosRetryCount (Counter32), dot11QosMultipleRetryCount (Counter32), dot11QosFrameDuplicateCount (Counter32), dot11QosRTSSuccessCount (Counter32), dot11QosRTSFailureCount (Counter32), dot11QosACKFailureCount (Counter32), dot11QosReceivedFragmentCount (Counter32), dot11QosTransmittedFrameCount (Counter32), dot11QosDiscardedFrameCount (Counter32), dot11QosMPDUsReceivedCount (Counter32), dot11QosRetriesReceivedCount (Counter32)
6	52	dot11QosCounters Group for UP4 for the Interface on which the STA Statistics Request was received: dot11QosTransmittedFragmentCount (Counter32), dot11QosFailedCount (Counter32), dot11QosRetryCount (Counter32), dot11QosMultipleRetryCount (Counter32), dot11QosFrameDuplicateCount (Counter32), dot11QosRTSSuccessCount (Counter32), dot11QosRTSFailureCount (Counter32), dot11QosACKFailureCount (Counter32), dot11QosReceivedFragmentCount (Counter32), dot11QosTransmittedFrameCount (Counter32), dot11QosDiscardedFrameCount (Counter32), dot11QosMPDUsReceivedCount (Counter32), dot11QosRetriesReceivedCount (Counter32)
7	52	dot11QosCounters Group for UP5 for the Interface on which the STA Statistics Request was received: dot11QosTransmittedFragmentCount (Counter32), dot11QosFailedCount (Counter32), dot11QosRetryCount (Counter32), dot11QosMultipleRetryCount (Counter32), dot11QosFrameDuplicateCount (Counter32), dot11QosRTSSuccessCount (Counter32), dot11QosRTSFailureCount (Counter32), dot11QosACKFailureCount (Counter32), dot11QosReceivedFragmentCount (Counter32), dot11QosTransmittedFrameCount (Counter32), dot11QosDiscardedFrameCount (Counter32), dot11QosMPDUsReceivedCount (Counter32), dot11QosRetriesReceivedCount (Counter32)

Table 8-88—Group Identity for a STA Statistics Report *(continued)*

Group Identity Requested	Statistics Group Data field length (octets)	Statistics Returned
8	52	dot11QosCounters Group for UP6 for the Interface on which the STA Statistics Request was received: dot11QosTransmittedFragmentCount (Counter32), dot11QosFailedCount (Counter32), dot11QosRetryCount (Counter32), dot11QosMultipleRetryCount (Counter32), dot11QosFrameDuplicateCount (Counter32), dot11QosRTSSuccessCount (Counter32), dot11QosRTSFailureCount (Counter32), dot11QosACKFailureCount (Counter32), dot11QosReceivedFragmentCount (Counter32), dot11QosTransmittedFrameCount (Counter32), dot11QosDiscardedFrameCount (Counter32), dot11QosMPDUsReceivedCount (Counter32), dot11QosRetriesReceivedCount (Counter32)
9	52	dot11QosCounters Group for UP7 for the Interface on which the STA Statistics Request was received: dot11QosTransmittedFragmentCount (Counter32), dot11QosFailedCount (Counter32), dot11QosRetryCount (Counter32), dot11QosMultipleRetryCount (Counter32), dot11QosFrameDuplicateCount (Counter32), dot11QosRTSSuccessCount (Counter32), dot11QosRTSFailureCount (Counter32), dot11QosACKFailureCount (Counter32), dot11QosReceivedFragmentCount (Counter32), dot11QosTransmittedFrameCount (Counter32), dot11QosDiscardedFrameCount (Counter32), dot11QosMPDUsReceivedCount (Counter32), dot11QosRetriesReceivedCount (Counter32)
10	8	dot11BSSAverageAccessDelay Group (only available at an AP): dot11STAStatisticsAPAverageAccessDelay (INTEGER), dot11STAStatisticsAverageAccessDelayBestEffort (INTEGER), dot11STAStatisticsAverageAccessDelayBackGround (INTEGER), dot11STAStatisticsAverageAccessDelayVideo (INTEGER), dot11STAStatisticsAverageAccessDelayVoice (INTEGER), dot11STAStatisticsStationCount (INTEGER), dot11STAStatisticsChannelUtilization (INTEGER)
11	40	STA Counters from dot11CountersGroup3 (A-MSDU): dot11TransmittedAMSDUCount (Counter32), dot11FailedAMSDUCount (Counter32), dot11RetryAMSDUCount (Counter32), dot11MultipleRetryAMSDUCount (Counter32), dot11TransmittedOctetsInAMSDUCount (Counter64), dot11AMSDUAckFailureCount (Counter32), dot11ReceivedAMSDUCount (Counter32), dot11ReceivedOctetsInAMSDUCount (Counter64)

Table 8-88—Group Identity for a STA Statistics Report *(continued)*

Group Identity Requested	Statistics Group Data field length (octets)	Statistics Returned
12	36	STA Counters from dot11CountersGroup3 (A-MPDU): dot11TransmittedAMPDUCount (Counter32), dot11TransmittedMPDUsInAMPDUCount (Counter32), dot11TransmittedOctetsInAMPDUCount (Counter64), dot11AMPDUReceivedCount (Counter32), dot11MPDUInReceivedAMPDUCount (Counter32), dot11ReceivedOctetsInAMPDUCount (Counter64), dot11AMPDUDelimiterCRCErrorCount (Counter32)
13	36	STA Counters from dot11CountersGroup3 (BlockAckReq, Channel Width, PSMP): dot11ImplicitBARFailureCount (Counter32), dot11ExplicitBARFailureCount (Counter32), dot11ChannelWidthSwitchCount (Counter32), dot11TwentyMHzFrameTransmittedCount (Counter32), dot11FortyMHzFrameTransmittedCount (Counter32), dot11TwentyMHzFrameReceivedCount (Counter32), dot11FortyMHzFrameReceivedCount (Counter32), dot11PSMPUTTGrantDuration (Counter32), dot11PSMPUTTUsedDuration (Counter32)
14	36	STA Counters from dot11CountersGroup3 (RD, dual CTS, L-SIG TXOP protection): dot11GrantedRDGUsedCount (Counter32), dot11GrantedRDGUnusedCount (Counter32), dot11TransmittedFramesInGrantedRDGCount (Counter32), dot11TransmittedOctetsInGrantedRDGCount (Counter64), dot11DualCTSSuccessCount (Counter32), dot11DualCTSFailureCount (Counter32), dot11RTSLSIGSuccessCount (Counter32), dot11RTSLSIGFailureCount (Counter32)
15	20	STA Counters from dot11CountersGroup3 (beamforming and STBC): dot11BeamformingFrameCount (Counter32), dot11STBCCTSSuccessCount (Counter32), dot11STBCCTSFailureCount (Counter32), dot11nonSTBCCTSSuccessCount (Counter32), dot11nonSTBCCTSFailureCount (Counter32)
16	28	dot11RSNAStats Group for the Interface on which the STA Statistics Request was received: dot11RSNAStatsCMACICVErrors (Counter32) dot11RSNAStatsCMACReplays (Counter32) dot11RSNAStatsRobustMgmtCCMPReplays(Counter32) dot11RSNAStatsTKIPICVErrors (Counter32) dot11RSNAStatsTKIPReplays (Counter32) dot11RSNAStatsCCMPDecryptErrors (Counter32) dot11RSNAStatsCCMPReplays (Counter32)
17–255		Reserved

The format of the Measurement Report field for dot11Counters Group is shown in Figure 8-154.

	dot11TransmittedFragmentCount	dot11Group Transmitted FrameCount	dot11Failed Count	dot11Received FragmentCount	dot11Group ReceivedFrame Count
Octets:	4	4	4	4	4

	dot11FCSError Count	dot11transmitted FrameCount
Octets:	4	4

Figure 8-154—Measurement Report field format for dot11Counters Group

The format of the Measurement Report field for dot11MACStatistics is shown in Figure 8-155.

	dot11Retry Count	dot11Multiple RetryCount	dot11Frame DuplicateCount	dot11RTS Success Count	dot11RTS FailureCount
Octets:	4	4	4	4	4

	dot11ACK FailureCount
Octets:	4

Figure 8-155—Measurement Report field format for dot11MACStatistics Group

The format of the Measurement Report field for dot11QosCounters Group for UPx is shown in Figure 8-156, where x is 0 – 7 and where the listed variables are obtained from the column of the QoS Counters Table indexed by x + 1. For example, the variables for dot11QosCounters Group for UP2 are from the third column of the dot11QosCountersTable, obtained from the dot11QosCountersEntry in which dot11QosCountersIndex is equal to 3.

	dot11Qos Transmitted Fragment Count	dot11Qos FailedCount	dot11Qos RetryCount	dot11Qos Multiple RetryCount	dot11Qos Frame DuplicateCount
Octets:	4	4	4	4	4

	dot11Qos RTS SuccessCount	dot11Qos RTS FailureCount	dot11Qos ACK FailureCount	dot11Qos Received FragmentCount	dot11Qos Transmitted Frame Count
Octets:	4	4	4	4	4

	dot11Qos Discarded FrameCount	dot11Qos MPDUs ReceivedCount	dot11Qos Retries ReceivedCount
Octets:	4	4	4

Figure 8-156—Measurement Report field format for dot11QosCounters Group for UPx

The format of the Measurement Report field for dot11BSSAverageAccessDelay Group is shown in Figure 8-157. Non-QoS APs set dot11STAStatisticsAverageAccessDelayBestEffort, dot11STAStatistics-AverageAccessDelayBackGround, dot11STAStatisticsAverageAccessDelayVideo, and dot11STA-StatisticsAverageAccessDelayVoice to 255 (not available).

	dot11STA StatisticsAP AverageAccess Delay	dot11STA Statistics AverageAccess DelayBestEffort	dot11STA Statistics AverageAccess Delay BackGround	dot11STA Statistics AverageAccess DelayVideo	dot11STA Statistics AverageAccess DelayVoice
Octets:	1	1	1	1	1

	dot11STA Statistics StationCount	dot11STA Statistics Channel Utilization
Octets:	2	1

Figure 8-157—Measurement Report field format for dot11BSSAverageAccessDelay Group

The Optional Subelements field format contains zero or more subelements, each consisting of a 1-octet Subelement ID field, a 1-octet Length field, and a variable-length Data field, as shown in Figure 8-402. Any optional subelements are ordered by nondecreasing subelement ID.

The Subelement ID field values for the defined optional subelements are shown in Table 8-89. A Yes in the Extensible column of a subelement listed in Table 8-89 indicates that the Length of the subelement might be extended in future revisions or amendments of this standard. When the Extensible column of an element is equal to Subelements, then the subelement might be extended in future revisions or amendments of this standard by defining additional subelements within the subelement. See 9.24.9.

Table 8-89—Optional subelement IDs for STA Statistics Report

Subelement ID	Name	Length field (octets)	Extensible
0	Reserved		
1	Reporting Reason	Variable	
2–220	Reserved		
221	Vendor Specific	1 to 240	
222–255	Reserved		

The format of the Measurement Report field for RSNA Counters Group is shown in Figure 8-158.

	dot11RSNAStats CMACICVErrors	dot11RSNAStatsCM ACReplays	dot11RSNAStats RobustMgmt CCMPReplays	dot11RSNAStats TKIPICVErrors
Octets:	4	4	4	4

	dot11RSNAStats TKIPReplays	dot11RSNAStats CCMPDecryptErrors	dot11RSNAStats CCMPReplays
Octets	4	4	4

Figure 8-158—Measurement Report field format for RSNA Counters Group

The Reporting Reason subelement indicates the reason why the measuring STA sent the STA Statistics report. It is present if Statistics Group Name is from STA Counters, QoS STA Counters, or RSNA Counters (see 10.11.9.5).

The Reporting Reason subelement for STA Statistics Group Identities 0 or 1 (STA Counters) is shown in Figure 8-159.

	B0	B1	B2	B3
	dot11Failed	dot11FCS Error	dot11Multiple Retry	dot11Frame Duplicate
Bits	1	1	1	1

	B4	B5	B6	B7
	dot11RTS Failure	dot11ACK Failure	dot11Retry	Reserved
Bits	1	1	1	1

Figure 8-159—Reporting Reason subelement for STA Counters

The Reporting Reason subelement for STA Statistics Group Identity 2 to 9 (QoS STA Counters) is shown in Figure 8-160.

	B0	B1	B2	B3
	dot11QoS Failed	dot11QoS Retry	dot11QoSMultiple Retry	dot11QoSFrame Duplicate
Bits	1	1	1	1

	B4	B5	B6	B7
	dot11QoSRTS Failure	dot11QoSACK Failure	dot11QoS Discarded	Reserved
Bits	1	1	1	1

Figure 8-160—Reporting Reason subelement for QoS STA Counters

The Reporting Reason subelement for STA Statistics Group Identity 16 (RSNA Counters) is shown in Figure 8-161.

	B0	B1	B2	B3
	dot11RSNAStats CMACICVErrors	dot11RSNA StatsCMACReplays	dot11RSNAStats RobustMgmt CCMPReplays	dot11RSNAStats TKIPICVErrors
Bits:	1	1	1	1

	B4	B5	B6	B7
	dot11RSNAStats TKIPReplays	dot11RSNA StatsTKIPReplays	dot11RSNAStats CCMPReplays	Reserved
Bits:	1	1	1	1

Figure 8-161—Reporting Reason subelement for RSNA Counters

In a nontriggered STA Statistics Report, all fields in the Reporting Reason subelement are set to 0.

The Vendor Specific subelements have the same format as their corresponding elements (see 8.4.2.28). Multiple Vendor Specific subelements are optionally present in the list of optional subelements.

8.4.2.24.10 Location Configuration Information Report

An LCI Report includes Latitude, Longitude, Altitude, and optional Azimuth information. The LCI report format is shown in Figure 8-162.

This structure and information fields are little-endian, per conventions defined in 8.2.2, and are based on the LCI format described in IETF RFC 3825.

The definition of elements within the LCI report are as defined in Section 2.1 of IETF RFC 3825 (July 2004) or as defined herein.

	B0 B7	B8 B15
	Element ID	Length
Bits	8	8

	B16 B21	B22 B46
	Latitude Resolution	Latitude Fraction
Bits	6	25

	B47 B55	B56 B61
	Latitude Integer	Longitude Resolution
Bits	9	6

	B62 B86	B87 B95
	Longitude Fraction	Longitude Integer
Bits	25	9

	B96 B99	B100 B105	B106 B113
	Altitude Type	Altitude Resolution	Altitude Fraction
Bits	4	6	8

	B114 B135	B136 B143
	Altitude Integer	Datum
Bits	22	8

	B144
	Optional Subelements
Bits	variable

Figure 8-162—Measurement Report field format for Location Configuration Information Report

NOTE—An example of fixed/fractional notation, using the longitude of the Sears Tower from p. 13 of IETF RFC 3825 (July 2004):

> Longitude 87.63602 degrees West (or –87.63602 degrees),
> Using twos complement, 34 bit fixed point, 25 bit fraction,
> Longitude = 0xf50ba5b97,
> Longitude = 1101010000101110100101101110010111 (big-endian)
> DSE registered location expression for a Longitude resolution of 34-bits:
> Bits 56–61 Longitude resolution = (bit 56) 0 1 0 0 0 1 (bit 61)
> Bits 62–86 Longitude fraction = (bit 62) 1 1 1 0 1 0 0 1 1 1 0 1 1 0 1 0 0 1 0 1 1 1 0 1 0 (bit 86)
> Bits 87–95 Longitude integer = (bit 87) 0 0 0 1 0 1 0 1 1 (bit 95)
>
> The octets in transmission order = E2 E5 96 2E D4.

The Optional Subelements field format contains zero or more subelements, each consisting of a 1-octet Subelement ID field, a 1-octet Length field, and a variable-length Data field, as shown in Figure 8-402. Any optional subelements are ordered by nondecreasing subelement ID.

The Subelement ID field values for the defined optional subelements are shown in Table 8-90. A Yes in the Extensible column of a subelement listed in Table 8-90 indicates that the Length of the subelement might be extended in future revisions or amendments of this standard. When the Extensible column of an element is equal to Subelements, then the subelement might be extended in future revisions or amendments of this standard by defining additional subelements within the subelement. See 9.24.9.

Table 8-90—Optional subelement IDs for Location Configuration Information Report

Subelement ID	Name	Length field (octets)	Extensible
0	Reserved		
1	Azimuth Report	2	Yes
2	Originator Requesting STA MAC Address	6	No
3	Target MAC Address	6	No
4–220	Reserved		
221	Vendor Specific	1 to 232	
222–255	Reserved		

The Azimuth Report subelement is used to report an azimuth. The Azimuth Report subelement is as shown in Figure 8-163.

	Subelement ID	Length	Azimuth Report
Octets:	1	1	2

Figure 8-163—Azimuth Report subelement format

The value of the subelement ID is equal to the Azimuth Report value in Table 8-90.

The value of the Length field in octets is equal to 1.

The Azimuth Report field of an Azimuth Report subelement contains three subfields as shown in Figure 8-164.

	B0 B1	B2	B3 B6	B7 B15
	Reserved	Azimuth Type	Azimuth Resolution	Azimuth
Bits:	2	1	4	9

Figure 8-164—Azimuth Report subfield

Azimuth Type is set to 1 to report the Azimuth of the bearing of the requestor with respect to the responder, and is set to 0 to report the Azimuth of front surface of the reporting STA.

Azimuth Resolution is 4 bits, indicating the number of valid most significant bits in the Azimuth.

Azimuth is a 9-bit unsigned integer value in degrees from true north, of the type defined by the Azimuth Type field.

The Originator Requesting STA MAC Address subelement contains the MAC address of the STA that requested the Location Information and it is present whenever the location subject definition field in the corresponding LCI Request was set to 2. The format of the Originator Requesting STA MAC Address subelement is shown in Figure 8-126.

The Target MAC Address subelement contains the MAC address of the STA whose Location Information was requested and it is present whenever the location subject definition field in the corresponding LCI Request was set to 2. The format of the Target MAC Address subelement is shown in Figure 8-127.

The Vendor Specific subelement has the same format as the Vendor Specific element (see 8.4.2.28). Multiple Vendor Specific subelements are optionally present in the list of optional subelements.

8.4.2.24.11 Transmit Stream/Category Measurement Report

The Transmit Stream/Category Measurement applies to TIDs for Traffic Streams associated with TSPECs and also to TIDs for Traffic Categories for QoS traffic without TSPECs. The format of the Measurement Report field corresponding to a Transmit Stream/Category Measurement Report is shown in Figure 8-165.

	Actual Measurement Start Time	Measurement Duration	Peer STA Address	Traffic Identifier	Reporting Reason	Transmitted MSDU Count
Octets:	8	2	6	1	1	4

	MSDU Discarded Count	MSDU Failed Count	MSDU Multiple Retry Count	QoS CF-Polls Lost Count	Average Queue Delay	Average Transmit Delay
Octets:	4	4	4	4	4	4

	Bin 0 Range	Bin 0	Bin 1	Bin 2	Bin 3	Bin 4	Bin 5	Optional Subelements
Octets:	1	4	4	4	4	4	4	variable

Figure 8-165—Measurement Report field format for Transmit Stream/ Category Measurement Report

Actual Measurement Start Time is set to the TSF at the time at which the measurement started, or for a triggered Transmit Stream/Category Measurement Report, the TSF value at the reporting QoS STA when the trigger condition was met.

Measurement Duration is set to the duration over which the Transmit Stream/Category Measurement Report was measured, expressed in units of TUs. For a triggered Transmit Stream/Category Measurement Report, metrics are reported over a number of transmitted MSDUs rather than a duration; hence Measurement Duration is set to 0; see 10.11.9.8.

The Peer STA Address contains a MAC address indicating the RA for the measured frames.

The Traffic Identifier field contains the TID subfield as shown in Figure 8-128. The TID subfield indicates the TC or TS for which traffic was measured.

The Reporting Reason field is a bit field indicating the reason that the measuring QoS STA sent the transmit stream/category measurement report. The Reporting Reason field is shown in Figure 8-166.

	B0	B1	B2	B3 B7
	Average Trigger	Consecutive Trigger	Delay Trigger	Reserved
Bits:	1	1	1	5

Figure 8-166—Reporting Reason field

- The Average Trigger bit set to 1 indicates that the Transmit Stream/Category Measurement Report was generated as a triggered report due to the Average Error trigger.
- The Consecutive Trigger bit set to 1 indicates that the Transmit Stream/Category Measurement Report was generated as a triggered report due to the Consecutive Error trigger.
- The Delay Trigger bit set to 1 indicates that the Transmit Stream/Category Measurement Report was generated as a triggered report due to the delay exceeding the Delay Threshold.

When a Transmit Stream/Category Measurement Report is sent as a direct response to a Transmit Stream/Category Measurement Request and not as a triggered Transmit Stream/Category Measurement Report, all bit fields in the Reporting Reason field are set to 0. This is termed a requested Transmit Stream/Category Measurement Report. Within a triggered Transmit Stream/Category Measurement Report, more than one bit field in the Reporting Reason field might be set to 1 if more than one trigger condition was met.

The Transmitted MSDU Count, MSDU Failed Count, MSDU Discarded Count, MSDU Multiple Retry Count, QoS CF-Polls Lost Count, Average Queue Delay, Average Transmit Delay, and delay histogram fields relate to transmissions to the QoS STA given in the Peer STA Address field. Metrics are reported over the Measurement Duration, or for triggered transmit stream/category measurements, over the Measurement Count. Any counter that increments to a value of 2^{32}–1 terminates the measurement.

The Transmitted MSDU Count field contains the number of MSDUs for the TC or the TS specified by the TID that were successfully transmitted.

The MSDU Discarded Count field contains the number of MSDUs for the TC or the TS specified by the TID that were discarded due either to the number of transmit attempts exceeding dot11ShortRetryLimit or dot11LongRetryLimit (as appropriate), or due to the MSDU lifetime having been reached.

The MSDU Failed Count field contains the number of MSDUs for the TC or the TS specified by the TID that were discarded due to the number of transmit attempts exceeding dot11ShortRetryLimit or dot11LongRetryLimit (as appropriate).

The MSDU Multiple Retry Count field contains the number of MSDUs for the TC or the TS specified by the TID that were successfully transmitted after more than one retransmission attempt.

The QoS CF-Polls Lost Count field contains the number of QoS (+)CF-Poll frames that were transmitted where there was no response from the QoS STA. QoS CF-Polls Lost Count are returned only if the reporting QoS STA is contained within an AP and the TID is for a TS. This field is set to 0 when QoS CF-Polls Lost Count is not returned.

Average Queue Delay is the average queuing delay of the frames (MSDUs) that are passed to the MAC for the indicated Peer STA Address and the indicated Traffic Identifier. Queue Delay is expressed in TUs and is measured from the time the MSDU is passed to the MAC until the point at which the first or only fragment begins transmission.

Average Transmit Delay is the average delay of the frames (MSDUs) that are successfully transmitted for the indicated Peer STA Address and TID. Average Transmit Delay is measured from the time the MSDU is passed to the MAC until the point at which the entire MSDU has been successfully transmitted, including receipt of the final ACK from the peer STA if the QoSAck service class is being used. Average Transmit delay is expressed in units of TUs.

Bin 0 Range field value indicates the delay range of the first bin (Bin 0) of the Transmit Delay Histogram, expressed in units of TUs. It is also used to calculate the delay ranges of the other five bins making up the histogram. The delay range for each bin increases in a binary exponential fashion as follows:

Bin 0 range: $0 \leq \text{Delay} < B_0 = \text{Bin 0 Range field value}$

Bin i range: $2^{i-1} \times B_0 \leq \text{Delay} < 2^i \times B_0$, for $1 \leq i \leq 4$

Bin 5 range: $16 \times B_0 \leq \text{Delay}$

For example, if Bin 0 Range field value is 10 TUs, the bin delay ranges are as defined in Table 8-91.

Table 8-91—Delay definitions for a Transmit Stream/Category Measurement Report for a Bin 0 Range field value of 10 TU

Bin	Measured MSDU Transmit Delay (TUs)
0	Delay < 10
1	10 ≤ Delay < 20
2	20 ≤ Delay < 40
3	40 ≤ Delay < 80
4	80 ≤ Delay < 160
5	160 ≤ Delay

To compute the value reported in Bin i (i.e., B_i for i = 0, 1...5 of the Transmit Delay Histogram), the STA initializes all bin values to 0. For each MSDU successfully transmitted, the measured MSDU Transmit Delay determines which bin is to be incremented. If the measured delay has a duration time t within Bin i, then Bin i is increased by one. MSDU Transmit Delay is measured from the time the MSDU is passed to the MAC until the point at which the entire MSDU has been successfully transmitted, including receipt of the final ACK from the peer STA if the QoSAck service class is being used. The sum of the values in all six bins is equal to the value reported in the Transmitted MSDU Count.

The Optional Subelements field format contains zero or more subelements, each consisting of a 1-octet Subelement ID field, a 1-octet Length field, and a variable-length Data field, as shown in Figure 8-402. Any optional subelements are ordered by nondecreasing subelement ID.

The Subelement ID field values for the defined optional subelements are shown in Table 8-92. A Yes in the Extensible column of a subelement listed in Table 8-92 indicates that the Length of the subelement might be extended in future revisions or amendments of this standard. When the Extensible column of an element is equal to Subelements, then the subelement might be extended in future revisions or amendments of this standard by defining additional subelements within the subelement. See 9.24.9.

Table 8-92—Optional subelement IDs for Transmit Stream/Category Measurement Report

Subelement ID	Name	Length field (octets)	Extensible
0–220	Reserved		
221	Vendor Specific	1 to 179	
222–255	Reserved		

The Vendor Specific subelements have the same format as their corresponding elements (see 8.4.2.28). Multiple Vendor Specific subelements are optionally present in the list of optional subelements.

8.4.2.24.12 Multicast Diagnostics Report

The format of the Measurement Report field of a Multicast Diagnostics report is shown in Figure 8-167.

	Measurement Time	Measurement Duration	Group MAC Address	Multicast Reporting Reason	Multicast Received MSDU Count
Octets:	8	2	6	1	4

	First Sequence Number	Last Sequence Number	Multicast Rate	Optional Subelements
Octets:	2	2	2	variable

Figure 8-167—Measurement Report field format for a Multicast Diagnostics Report

The Measurement Time field is the value of the STA TSF timer at the time the measurement started. For a triggered Multicast Diagnostics report, this is the TSF value at the reporting STA when the trigger condition was met. When the reason for sending the report is Performance Measurement and the Multicast Received MSDU Count is nonzero, the Measurement Time field is the value of the STA TSF timer at the time of the first group addressed MSDU received during the measurement interval.

The Measurement Duration field specifies the period over which the Multicast Diagnostic Report was generated, expressed in units of TUs.

The Group MAC Address field contains the value from the Group MAC Address field from the Multicast Diagnostics Request to which the report relates.

The Multicast Reporting Reason field indicates the reason why the measuring STA sent the Multicast Diagnostics report. The Multicast Reporting Reason field is shown in Figure 8-168.

	B0	B1	B2 B7
	Inactivity Timeout Trigger	Measurement Result	Reserved
Bits:	1	1	6

Figure 8-168—Multicast Reporting Reason field

The subfields of the Multicast Reporting Reason field are defined as follows:

- The Inactivity Timeout Trigger field set to 1 indicates that Multicast Diagnostics Report was generated as a triggered report due to the timeout of the multicast diagnostic timer.
- The Measurement Result field set to 1 indicates that the Multicast Diagnostic Report contains the result of completing a multicast diagnostic request that did not contain a Multicast Triggered Reporting subelement.

All the bits in the Multicast Reporting Reason field are independent.

The Multicast Received MSDU Count field contains the total number of group addressed MSDUs with the indicated Multicast MAC Address that were received during the Measurement Duration. For a triggered multicast diagnostics measurement, the Multicast Received MSDU Count field contains the total number of group addressed MSDUs with the indicated Multicast MAC Address that were received between the acceptance of the multicast diagnostics measurement request and the occurrence of the trigger condition.

When the LSB of the first octet of the Multicast MAC address field in the multicast diagnostic request is 1, the twelve LSBs of the First Sequence Number field contain the sequence number of the first frame received with destination address equal to the value in the Multicast MAC address field during the measurement period. When the LSB of the first octet of the Multicast MAC address field in the multicast diagnostic request is 0, the twelve LSBs of the First Sequence Number field contain the sequence number of the first group addressed frame, that does not have the broadcast MAC address as its destination, received during the measurement period. The four most significant bits of the First Sequence Number field are set to 0.

When the LSB of the first octet of the Multicast MAC address field in the multicast diagnostic request is 1, the twelve LSBs of the Last Sequence Number field contain the sequence number of the last frame received with destination address equal to the value in the Multicast MAC address field during the measurement period. When the LSB of the first octet of the Multicast MAC address field in the multicast diagnostic request is 0, the twelve LSBs of the Last Sequence Number field contain the sequence number of the last group addressed frame, that does not have the broadcast MAC address as its destination, received during the measurement period. The four most significant bits of the Last Sequence Number field are set to 0.

The First Sequence Number field and the Last Sequence Number field are set to 0 if the Multicast Received MSDU Count is 0.

The Multicast Rate field specifies the highest data rate, in 0.5 Mb/s units, at which the STA has received a group addressed frame with a valid FCS during the measurement period.The Multicast Rate field is encoded with the MSB set to 1 to indicate that the data rate is in the basic rate set, and set to 0 to indicate that the data rate is not in the basic rate set. The remaining 15 bit value is multiplied by 0.5 Mb/s to indicate the data rate. The Multicast Rate field is 0 by the STA to indicate that it has not received a group addressed frame with a valid FCS during the measurement period.

The Optional Subelements field format contains zero or more subelements, each consisting of a 1-octet Subelement ID field, a 1-octet Length field and a variable-length Data field, as shown in Figure 8-402. The optional subelements are ordered by nondecreasing subelement ID.

The Subelement ID field values for the defined optional subelements are shown in Table 8-93. A Yes in the Extensible column of a subelement listed in Table 8-93 indicates that the length of the subelement might be extended in future revisions or amendments of this standard. When the Extensible column of an element is Subelement, then the subelement might be extended in future revisions or amendments of this standard by defining additional subelements within the subelement. See 9.24.9.

Table 8-93—Optional subelement IDs for Multicast Diagnostics Report

Subelement ID	Name	Length field (octets)	Extensible
0–220	Reserved		
221	Vendor Specific	1 to 239	
222–255	Reserved		

The summary of fields used in the STA Multicast Diagnostics Report is shown in Table 8-94.

Table 8-94—Summary of fields used in the STA Multicast Diagnostics Report

Field	Measurement Result	Triggered Report
Measurement Time	Yes	Yes
Measurement Duration	Yes	Yes
Group MAC Address	Yes	Yes
Multicast Reporting Reason	Yes	Yes
Multicast Received MSDU Count	Yes	Yes
First Sequence Number	Yes	Yes
Last Sequence Number	Yes	Yes
Multicast Rate	Yes	Yes
Optional Subelements	Optional	Optional

The use of Multicast Diagnostics Report is defined in 10.11.19.

8.4.2.24.13 Location Civic Report

The Location Civic Report includes the location information defined in Civic format for the location subject provided in the Location Civic measurement request, as shown in Figure 8-169.

	Civic Location Type	Optional Subelements	Civic Location
Octets:	1	variable	variable

Figure 8-169—Location Civic Report field format

The Civic Location Type field contains the format of location information in the Civic Location field, as indicated in Table 8-77.

When the Civic Location Type is IETF RFC4776-2006, the Optional Subelements field may optionally include the Location Reference, Location Shape, Map Image, and Vendor Specific subelements as defined in Table 8-95.

When the Civic Location Type value is Vendor Specific, a Vendor Specific subelement is included in the Optional Subelements field that identifies the Organization Identifier corresponding to the Civic Location Type.

The Optional Subelements field format contains zero or more subelements, each consisting of a 1-octet Subelement ID field, a 1-octet Length field, and a variable-length Data field, as shown in Figure 8-402. The optional subelements are ordered by nondecreasing subelement ID.

The Subelement ID field values for the defined optional subelements are shown in Table 8-95. A Yes in the Extensible column of a subelement listed in Table 8-95 indicates that the length of the subelement might be extended in future revisions or amendments of this standard. When the Extensible column of an element is Subelement, then the subelement might be extended in future revisions or amendments of this standard by defining additional subelements within the subelement. See 9.24.9.

Table 8-95—Optional subelement IDs for Location Civic Report

Subelement ID	Name	Length field (octets)	Extensible
0	Reserved		
1	Originator Requesting STA MAC Address	6	No
2	Target MAC Address	6	No
3	Location Reference	Variable	
4	Location Shape	Variable	
5	Map Image	Variable	
6–220	Reserved		
221	Vendor Specific	1 to 237	
222–255	Reserved		

The Originator Requesting STA MAC Address subelement contains the MAC address of the STA that requested the Location Information and it is present whenever the location subject definition field in the corresponding Location Civic Request was set to 2. The format of the Originator Requesting STA MAC Address subelement is shown in Figure 8-126.

The Target MAC Address subelement contains the MAC address of the STA whose Location Information was requested and it is present whenever the location subject definition field in the corresponding Location Civic Request was set to 2. The format of the Target MAC Address subelement is shown in Figure 8-127.

The format of the Location Reference subelement is shown in Figure 8-170.

	Subelement ID	Length	Location Reference
Octets:	1	1	variable

Figure 8-170—Location Reference subelement format

The Location Reference is an ASCII string that defines a position on a floor from which the relative location contained in the Location Shape subelement is offset. A Location Reference value of 0 length indicates that the position of the Location Shape is top north west corner (i.e., 0,0) of the floor plan that on which the Location Shape is defined.

The format of the Location Shape subelement is shown in Figure 8-171.

	Subelement ID	Length	Location Shape ID	Location Shape Value
Octets:	1	1	1	variable

Figure 8-171—Location Shape subelement format

The Location Shape subelement defines the position in meters, including uncertainty, of the entity being located. A Shape is specified with respect to either a 2-Dimensional or 3-Dimensional Coordinate Reference System where each point in the shape defines the direction from the Location Reference starting point. A positive X-axis value corresponds to an easterly direction relative to the Location Reference; a negative X-axis value corresponds to a westerly direction relative to the Location Reference; a positive Y-axis value corresponds to a northerly direction relative to the Location Reference; a negative Y-axis value corresponds to a southerly direction relative to the Location Reference and the Z-axis value corresponds to the altitude above the horizontal plane at the Location Reference.

The Location Shape ID field contains a one-octet identifier that defines the shape contained in the subelement and is one of the values defined in Table 8-96.

Table 8-96—Location Shape IDs

Location Shape ID	Name	Length (octets)
0	Reserved	
1	2-Dimension Point	8
2	3-Dimension Point	12
3	Circle	12
4	Sphere	16
5	Polygon	Variable
6	Prism	Variable
7	Ellipse	18
8	Ellipsoid	26
9	Arcband	20
10–255	Reserved	

The Location Shape Value field contains the location shape value for each corresponding Location Shape ID. The formats of Location Shape Values are described in the following text.

All shape field value units that are 4-octet single precision floating point values are in meters and are represented by binary32 floating point values as defined in IEEE Std 754-2008, with the least significant bit of the fraction occurring in bit 0 of the field.

The format of the 2-Dimension Point Location Shape Value is defined in Figure 8-172.

	X-coordinate	Y-coordinate
Octets:	4	4

Figure 8-172—2-Dimension Point Location Shape Value format

The X-coordinate field contains a 4-octet single precision floating point value.

The Y-coordinate field contains a 4-octet single precision floating point value.

The format of the 3-Dimension Point Location Shape Value is defined in Figure 8-173.

	X-coordinate	Y-coordinate	Z-coordinate
Octets:	4	4	4

Figure 8-173—3-Dimension Point Location Shape Value format

The X-coordinate field contains a 4-octet single precision floating point value.

The Y-coordinate field contains a 4-octet single precision floating point value.

The Z-coordinate field contains a 4-octet single precision floating point value.

The format of the Circle Location Shape Value is defined in Figure 8-174.

	X-coordinate	Y-coordinate	Radius
Octets:	4	4	4

Figure 8-174—Circle Location Shape Value format

The X-coordinate field contains a 4-octet single precision floating point value.

The Y-coordinate field contains a 4-octet single precision floating point value.

The Radius field contains a 4-octet single precision floating point value.

The format of the Sphere Location Shape Value is defined in Figure 8-175.

	X-coordinate	Y-coordinate	Z-coordinate	Radius
Octets:	4	4	4	4

Figure 8-175—Sphere Location Shape Value format

The X-coordinate field contains a 4-octet single precision floating point value.

The Y-coordinate field contains a 4-octet single precision floating point value.

The Z-coordinate field contains a 4-octet single precision floating point value.

The Radius field contains a 4-octet single precision floating point value.

The format of the Polygon Location Shape Value is defined in Figure 8-176.

	Number of Points	List of 2-Dimension Points
Octets:	1	variable

Figure 8-176—Polygon Location Shape Value format

The Number of Points field is a 1 octet unsigned integer that specifies the number of points defined in the polygon. The value 0 is reserved.

The List of 2-Dimension Points is a sequence of 2D Point field values that define the closed polygon.

The format of the Prism Location Shape Value is defined in Figure 8-177.

	Number of Points	List of 3-Dimension Points
Octets:	1	variable

Figure 8-177—Prism Location Shape Value format

The Number of Points field is a 1 octet unsigned integer that specifies the number of points defined in the prism. The value 0 is reserved.

The List of 3-Dimension Points is a sequence of 3-Dimension Point field values that define the closed prism.

The format of the Ellipse Location Shape Value is defined in Figure 8-178.

	X-coordinate	Y-coordinate	Angle	Semi-Major Axis	Semi-Minor Axis
Octets:	4	4	2	4	4

Figure 8-178—Ellipse Location Shape Value format

The X-coordinate field contains a 4-octet single precision floating point value.

The Y-coordinate field contains a 4-octet single precision floating point value.

The Angle field contains a 2-octet unsigned integer between 0 and 359 degrees.

The Semi-Major Axis field contains a 4-octet single precision floating point value.

The Semi-Minor Axis field contains a 4-octet single precision floating point value.

The format of the Ellipsoid Location Shape Value is defined in Figure 8-179.

	X-coordinate	Y-coordinate	Z-coordinate	Angle	Semi-Major Axis	Semi-Minor Axis	Semi-Vertical Axis
Octets:	4	4	4	2	4	4	4

Figure 8-179—Ellipsoid Location Shape Value format

The X-coordinate field contains a 4-octet single precision floating point value.

The Y-coordinate field contains a 4-octet single precision floating point value.

The Angle field contains a 2-octet unsigned integer between 0 and 359 degrees.

The Semi-Major Axis field contains a 4-octet single precision floating point value.

The Semi-Minor Axis field contains a 4-octet single precision floating point value.

The Semi-Vertical Axis field contains a 4-octet single precision floating point value.

The format of the Arcband Location Shape Value is defined in Figure 8-180.

	X-coordinate	Y-coordinate	Inner Radius	Outer Radius	Start Angle	Opening Angle
Octets:	4	4	4	4	2	2

Figure 8-180—Arcband Location Shape Value format

The X-coordinate field contains a 4-octet single precision floating point value.

The Y-coordinate field contains a 4-octet single precision floating point value.

The Inner Radius field contains a 4-octet single precision floating point value.

The Outer Radius field contains a 4-octet single precision floating point value.

The Start Angle field contains a 2-octet unsigned integer between 0 and 359.

The Opening Angle field contains a 2-octet unsigned integer between 0 and 359.

The Map Image subelement contains a map reference that is used in combination with the Location Reference and Location Shape subelements. The format of the Map Image subelement is shown in Figure 8-181.

	Subelement ID	Length	Map Type	Map URL
Octets:	1	1	1	variable

Figure 8-181—Map Image subelement format

The Map Type field is a 1-octet unsigned integer that defines the type of map referred to by the Map URL field, as defined in Table 8-97.

Table 8-97—Map Types

Map Type Value	Name
0	URL Defined
1	Png
2	Gif
3	Jpeg
4	Svg
5	dxf
6	Dwg
7	Dwf
8	cad
9	Tiff
10	gml
11	Kml
12	Bmp
13	Pgm
14	ppm
15	Xbm
16	Xpm
17	ico
18–255	Reserved

The Map Type field value “URL Defined” indicates the Map URL field value has a file extension, defined as a mime type and is self-descriptive.

The Map URL field is a variable-length field formatted in accordance with IETF RFC 3986-2005 and provides the location of a floor map.

The Civic Location field of the Location Civic Report (see Figure 8-169) is a variable octet field and contains the location information in the format as indicated in the Civic Location Type field.

8.4.2.24.14 Location Identifier Report

The Location Identifier Report includes an indirect reference to the location information for the location subject provided in the Location Identifier measurement request, as shown in Figure 8-182.

	Expiration TSF	Optional Subelements	Public Identifier URI
Octets:	8	variable	variable

Figure 8-182—Location Identifier Report field format

The Expiration TSF field is the value of the TSF when the Public Identifier URI field value is no longer valid. The Expiration TSF field set to 0 indicates the Public Identifier URI does not expire.

The Optional Subelements field format contains zero or more subelements, each consisting of a 1-octet Subelement ID field, a 1-octet Length field and a variable-length Data field, as shown in Figure 8-402. Any optional subelements are ordered by nondecreasing subelement ID.

The Subelement ID field values for the defined optional subelements are shown in Table 8-98. A Yes in the Extensible column of a subelement listed in Table 8-98 indicates that the Length of the subelement might be extended in future revisions or amendments of this standard. When the Extensible column of an element is Subelement, then the subelement might be extended in future revisions or amendments of this standard by defining additional subelements within the subelement. See 9.24.9.

Table 8-98—Optional subelement IDs for Location Identifier Report

Subelement ID	Name	Length field (octets)	Extensible
0	Reserved		
1	Originator Requesting STA MAC Address	6	No
2	Target MAC Address	6	No
3–220	Reserved		
221	Vendor Specific	1 to 237	
222–255	Reserved		

The Originator Requesting STA MAC Address subelement contains the MAC address of the STA that requested the Location Information and it is present whenever the location subject definition field in the corresponding Location Identifier Request was set to 2. The format of the Originator Requesting STA MAC Address subelement is shown in Figure 8-126.

The Target MAC Address subelement contains the MAC address of the STA whose Location Information was requested and it is present whenever the location subject definition field in the corresponding Location Identifier Request was set to 2. The format of the Target MAC Address subelement is shown in Figure 8-127.

The Public Identifier URI field contains a value in URI format that points to a location object. It can be used to return the location value for the requesting STA. The format of the location value returned when the URI is dereferenced is dependent on the provider of the URI and is beyond the scope of this document. The Public Identifier URI confirms the validity of the location estimate to an external agent when a STA forwards a location estimate to that agent. The protocol used to query the infrastructure for a location report based on the Public Identifier URI is beyond the scope of this standard.

8.4.2.25 Quiet element

The Quiet element defines an interval during which no transmission occurs in the current channel. This interval may be used to assist in making channel measurements without interference from other STAs in the BSS. The format of the Quiet element is shown in Figure 8-183.

	Element ID	Length	Quiet Count	Quiet Period	Quiet Duration	Quiet Offset
Octets:	1	1	1	1	2	2

Figure 8-183—Quiet element format

The Length field is set to 6.

The Quiet Count field is set to the number of TBTTs until the beacon interval during which the next quiet interval starts. A value of 1 indicates the quiet interval starts during the beacon interval starting at the next TBTT. A value of 0 is reserved.

The Quiet Period field is set to the number of beacon intervals between the start of regularly scheduled quiet intervals defined by this Quiet element. A value of 0 indicates that no periodic quiet interval is defined.

The Quiet Duration field is set to the duration of the quiet interval, expressed in TUs.

The Quiet Offset field is set to the offset of the start of the quiet interval from the TBTT specified by the Quiet Count field, expressed in TUs. The value of the Quiet Offset field is less than one beacon interval.

The Quiet element may be included in Beacon frames, as described in 8.3.3.2, and Probe Response frames, as described in 8.3.3.10. The use of Quiet elements is described in 10.9.3.

8.4.2.26 IBSS DFS element

The IBSS DFS element contains information for DFS operation in an IBSS. The format of the IBSS DFS element is shown in Figure 8-184.

	Element ID	Length	DFS Owner	DFS Recovery Interval	Channel Map (see Figure 8-185)
Octets:	1	1	6	1	2×*n*

Figure 8-184—IBSS DFS element format

	Channel Number	Map
Octets:	1	1

n tuples, one for each supported channel

Figure 8-185—Channel Map field format

The Length field is variable.

The DFS Owner field is set to the individual IEEE MAC address of the STA that is the currently known DFS Owner in the IBSS.

The DFS Recovery Interval field indicates the time interval that is used for DFS owner recovery, expressed as an integral number of beacon intervals. The DFS Recovery Interval value is static throughout the lifetime of the IBSS and is determined by the STA that starts the IBSS.

The Channel Map field shown in Figure 8-185 contains a Channel Number field and a Map field (see 8.4.2.24.2) for each channel supported by the STA transmitting the IBSS DFS element. Note that *n* in

Figure 8-184 is the number of channels supported by the STA.

The IBSS DFS element may be included in Beacon frames, as described in 8.3.3.2, and Probe Response frames, as described in 8.3.3.10. The use of IBSS DFS elements is described in 10.9.8.3.

8.4.2.27 RSNE

8.4.2.27.1 General

The RSNE contains authentication and pairwise cipher suite selectors, a single group data cipher suite selector, an RSN Capabilities field, the PMK identifier (PMKID) count, a PMKID list, and a single group management cipher suite selector. See Figure 8-186. The size of the RSNE is limited by the size of an element, which is 255 octets. Therefore, the number of pairwise cipher suites, AKM suites, and PMKIDs is limited.

	Element ID	Length	Version	Group Data Cipher Suite	Pairwise Cipher Suite Count	Pairwise Cipher Suite List
Octets:	1	1	2	4	2	4 × *m*

	AKM Suite Count	AKM Suite List	RSN Capabilities	PMKIDCount	PMKID List	Group Management Cipher Suite
Octets:	2	4 × *n*	2	2	16 × *s*	4

Figure 8-186—RSNE format

In Figure 8-186, *m* denotes the pairwise cipher suite count, *n* the AKM suite count, and *s* is the PMKID count.

All fields use the bit convention from 8.2.2. The RSNE contains up to and including the Version field. All fields after the Version field are optional. If any nonzero length field is absent, then none of the subsequent fields is included.

Element ID is set to the value for RSN, specified in Table 8-54.

Length gives the number of octets in the Information field (field(s) following the Element ID and Length fields) of the element.

The Version field indicates the version number of the RSNA protocol. The range of Version field values a STA supports is contiguous. Values 0 and 2 or higher of the Version field are reserved. RSN Version 1 is defined in this standard.

NOTE—The following represent sample elements:

802.1X authentication, CCMP pairwise and group data cipher suites (WEP-40, WEP-104, and TKIP not allowed):

```
30, // element id, 48 expressed as Hex value
14, // length in octets, 20 expressed as Hex value
01 00, // Version 1
00 0F AC 04, // CCMP as group data cipher suite
01 00, // pairwise cipher suite count
00 0F AC 04, // CCMP as pairwise cipher suite
01 00, // authentication count
00 0F AC 01 // 802.1X authentication
00 00 // No capabilities
```

802.1X authentication, CCMP pairwise and group data cipher suites (WEP-40, WEP-104 and TKIP not allowed), preauthentication supported:

```
30, // element id, 48 expressed as Hex value
14, // length in octets, 20 expressed as Hex value
01 00, // Version 1
00 0F AC 04, // CCMP as group data cipher suite
01 00, // pairwise cipher suite count
00 0F AC 04, // CCMP as pairwise cipher suite
01 00, // authentication count
00 0F AC 01 // 802.1X authentication
01 00 // Preauthentication capabilities
```

802.1X authentication, Use GTK for pairwise cipher suite, WEP-40 group data cipher suites, optional RSN Capabilities field omitted:

```
30, // element id, 48 expressed as Hex value
12, // length in octets, 18 expressed as Hex value
01 00, // Version 1
00 0F AC 01, // WEP-40 as group data cipher suite
01 00, // pairwise cipher suite count
00 0F AC 00, // Use group key as pairwise cipher suite
01 00, // authentication count
00 0F AC 01 // 802.1X authentication
```

802.1X authentication, Use CCMP for pairwise cipher suite, CCMP group data cipher suites, preauthentication and a PMKID:

```
30, // element id, 48 expressed as Hex value
26 // length in octets, 38 expressed as Hex value
01 00, // Version 1
00 0F AC 04, // CCMP as group data cipher suite
01 00, // pairwise cipher suite count
00 0F AC 04, // CCMP as pairwise cipher suite
01 00, // authentication count
00 0F AC 01 // 802.1X authentication
01 00 // Preauthentication capabilities
01 00 // PMKID Count
01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 // PMKID
```

802.1X authentication, CCMP pairwise and group data cipher suites (WEP-40, WEP-104, and TKIP are not allowed), and management frame protection with AES-128-CMAC as the group management suite selector.

```
30, // element id, 48 expressed as Hex value
1A, // length in octets, 26 expressed as Hex value
01 00, // Version 1
00 0F AC 04, // CCMP as group data cipher suite
01 00, // pairwise cipher suite count
00 0F AC 04, // CCMP as pairwise cipher suite
01 00, // authentication count
00 0F AC 01 // IEEE 802.1X authentication
80 00 // management frame protection is enabled but not required
00 00 // No PMKIDs
00 0F AC 06, // BIP as the broadcast/multicast management cipher suite
```

8.4.2.27.2 Cipher suites

The Group Data Cipher Suite field contains the cipher suite selector used by the BSS to protect group addressed frames.

The Pairwise Cipher Suite Count field indicates the number of pairwise cipher suite selectors that are contained in the Pairwise Cipher Suite List field.

The Pairwise Cipher Suite List field contains a series of cipher suite selectors that indicate the pairwise cipher suites contained in the RSNE.

The Group Management Cipher Suite field contains the cipher suite selector used by the BSS to protect group addressed robust management frames.

When management frame protection is negotiated, the negotiated pairwise cipher suite is used to protect individually addressed robust management frames, and the group management cipher suite is used to protect group addressed robust management frames. Use of AES-128-CMAC is not valid as a data cipher suite.

A suite selector has the format shown in Figure 8-187.

	OUI	Suite Type
Octets:	3	1

Figure 8-187—Suite selector format

The order of the OUI (organizationally unique identifier) field is described in 8.2.2.

Table 8-99 provides the cipher suite selectors defined by this standard.

Table 8-99—Cipher suite selectors

OUI	Suite type	Meaning
00-0F-AC	0	Use group cipher suite
00-0F-AC	1	WEP-40
00-0F-AC	2	TKIP
00-0F-AC	3	Reserved
00-0F-AC	4	CCMP – default pairwise cipher suite and default group cipher suite for data frames in an RSNA
00-0F-AC	5	WEP-104
00-0F-AC	6	BIP—default group management cipher suite in an RSNA with management frame protection enabled
00-0F-AC	7	Group addressed traffic not allowed
00-0F-AC	8–255	Reserved
Vendor OUI	Other	Vendor-specific
Other	Any	Reserved

The cipher suite selector 00-0F-AC:4 (CCMP) is the default cipher suite value.

The cipher suite selectors 00-0F-AC:1 (WEP-40) and 00-0F-AC:5 (WEP-104) are only valid as a group cipher suite in a transition security network (TSN) to allow pre-RSNA devices to join an IBSS or to associate with an infrastructure BSS.

Use of any group cipher suite other than TKIP, WEP-104, or WEP-40 with TKIP as the pairwise cipher suite is not supported.

The cipher suite selector 00-0F-AC:0 (Use group cipher suite) is only valid as the pairwise cipher suite. An AP may specify the selector 00-0F-AC:0 (Use group cipher suite) for a pairwise cipher suite if it does not support any pairwise cipher suites. If an AP specifies 00-0F-AC:0 (Use group cipher suite) as the pairwise cipher selection, this is the only pairwise cipher selection the AP advertises.

If any cipher suite other than TKIP, WEP-104, or WEP-40 is enabled, then the AP supports pairwise keys, and thus the suite selector 00-0F-AC:0 (Use group cipher suite) is not a valid option.

Table 8-100 indicates the circumstances under which each cipher suite is used.

Table 8-100—Cipher suite usage

Cipher suite selector	GTK	PTK	IGTK
Use group key	No	Yes	No
WEP-40	Yes	No	No
WEP-104	Yes	No	No
TKIP	Yes	Yes	No
CCMP	Yes	Yes	No
BIP	No	No	Yes

8.4.2.27.3 AKM suites

The AKM Suite Count field indicates the number of AKM suite selectors that are contained in the AKM Suite List field.

The AKM Suite List field contains a series of AKM suite selectors contained in the RSNE. In an IBSS only a single AKM suite selector may be specified because STAs in an IBSS use the same AKM suite and because there is no mechanism to negotiate the AKMP in an IBSS (see 11.5.5).

Each AKM suite selector specifies an AKMP. Table 8-101 gives the AKM suite selectors defined by this standard. An AKM suite selector has the format shown in Figure 8-187.

Table 8-101—AKM suite selectors

OUI	Suite type	Meaning		
		Authentication type	Key management type	Key derivation type
00-0F-AC	0	Reserved	Reserved	Reserved
00-0F-AC	1	Authentication negotiated over IEEE 802.1X or using PMKSA caching as defined in 11.5.9.3 – RSNA default	RSNA key management as defined in 11.6 or using PMKSA caching as defined in 11.5.9.3 – RSNA default	Defined in 11.6.1.2
00-0F-AC	2	PSK	RSNA key management as defined in 11.6, using PSK	Defined in 11.6.1.2
00-0F-AC	3	FT authentication negotiated over IEEE 802.1X	FT key management as defined in 11.6.1.7	Defined in 11.6.1.7.2
00-0F-AC	4	FT authentication using PSK	FT key management as defined in 11.6.1.7	Defined in 11.6.1.7.2

Table 8-101—AKM suite selectors *(continued)*

OUI	Suite type	Meaning		
		Authentication type	Key management type	Key derivation type
00-0F-AC	5	Authentication negotiated over IEEE 802.1X or using PMKSA caching as defined in 11.5.9.3 with SHA256 Key Derivation	RSNA Key Management as defined in 8.5 or using PMKSA caching as defined in 11.5.9.3, with SHA256 Key Derivation	Defined in 11.6.1.7.2
00-0F-AC	6	PSK with SHA256 Key Derivation	RSNA Key Management as defined in 11.6 using PSK with SHA256 Key Derivation	Defined in 11.6.1.7.2
00-0F-AC	7	TDLS	TPK Handshake	Defined in 11.6.1.7.2
00-0F-AC	8	SAE Authentication with SHA-256 or using PMKSA caching as defined in 11.5.9.3 with SHA-256 key derivation	RSNA key management as defined in 11.6, PMKSA caching as defined in 11.5.9.3 with SHA256 key derivation or authenticated mesh peering exchange as defined in 13.5	Defined in 11.6.1.7.2
00-0F-AC	9	FT authentication over SAE with SHA-256	FT key management defined in 11.6.1.7	Defined in 11.6.1.7.2
00-0F-AC	10–255	Reserved	Reserved	Reserved
Vendor OUI	Any	Vendor-specific	Vendor-specific	Vendor-specific
Other	Any	Reserved	Reserved	Reserved

The AKM suite selector value 00-0F-AC:1 (i.e., Authentication negotiated over IEEE 802.1X with RSNA key management as defined in 11.6 or using PMKSA caching as defined in 11.5.9.3) is the assumed default when the AKM suite selector field is not supplied.

NOTE—The selector value 00-0F-AC:1 specifies only that IEEE Std 802.1X-2004 is used as the authentication transport. IEEE Std 802.1X-2004 selects the authentication mechanism.

The AKM suite selector value 00-0F-AC:8 (i.e., SAE Authentication with SHA-256 or using PMKSA caching as defined in 11.5.9.3 with SHA-256 key derivation) is used when either a password or PSK is used with RSNA key management.

NOTE—Selector values 00-0F-AC:1 and 00-0F-AC:8 can simultaneously be enabled by an Authenticator.

The AKM suite selector value 00-0F-AC:2 (PSK) is used when an alternate form of PSK is used with RSNA key management.

NOTE—Selector values 00-0F-AC:1 and 00-0F-AC:2 can simultaneously be enabled by an Authenticator.

8.4.2.27.4 RSN capabilities

The RSN Capabilities field indicates requested or advertised capabilities. If the RSN Capabilities field is not present, the default value of 0 is used for all the capability subfields.

The length of the RSN Capabilities field is 2 octets. The format of the RSN Capabilities field is as illustrated in Figure 8-188 and described after the figure.

	B0	B1	B2 B3	B4 B5	B6	B7
	Preauthentication	No Pairwise	PTKSA Replay Counter	GTKSA Replay Counter	Management Frame Protection Required (MFPR)	Management Frame Protection Capable (MFPC)
Bits:	1	1	2	2	1	1

	B8	B9	B10	B11	B12	B13	B14 B15
	Reserved	Peerkey Enabled	SPP A-MSDU Capable	SPP A-MSDU Required	PBAC	Extended Key ID for Individually Addressed Frames	Reserved
Bits:	1	1	1	1	1	1	2

Figure 8-188—RSN Capabilities field format

— Bit 0: Preauthentication. An AP sets the Preauthentication subfield of the RSN Capabilities field to 1 to signal it supports preauthentication (see 11.5.9.2) and sets the subfield to 0 when it does not support preauthentication. A non-AP STA sets the Preauthentication subfield to 0.

— Bit 1: No Pairwise. If a STA supports WEP default key 0 simultaneously with a pairwise key (see 11.6.1), then the STA sets the No Pairwise subfield of the RSN Capabilities field to 0.

If a STA does not support WEP default key 0 simultaneously with a pairwise key (see 11.6.1), then the STA sets the No Pairwise subfield of the RSN Capabilities field to 1.

The No Pairwise subfield describes a capability of a non-AP STA. STAs in an IBSS and APs set the No Pairwise subfield to 0.

The No Pairwise subfield is set to 1 only in a TSN and when the pairwise cipher suite selected by the STA is TKIP.

— Bits 2–3: PTKSA Replay Counter. A STA sets the PTKSA Replay Counter subfield of the RSN Capabilities field to the value contained in dot11RSNAConfigNumberofPTKSAReplayCounters. The least significant bit (LSB) of dot11RSNAConfigNumberofPTKSAReplayCounters is put in bit 2. See 11.4.2.6 and 11.4.3.4.4. The meaning of the value in the PTKSA/GTKSA/STKSA Replay Counter subfield is defined in Table 8-102. The number of replay counters per STKSA is the same as the number of replay counters per PTKSA.

Table 8-102—PTKSA/GTKSA/STKSA replay counters usage

Replay counter value	Meaning
0	1 replay counter per PTKSA/GTKSA/STKSA
1	2 replay counters per PTKSA/GTKSA/STKSA
2	4 replay counters per PTKSA/GTKSA/STKSA
3	16 replay counters per PTKSA/GTKSA/STKSA

— Bits 4–5: GTKSA Replay Counter. A STA sets the GTKSA Replay Counter subfield of the RSN Capabilities field to the value contained in dot11RSNAConfigNumberofPTKSAReplayCounters. The LSB of dot11RSNAConfigNumberofGTKSAReplayCounters is put in bit 4. See 11.4.2.6 and 11.4.3.4.4. The meaning of the value in the GTKSA Replay Counter subfield is defined in Table 8-102.

— Bit 6: Management Frame Protection Required (MFPR). A STA sets this bit to 1 to advertise that protection of robust management frames is mandatory. A STA sets this bit to 1 when dot11RSNAProtectedManagementFramesActivated is true and dot11RSNAUnprotectedManagementFramesAllowed is false; otherwise it sets this bit to 0. If a STA sets this bit to 1, then that STA only allows RSNAs with STAs that provide Management Frame Protection.

— Bit 7: Management Frame Protection Capable (MFPC). A STA sets this bit to 1 when dot11RSNAProtectedManagementFramesActivated is true to advertise that protection of robust management frames is enabled.

— Bits 9: PeerKey Enabled. An AP STA sets the PeerKey Enabled subfield of the RSN Capabilities field to 1 to signal it supports PeerKey Handshake (see 11.6.8). This field is used by AP STA to describe its ability to support PeerKey Handshake.

— Bit 10: SPP A-MSDU Capable. A STA sets the SPP A-MSDU Capable subfield of the RSN Capabilities field to 1 to signal that it supports signaling and payload protected A-MSDUs (SPP A-MSDUs) (see 10.18). Otherwise, this subfield is set to 0.

— Bit 11: SPP A-MSDU Required. A STA sets the SPP A-MSDU Required subfield of the RSN Capabilities field to 1 when it allows only SPP A-MSDUs (i.e., does not send or receive payload protected A-MSDUs (PP A-MSDUs) (see 10.18). Otherwise, this subfield is set to 0.

— Bit 12: PBAC (protected block ack agreement capable). A STA sets the PBAC subfield of RSN Capabilities field to 1 to indicate it supports PBAC. Otherwise, this subfield is set to 0.

— Bit 13: Extended Key ID for Individually Addressed Frames. This subfield is set to 1 to indicate that the STA supports Key ID values in the range 0 to 1 for a PTKSA and STKSA when the cipher suite is CCMP. A value of 0 indicates that the STA only supports Key ID 0 for a PTKSA and STKSA.

— Bits 8 and 14–15: Reserved. The remaining subfields of the RSN Capabilities field are reserved.

8.4.2.27.5 PMKID

The PMKID Count and List fields are used only in the RSNE in the (Re)Association Request frame to an AP and in FT authentication sequence frames. The PMKID Count specifies the number of PMKIDs in the PMKID List field. The PMKID list contains 0 or more PMKIDs that the STA believes to be valid for the destination AP. The PMKID can refer to

a) A cached PMKSA that has been obtained through preauthentication with the target AP

b) A cached PMKSA from an EAP or SAE authentication

c) A PMKSA derived from a PSK for the target AP

d) A PMK-R0 security association derived as part of an FT initial mobility domain association

e) A PMK-R1 security association derived as part of an FT initial mobility domain association or as part of a fast BSS transition.

See 11.6.1.3 for the construction of the PMKID, 12.8 for the population of PMKID for fast BSS transitions, and 11.6.1.7 for the construction of PMKR0Name and PMKR1Name.

NOTE—A STA need not insert a PMKID in the PMKID List field if the STA will not be using that PMKSA.

8.4.2.28 Vendor Specific element

The Vendor Specific element is used to carry information not defined in this standard within a single defined format, so that reserved element IDs are not usurped for nonstandard purposes and so that interoperability is more easily achieved in the presence of nonstandard information. The element is in the format shown in Figure 8-189 and requires that the first 3 or more octets of the Information field identify the entity that has defined the content of the particular Vendor Specific element. The length of the Organization Identifier field (see 8.4.1.31) is *j* octets, and the order of this field is described in 8.2.2. The length of the Information field (*n*) is constrained by $j \le n \le 255$. The length of the vendor-specific content is *n–j* octets.

	Element ID	Length	Organization Identifier (see 8.4.1.31)	Vendor-specific content
Octets:	1	1	*j*	*n–j*

Figure 8-189—Vendor Specific element format

Multiple Vendor Specific elements are optionally present in a single frame. Each Vendor Specific element might have a different Organization Identifier value. The number of Vendor Specific elements that may appear in a frame is limited only by the maximum frame size.

8.4.2.29 Extended Capabilities element

The Extended Capabilities element carries information about the capabilities of an IEEE 802.11 STA that augment the Capability Information field (CIF). The format of this element is shown in Figure 8-190.

	Element ID	Length	Capabilities
Octets:	1	1	n

Figure 8-190—Extended Capabilities element format

The Element ID field is set to the value for Extended Capabilities, specified in Table 8-54.

The value of the Length field is equal to the number of octets in the Capabilities field.

The Capabilities field is a bit field indicating the capabilities being advertised by the STA transmitting the element. The length of the Capabilities field is a variable *n*. The Capabilities field is shown in Table 8-103.

Table 8-103—Capabilities field

Bit	Information	Notes
0	20/40 BSS Coexistence Management Support	The 20/40 BSS Coexistence Management Support field indicates support for the 20/40 BSS Coexistence Management frame and its use. The 20/40 BSS Coexistence Management Support field is set to 1 to indicate support for the communication of STA information through the transmission and reception of the 20/40 BSS Coexistence Management frame. The 20/40 BSS Coexistence Management Support field is set to 0 to indicate a lack of support for the communication of STA information through the transmission and reception of the 20/40 BSS Coexistence Management frame.
1	Reserved	

Table 8-103—Capabilities field *(continued)*

Bit	Information	Notes
2	Extended Channel Switching	The Extended Channel Switching field is 1 to indicate support for the communication of channel switching information through the transmission and reception of the Extended Channel Switch Announcement element and management frame, as described in 8.5.8.7. The Extended Channel Switching field is 0 to indicate a lack of support for extended channel switching.
3	Reserved	
4	PSMP Capability	This bit in the Extended Capabilities element is set to 1 if the STA supports PSMP operation described in 9.26. In Beacon and Probe Response frames transmitted by an AP: Set to 0 if the AP does not support PSMP operation Set to 1 if the AP supports PSMP operation In Beacon frames transmitted by a non-AP STA: Set to 0 Otherwise: Set to 0 if the STA does not support PSMP operation Set to 1 if the STA supports PSMP operation
5	Reserved	
6	S-PSMP Support	Indicates support for scheduled PSMP. When PSMP Support is equal to 0, S-PSMP support is set to 0. When PSMP Support is equal to 1, the S-PSMP Support field is defined as follows: Set to 0 if STA does not support S-PSMP Set to 1 if STA supports S-PSMP
7	Event	The STA sets the Event field to 1 when dot11MgmtOptionEventsActivated is true, and sets it to 0 otherwise. See 10.23.2.
8	Diagnostics	The STA sets the Diagnostics field to 1 when dot11MgmtOptionDiagnosticsActivated is true, and sets it to 0 otherwise. See 10.23.3.
9	Multicast Diagnostics	The STA sets the Multicast Diagnostics field to 1 when dot11MgmtOptionMulticastDiagnosticsActivated is true, and sets it to 0 otherwise. See 10.23.3.
10	Location Tracking	The STA sets the Location Tracking field to 1 when dot11MgmtOptionLocationTrackingActivated is true, and sets it to 0 otherwise. See 10.23.4.
11	FMS	The STA sets the FMS field to 1 when dot11MgmtOptionFMSActivated is true, and sets it to 0 otherwise. See 10.2.1.16 and 10.23.7.
12	Proxy ARP Service	The AP sets the Proxy ARP Service field to 1 when dot11MgmtOptionProxyARPActivated is true, and sets it to 0 otherwise. See 10.23.13. A non-AP STA sets the Proxy ARP Service field to 0.
13	Collocated Interference Reporting	The STA sets the Collocated Interference Reporting field to 1 when dot11MgmtOptionCoLocIntfReportingActivated is true, and sets it to 0 otherwise. See 10.23.9.

Table 8-103—Capabilities field *(continued)*

Bit	Information	Notes
14	Civic Location	The STA sets the Civic Location field to 1 when dot11RMCivicMeasurementActivated is true, and sets it to 0 otherwise. See 10.11.9.9.
15	Geospatial Location	The STA sets the Geospatial Location field to 1 when dot11RMLCIMeasurementActivated is true, and sets it to 0 otherwise. See 10.11.9.6.
16	TFS	The STA sets the TFS field to 1 when dot11MgmtOptionTFSActivated is true, and sets it to 0 otherwise. See 10.23.11.
17	WNM-Sleep Mode	The STA sets the WNM-Sleep Mode field to 1 when dot11MgmtOptionWNMSleepModeActivated is true, and sets it to 0 otherwise. See 10.2.1.18.
18	TIM Broadcast	The STA sets the TIM Broadcast field to 1 when dot11MgmtOptionTIMBroadcastActivated is true, and sets it to 0 otherwise. See 10.2.1.17.
19	BSS Transition	The STA sets the BSS Transition field to 1 when dot11MgmtOptionBSSTransitionActivated is true, and sets it to 0 otherwise. See 10.23.6.
20	QoS Traffic Capability	The STA sets the QoS Traffic Capability field to 1 when dot11MgmtOptionQoSTrafficCapabilityActivated is true, and sets it to 0 otherwise. See 10.23.9.
21	AC Station Count	The STA sets the AC Station Count field to 1 when dot11MgmtOptionACStationCountActivated is true, and sets it to 0 otherwise. See 10.23.10.
22	Multiple BSSID	The STA sets the Multiple BSSID field to 1 when dot11MgmtOptionMultiBSSIDActivated is true, and sets it to 0 otherwise. See 10.11.14 and 10.1.3.6.
23	Timing Measurement	The STA sets the Timing Measurement field to 1 when dot11MgmtOptionTimingMsmtActivated is true, and sets it to 0 otherwise. See 10.23.5.
24	Channel Usage	The STA sets the Channel Usage field to 1 when dot11MgmtOptionChannelUsageActivated is true and sets it to 0 otherwise. See 10.23.14.
25	SSID List	The STA sets the SSID List field to 1 when dot11MgmtOptionSSIDListActivated is true, and sets it to 0 otherwise. See 10.1.4.
26	DMS	The STA sets the DMS field to 1 when dot11MgmtOptionDMSActivated is true and sets it to 0 otherwise. See 10.23.15.
27	UTC TSF Offset	The STA sets the UTC TSF Offset field to 1 when dot11MgmtOptionUTCTSFOffsetActivated is true and sets it to 0 otherwise. See 10.21.3.
28	TDLS Peer U-APSD Buffer STA Support	The TDLS Peer U-APSD Buffer STA Support subfield indicates support for the TDLS Peer U-APSD Buffer STA function, as defined in 10.2.1.15. When dot11TDLSPeerUAPSDBufferSTAActivated is true, and to indicate support for TDLS Peer U-APSD on this link, the TDLS Peer U-APSD Buffer STA Support subfield is set to 1. Otherwise, the TDLS Peer U-APSD Buffer STA Support subfield is set to 0 to indicate that this capability is not supported on this link.

Table 8-103—Capabilities field *(continued)*

Bit	Information	Notes
29	TDLS Peer PSM Support	The TDLS Peer PSM Support subfield indicates support for TDLS Peer PSM, as defined in 10.2.1.14. When dot11TDLSPeerPSMActivated is true, and to indicate support for TDLS Peer PSM on this link, the TDLS Peer PSM Support subfield is set to 1. Otherwise, the TDLS Peer PSM Support subfield is set to 0 to indicate that this capability is not supported on this link.
30	TDLS channel switching	When dot11TDLSChannelSwitchingActivated is true, and to indicate that the STA supports TDLS with TDLS Channel Switching on this link as described in 10.22, the TDLS Channel Switching capability subfield is set to 1. Otherwise, the TDLS Channel Switching subfield is set to 0 to indicate that this capability is not supported on this link.
31	Interworking	When dot11InterworkingServiceActivated is true, the Interworking field is set to 1 to indicate the STA supports interworking service as described in 10.24. When dot11InterworkingServiceActivated is false, the Interworking field is set to 0 to indicate the STA does not support this capability.
32	QoS Map	When dot11QosMapActivated is true, the QoS Map field is set to 1 to indicate the STA supports QoS Map service as described in 10.24.9. When dot11QosMapActivated is false, the QoS Map field is set to 0 to indicate the STA does not support this capability.
33	EBR	When dot11EBRActivated is true, the EBR field is set to 1 to indicate the STA supports EBR operation as described in 10.4. When dot11EBRActivated is false, the EBR field is set to 0 to indicate the STA does not support this capability.
34	SSPN Interface	When dot11SSPNInterfaceActivated is true, the SSPN Interface field is set to 1 to indicate the AP supports SSPN Interface service as described in 10.24.5. When dot11SSPNInterfaceActivated is false, the SSPN Interface is set to 0 to indicate the AP does not support this capability. Non-AP STAs set this field to 0.
35	Reserved	
36	MSGCF Capability	When dot11MSGCFActivated is true, the MSGCF Capability field is set to 1 to indicate the non-AP STA supports the MSGCF in 6.4. When dot11MSGCFActivated is false, the MSGCF Capability is set to 0 to indicate the non-AP STA does not support this capability. APs set this field to 0.
37	TDLS Support	The TDLS Support subfield indicates support for TDLS, as defined in 10.22. When dot11TunneledDirectLinkSetupImplemented is true, this field is set to 1 to indicate support for TDLS. The field is set to 0 otherwise, to indicate that TDLS is not supported.
38	TDLS Prohibited	The TDLS Prohibited subfield indicates whether the use of TDLS is prohibited. The field is set to 1 to indicate that TDLS is prohibited and to 0 to indicate that TDLS is allowed.
39	TDLS Channel Switching Prohibited	The TDLS Channel Switching Prohibited subfield indicates whether the use of TDLS Channel Switching is prohibited. The field is set to 1 to indicate that TDLS Channel Switching is prohibited and to 0 to indicate that TDLS Channel Switching is allowed.
40	Reject Unadmitted Frame	When dot11RejectUnadmittedTraffic is true, the Reject Unadmitted Frame bit is set to 1 to indicate that the STA rejects MA-UNITDATA.request primitives for frames belonging to an un-admitted TS. When dot11RejectUnadmittedTraffic is false, the Reject Unadmitted Frame bit is set to 0 to indicate that the STA is not required to reject MA-UNITDATA.request primitives for frames belonging to an un-admitted TS. When dot11RejectUnadmittedTraffic is not present, the Reject Unadmitted frame bit is set to 0.

Table 8-103—Capabilities field *(continued)*

Bit	Information	Notes
41-43	Service Interval Granularity	Duration of the shortest service interval (SI). Used for scheduled PSMP only. This field is defined when the S-PSMP Support field is set to 1; otherwise, it is reserved. See 10.4.6. Set to 0 for 5 ms Set to 1 for 10 ms Set to 2 for 15 ms Set to 3 for 20 ms Set to 4 for 25 ms Set to 5 for 30 ms Set to 6 for 35 ms Set to 7 for 40 ms
44	Identifier Location	The STA sets the Identifier Location field to 1 when dot11RMIdentifierMeasurementActivated is true, and sets it to 0 otherwise. See 10.11.9.10.
45	U-APSD Coexistence	The STA sets the U-APSD Coexistence field to 1 when dot11MgmtOptionUAPSDCoexistenceActivated is true and sets it to 0 otherwise. See 10.2.1.5.2.
46	WNM-Notification	The STA sets the WNM-Notification field to 1 when dot11MgmtOptionWNMNotificationActivated is true and sets it to 0 otherwise. See 10.23.16.
47	Reserved	
48	UTF-8 SSID	The SSID in this BSS is interpreted using UTF-8 encoding
49–*n*	Reserved	

If a STA does not support any of capabilities defined in the Extended Capabilities element, then the STA is not required to transmit the Extended Capabilities element.

NOTE—The fields of the Extended Capabilities element are not dynamic. They are determined by the parameters of the MLME-START.request or MLME-JOIN.request primitive that caused the STA to start or join its current BSS, and they remain unchanged until the next MLME-START.request or MLME-JOIN.request primitive.

8.4.2.30 BSS Load element

The BSS Load element contains information on the current STA population and traffic levels in the BSS. The element information format is defined in Figure 8-191. This element may be used by the STA for vendor-specific AP selection algorithm when roaming.

	Element ID	Length (5)	Station Count	Channel Utilization	Available Admission Capacity
Octets:	1	1	2	1	2

Figure 8-191—BSS Load element format

The STA Count field is interpreted as an unsigned integer that indicates the total number of STAs currently associated with this BSS.

The Channel Utilization field is defined as the percentage of time, linearly scaled with 255 representing 100%, that the AP sensed the medium was busy, as indicated by either the physical or virtual carrier sense (CS) mechanism. When more than one channel is in use for the BSS, the Channel Utilization field value is calculated only for the primary channel. This percentage is computed using the formula,

$$\text{Channel Utilization} = \text{Integer}((\text{channel busy time}/(\text{dot11ChannelUtilizationBeaconIntervals} \times \text{dot11BeaconPeriod} \times 1024)) \times 255),$$

where

channel busy time is defined to be the number of microseconds during which the CS mechanism, as defined in 9.3.2.1, has indicated a channel busy indication,

dot11ChannelUtilizationBeaconIntervals represents the number of consecutive beacon intervals during which the channel busy time is measured. The default value of dot11ChannelUtilizationBeaconIntervals is defined in Annex C.

The Available Admission Capacity field is 2 octets long and contains an unsigned integer that specifies the remaining amount of medium time available via explicit admission control, in units of 32 µs/s. The field is helpful for roaming STAs to select an AP that is likely to accept future admission control requests, but it does not represent an assurance that the HC admits these requests.

8.4.2.31 EDCA Parameter Set element

The EDCA Parameter Set element provides information needed by STAs for proper operation of the QoS facility during the CP. The format of the EDCA Parameter Set element is defined in Figure 8-192.

	Element ID	Length (18)	QoS Info	Reserve d	AC_BE Parameter Record	AC_BK Parameter Record	AC_VI Parameter Record	AC_VO Parameter Record
Octets:	1	1	1	1	4	4	4	4

Figure 8-192—EDCA Parameter Set element

For an infrastructure BSS, the EDCA Parameter Set element is used by the AP to establish policy (by changing default MIB attribute values), to change policies when accepting new STAs or new traffic, or to adapt to changes in offered load. The most recent EDCA parameter set element received by a STA is used to update the appropriate MIB values.

The format of the QoS Info field is defined in 8.4.1.17. The QoS Info field contains the EDCA Parameter Set Update Count subfield, which is initially set to 0 and is incremented each time any of the AC parameters changes. This subfield is used by non-AP STAs to determine whether the EDCA parameter set has changed and requires updating the appropriate MIB attributes.

The formats of AC_BE, AC_BK, AC_VI, and AC_VO Parameter Record fields are identical and are illustrated in Figure 8-193.

	ACI / AIFSN	ECWmin / ECWmax	TXOP Limit
Octets:	1	1	2

Figure 8-193—AC_BE, AC_BK, AC_VI, and AC_VO Parameter Record field format

The format of the ACI/AIFSN field is illustrated in Figure 8-194.

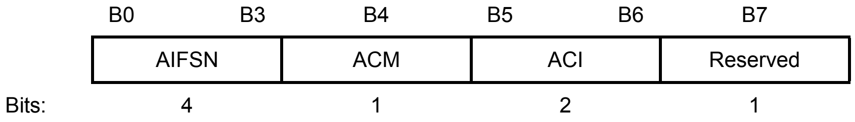


Figure 8-194—ACI/AIFSN field

The value of the AC index (ACI) references the AC to which all parameters in this record correspond. The mapping between ACI and AC is defined in Table 8-104. The ACM (admission control mandatory) subfield indicates that admission control is required for the AC. If the ACM subfield is equal to 0, then there is no admission control for the corresponding AC. If the ACM subfield is set to 1, admission control has to be used prior to transmission using the access parameters specified for this AC. The AIFSN subfield indicates the number of slots after a SIFS duration a STA should defer before either invoking a backoff or starting a transmission. The minimum value for the AIFSN subfield is 2.

Table 8-104—ACI-to-AC coding

ACI	AC	Description
00	AC_BE	Best effort
01	AC_BK	Background
10	AC_VI	Video
11	AC_VO	Voice

The ECWmin and ECWmax fields are illustrated in Figure 8-195.

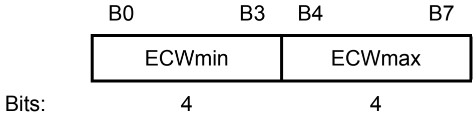


Figure 8-195—ECWmin and ECWmax fields

The ECWmin and ECWmax fields encode the values of CWmin and CWmax, respectively, in an exponent form. The ECWmin and ECWmax values are defined so that

$$\mathrm{CWmin} = 2^{\mathrm{ECWmin}} - 1$$

$$\mathrm{CWmax} = 2^{\mathrm{ECWmax}} - 1$$

Hence the minimum encoded value of CWmin and CWmax is 0, and the maximum value is 32 767.

The value of the TXOP Limit field is specified as an unsigned integer, with the least significant octet transmitted first, in units of 32 µs. A TXOP Limit field value of 0 has a special meaning (see 9.19.2.2).

Table 8-105 defines the default EDCA parameter values used by a non-AP STA if dot11OCBActivated is false.[22]

Table 8-105—Default EDCA Parameter Set element parameter values if dot11OCBActivated is false

AC	CWmin	CWmax	AIFSN	TXOP limit		
				For PHYs defined in Clause 16 and Clause 17	For PHYs defined in Clause 18, Clause 19, and Clause 20	Other PHYs
AC_BK	aCWmin	aCWmax	7	0	0	0
AC_BE	aCWmin	aCWmax	3	0	0	0
AC_VI	(aCWmin+1)/2 – 1	aCWmin	2	6.016 ms	3.008 ms	0
AC_VO	(aCWmin+1)/4 – 1	(aCWmin+1)/2 – 1	2	3.264 ms	1.504 ms	0

If dot11OCBActivated is true, the default EDCA parameter set for STAs transmitting QoS frames is given in Table 8-106.

Table 8-106—Default EDCA parameter set for STA operation if dot11OCBActivated is true

AC	CWmin	CWmax	AIFSN	TXOP Limit OFDM/CCK-OFDM PHY
AC_BK	aCWmin	aCWmax	9	0
AC_BE	aCWmin	aCWmax	6	0
AC_VI	(aCWmin+1)/2–1	aCWmin	3	0
AC_VO	(aCWmin+1)/4–1	(aCWmin+1)/2–1	2	0

8.4.2.32 TSPEC element

The TSPEC element contains the set of parameters that define the characteristics and QoS expectations of a traffic flow, in the context of a particular STA, for use by the HC and STA(s) in support of QoS traffic transfer using the procedures defined in 10.4. The element information format comprises the items as defined in this subclause, and the structure is defined in Figure 8-196.

	Element ID	Length (55)	TS Info	Nominal MSDU Size	Maximum MSDU Size	Minimum Service Interval	Maximum Service Interval	Inactivity Interval	Suspension Interval
Octets:	1	1	3	2	2	4	4	4	4

	Service Start Time	Minimum Data Rate	Mean Data Rate	Peak Data Rate	Burst Size	Delay Bound	Minimum PHY Rate	Surplus Bandwidth Allowance	Medium Time
Octets:	4	4	4	4	4	4	4	2	2

Figure 8-196—TSPEC element format

[22] The default values for TXOP limit are expressed in milliseconds and are multiples of 32 µs.

The TSPEC allows a set of parameters more extensive than may be needed, or may be available, for any particular instance of parameterized QoS traffic. Unless indicated otherwise, fields that follow the TS Info field are set to 0 for any unspecified parameter values. STAs set the value of any parameters to unspecified if they have no information for setting that parameter. The HC may change the value of parameters that have been set unspecified by the STA to any value that it deems appropriate, including leaving them unspecified.

The structure of the TS Info field is defined in Figure 8-197.

	B0	B1 B4	B5 B6	B7 B8	B9	B10	B11 B13	B14 B15	B16	B17 B23
	Traffic Type	TSID	Direction	Access Policy	Aggregation	APSD	User Priority	TSInfo Ack Policy	Schedule	Reserved
Bits:	1	4	2	2	1	1	3	2	1	7

Figure 8-197—TS Info field

The subfields of the TS Info field are defined as follows:

- The Traffic Type subfield is a single bit and is set to 1 for a periodic traffic pattern (e.g., isochronous TS of MSDUs or A-MSDUs, with constant or variable sizes, that are originated at fixed rate) or set to 0 for an aperiodic, or unspecified, traffic pattern (e.g., asynchronous TS of low-duty cycles).
- The TSID subfield is 4 bits in length and contains a value that is a TSID. Note that the MSB (bit 4 in TS Info field) of the TSID subfield is always set to 1.
- The Direction subfield specifies the direction of data carried by the TS as defined in Table 8-107.

Table 8-107—Direction subfield encoding

Bit 5	Bit 6	Usage
0	0	Uplink (MSDUs or A-MSDUs are sent from the non-AP STA to HC)
1	0	Downlink (MSDUs or A-MSDUs are sent from the HC to the non-AP STA)
0	1	Direct link (MSDUs or A-MSDUs are sent from the non-AP STA to another non-AP STA)
1	1	Bidirectional link (equivalent to a downlink request plus an uplink request, each direction having the same parameters). The fields in the TSPEC element specify resources for a single direction. Double the specified resources are required to support both streams.

- The Access Policy subfield is 2 bits in length, specifies the access method to be used for the TS, and is defined in Table 8-108.

Table 8-108—Access Policy subfield

Bit 7	Bit 8	Usage
0	0	Reserved
1	0	Contention-based channel access (EDCA)
0	1	Controlled channel access (HCCA)
1	1	HCCA, EDCA mixed mode (HEMM)

- The Aggregation subfield is 1 bit in length. The Aggregation subfield is valid only when access method is HCCA or when the access method is EDCA and the Schedule subfield is equal to 1. It is set to 1 by a non-AP STA to indicate that an aggregate schedule is required. It is set to 1 by the AP if an aggregate schedule is being provided to the STA. It is set to 0 otherwise. In all other cases, the Aggregation subfield is reserved.

— The APSD subfield is a single bit and is set to 1 to indicate that automatic PS delivery is to be used for the traffic associated with the TSPEC and set to 0 otherwise.

— The UP subfield is 3 bits and indicates the actual value of the UP to be used for the transport of MSDUs or A-MSDUs belonging to this TS in cases where relative prioritization is required. When the TCLAS element is present in the request, the UP subfield in TS Info field of the TSPEC element is reserved.

— The TS Info Ack Policy subfield is 2 bits in length and indicates whether MAC acknowledgments are required for MPDUs or A-MSDUs belonging to this TID and the desired form of those acknowledgments. The encoding of the TS Info Ack Policy subfield is shown in Table 8-109.

Table 8-109—TS Info Ack Policy subfield encoding

Bit 14	Bit 15	Usage
0	0	Normal IEEE 802.11 acknowledgment The addressed recipient returns an ACK or QoS +CF-Ack frame after a SIFS period, according to the procedures defined in 9.3.2.8, 9.4.4, and 9.19.3.5.
1	0	No Ack: The recipient(s) do not acknowledge the transmission.
0	1	Reserved
1	1	Block Ack: A separate Block Ack setup mechanism described in 9.21 is used.

— The Schedule subfield is 1 bit in length and specifies the requested type of schedule. The setting of the subfield when the access policy is EDCA is shown in Table 8-110. When the Access Policy subfield is equal to any value other than EDCA, the Schedule subfield is reserved. When the Schedule and APSD subfields are equal to 1, the AP sets the aggregation bit to 1, indicating that an aggregate schedule is being provided to the STA.

Table 8-110—Setting of Schedule subfield

APSD	Schedule	Usage
0	0	No Schedule
1	0	Unscheduled APSD
0	1	Scheduled PSMP
1	1	Scheduled APSD

The Nominal MSDU Size field is 2 octets long, contains an unsigned integer that specifies the nominal size, in octets, of MSDUs or A-MSDUs belonging to the TS under this TSPEC, and is defined in Figure 8-198. If the Fixed subfield is equal to 1, then the size of the MSDU or A-MSDU is fixed and is indicated by the Size subfield. If the Fixed subfield is equal to 0, then the size of the MSDU or A-MSDU might not be fixed and the Size subfield indicates the nominal MSDU size. If both the Fixed and Size subfields are equal to 0, then the nominal MSDU size is unspecified.

	B0 B15	B15
	Size	Fixed
Bits:	15	1

Figure 8-198—Nominal MSDU Size field

The Maximum MSDU Size field is 2 octets long and contains an unsigned integer that specifies the maximum size, in octets, of MSDUs or A-MSDUs belonging to the TS under this TSPEC.

The Minimum Service Interval field is 4 octets long and contains an unsigned integer that specifies the minimum interval, in microseconds, between the start of two successive SPs.

The Maximum Service Interval field is 4 octets long and contains an unsigned integer that specifies the maximum interval, in microseconds, between the start of two successive SPs.

The Inactivity Interval field is 4 octets long and contains an unsigned integer that specifies the minimum amount of time, in microseconds, that may elapse without arrival or transfer of an MPDU belonging to the TS before this TS is deleted by the MAC entity at the HC.

The Suspension Interval field is 4 octets long and contains an unsigned integer that specifies the minimum amount of time, in microseconds, that may elapse without arrival or transfer of an MSDU belonging to the TS before the generation of successive QoS(+)CF-Poll is stopped for this TS. A value of 4 294 967 295 (= $2^{32} – 1$) disables the suspension interval, indicating that polling for the TS is not to be interrupted based on inactivity. The value of the suspension interval is always less than or equal to the inactivity interval.

The Service Start Time field is 4 octets and contains an unsigned integer that specifies the time, expressed in microseconds, when the first scheduled SP starts. The service start time indicates to the AP the time when a STA first expects to be ready to send frames and a power-saving STA needs to be awake to receive frames. This may help the AP to schedule service so that the MSDUs encounter small delays in the MAC and help the power-saving STAs to reduce power consumption. The field represents the four lower order octets of the TSF timer at the start of the SP. If APSD subfield is 0, this field is also set to 0 (unspecified).

The Minimum Data Rate field is 4 octets long and contains an unsigned integer that specifies the lowest data rate specified at the MAC_SAP, in bits per second, for transport of MSDUs or A-MSDUs belonging to this TS within the bounds of this TSPEC. The minimum data rate does not include the MAC and PHY overheads incurred in transferring the MSDUs or A-MSDUs.

The Mean Data Rate[23] field is 4 octets long and contains an unsigned integer that specifies the average data rate specified at the MAC_SAP, in bits per second, for transport of MSDUs or A-MSDUs belonging to this TS within the bounds of this TSPEC. The mean data rate does not include the MAC and PHY overheads incurred in transferring the MSDUs or A-MSDUs.

The Peak Data Rate field is 4 octets long and contains an unsigned integer that specifies the maximum allowable data rate, in bits per second, for transfer of MSDUs or A-MSDUs belonging to this TS within the bounds of this TSPEC. If p is the peak rate in bits per second, then the maximum amount of data, belonging to this TS, arriving in any time interval [$t1$,$t2$], where $t1 < t2$ and $t2 – t1 > 1$ TU, does not exceed $p \times (t2 – t1)$ bits.

[23] The mean data rate, the peak data rate, and the burst size are the parameters of the token bucket model, which provides standard terminology for describing the behavior of a traffic source. The token bucket model is described in IETF RFC 2212-1997 [B27], IETF RFC 2215-1997 [B28], and IETF RFC 3290-2002 [B33].

The Burst Size field is 4 octets long and contains an unsigned integer that specifies the maximum burst, in octets, of the MSDUs or A-MSDUs belonging to this TS that arrive at the MAC_SAP at the peak data rate. A value of 0 indicates that there are no bursts.

The Delay Bound field is 4 octets long and contains an unsigned integer that specifies the maximum amount of time, in microseconds, allowed to transport an MSDU or A-MSDU belonging to the TS in this TSPEC, measured between the time marking the arrival of the MSDU, or the first MSDU of the MSDUs constituting an A-MSDU, at the local MAC sublayer from the local MAC_SAP and the time of completion of the successful transmission or retransmission of the MSDU or A-MSDU to the destination. The completion of the MSDU or A-MSDU transmission includes the relevant acknowledgment frame transmission time, if present.

The Minimum PHY Rate field is 4 octets long and contains an unsigned integer that specifies the desired minimum PHY rate to use for this TS, in bits per second, that is required for transport of the MSDUs or A-MSDUs belonging to the TS in this TSPEC.[24] See 10.4.2 for constraints on the selection of this field.

The Surplus Bandwidth Allowance field is 2 octets long and specifies the excess allocation of time (and bandwidth) over and above the stated application rates required to transport an MSDU or A-MSDU belonging to the TS in this TSPEC. This field is represented as an unsigned binary number and, when specified, is greater than 0. The 13 least significant bits (LSBs) indicate the decimal part while the three MSBs indicate the integer part of the number. This field takes into account the retransmissions, as the rate information does not include retransmissions. It represents the ratio of over-the-air bandwidth (i.e., time that the scheduler allocates for the transmission of MSDUs or A-MSDUs at the required rates) to bandwidth of the transported MSDUs or A-MSDUs required for successful transmission (i.e., time that would be necessary at the minimum PHY rate if there were no errors on the channel) to meet throughput and delay bounds under this TSPEC, when specified. As such, it should be greater than unity. A value of 1 indicates that no additional allocation of time is requested.

The Medium Time field is a 16-bit unsigned integer and contains the amount of time admitted to access the medium, in units of 32 µs/s. This field is reserved in the ADDTS Request frame and is set by the HC in the ADDTS Response frame. The derivation of this field is described in N.2.2. This field is not used for controlled channel access.

The UP, Minimum Data Rate, Mean Data Rate, Peak Data Rate, Burst Size, Minimum PHY Rate, and Delay Bound fields in a TSPEC element express the QoS expectations requested by a STA, if this TSPEC was issued by that STA, or provided by the HC, if this TSPEC was issued by the HC, when these fields are specified with nonzero values. Unspecified parameters in these fields as indicated by a value of 0 indicate that the STA does not have specific requirements for these parameters if the TSPEC was issued by that STA or that the HC does not provide any specific values for these parameters if the TSPEC was issued by the HC.

8.4.2.33 TCLAS element

The TCLAS element specifies an element that contains a set of parameters necessary to identify incoming MSDUs (from a higher layer in all STAs or from the DS in an AP) that belong to a particular TS. The TCLAS element is also used when the traffic does not belong to a TS, for example, by the FMS, DMS, and TFS services. If required, the TCLAS element is provided in ADDTS Request and ADDTS Response frames only for the downlink or bidirectional links. TCLAS element need not be provided for the uplink or direct- link transmissions. The structure of this element is shown in Figure 8-199.

[24] This rate information is intended to confirm that the TSPEC parameter values resulting from an admission control negotiation are sufficient to provide the required throughput for the TS. In a typical implementation, a TS is admitted only if the defined traffic volume can be accommodated at the specified rate within an amount of WM occupancy time that the admissions control entity is willing to allocate to this TS.

	Element ID	Length (L+1)	User Priority	Frame Classifier
Octets:	1	1	1	variable

Figure 8-199—TCLAS element format

The UP field contains the value of the UP of the associated MSDUs.

The Frame Classifier field is 3–255 octets in length and is defined in Figure 8-200.

	Classifier Type	Classifier Mask	Classifier Parameters
Octets:	1	1	1–252

Figure 8-200—Frame Classifier field

The Frame Classifier field comprises the following subfields: Classifier Type, Classifier Mask, and Classifier Parameters. The Classifier Type subfield is 1 octet in length and specifies the type of classifier parameters in this TCLAS as defined in Table 8-111.

Table 8-111—Frame classifier type

Classifier type	Classifier parameters
0	Ethernet parameters
1	TCP/UDP IP parameters
2	IEEE 802.1Q parameters
3	Filter Offset parameters
4	IP and higher layer parameters
5	IEEE 802.1D/Q parameters
6–255	Reserved

The Classifier Mask subfield specifies a bitmap in which bits that have the value 1 identify a subset of the classifier parameters whose values need to match those of the corresponding parameters in a given MSDU for that MSDU to be classified to the TS of the affiliated TSPEC. The bitmap is ordered from the LSB to the MSB, with each bit pointing to one of the classifier parameters of the same relative position as shown in this subclause based on classifier type. An incoming MSDU that failed to be classified to a particular TS may be classified to another active TS based on the frame classifier for that TS. If, however, all the frame classifiers for the active TS have been exhausted, the MSDU does not belong to any active TS and is classified to be a best-effort MSDU. In cases where there are more bits in the bitmap than classifier parameters that follow, the MSBs that do not point to any classifier parameters are reserved.

For Classifier Type 0, the classifier parameters are the following parameters contained in an Ethernet packet header: Source Address, Destination Address, and Type (“Ethernet” [B12]). The endianness of the Type field is as defined in Ethernet [B12]. The Frame Classifier field for Classifier Type 0 is defined in Figure 8-201. It has a length of 16 octets.

	Classifier Type (0)	Classifier Mask	Source Address	Destination Address	Type
Octets:	1	1	6	6	2

Figure 8-201—Frame Classifier field of Classifier Type 0

For Classifier Type 1, frame classifier is defined for both IPv4 and IPv6, shown in Figure 8-202 and Figure 8-203, and distinguished by the Version field. Use of Classifier Type 1 for IPv6 is deprecated and replaced by Classifier Type 4. The subfields in the classifier parameters are represented and transmitted in the big-endian format. The classifier parameters are the following parameters:

- In a TCP or UDP header: Source Address, Destination Address, Source Port, Destination Port, and Version, plus
- One of the following:
  - In an IPv4 header: Differentiated Services Code Point (DSCP) (IETF RFC 2474-1998 [B29]) and Protocol, or
  - In an IPv6 header: Flow Label.

	Classifier Type (1)	Classifier Mask	Version	Source IP Address	Destination IP Address	Source Port	Destination Port	DSCP	Protocol	Reserved
Octets:	1	1	1	4	4	2	2	1	1	1

Figure 8-202—Frame Classifier field of Classifier Type 1 for traffic over IPv4

	Classifier Type (1)	Classifier Mask	Version	Source IP Address	Destination IP Address	Source Port	Destination Port	Flow Label
Octets:	1	1	1	16	16	2	2	3

Figure 8-203—Frame Classifier field of Classifier Type 1 for traffic over IPv6

The DSCP field contains the value in the 6 LSBs, and the 2 MSBs are set to 0. The 2 MSBs of the DSCP field are ignored for frame classification.

The value in the Version subfield is the value specified in IETF RFC 791-1981 or IETF RFC 2460-1998.

For Classifier Type 2, the Classifier Parameter is the IEEE 802.1Q-2003 [B22] VLAN Tag TCI. The endianness of the 802.1Q VLAN TCI field is as defined IEEE 802.1Q-2003 [B22] for the VLAN Tag TCI. The Frame Classifier field for Classifier Type 2 is defined in Figure 8-204.

	Classifier Type (2)	Classifier Mask	802.1Q VLAN TCI
Octets:	1	1	2

Figure 8-204—Frame Classifier field of Classifier Type 2

For Classifier Type 3, the classifier parameters are defined by a Filter Offset subfield, a Filter Value subfield and a Filter Mask subfield. The Frame Classifier subfield of Classifier Type 3 is defined in Figure 8-205. It has a variable length.

	Classifier Type (3)	Classifier Mask	Filter Offset	Filter Value	Filter Mask
Octets:	1	1	2	variable	variable

Figure 8-205—Frame Classifier field of Classifier Type 3

The Classifier Mask subfield is reserved.

The value of the Filter Offset subfield is the number of octets from the beginning of the MSDU or MMPDU at which the Filter Value is compared. A value of 0 for the Filter Offset indicates that the Filter Value subfield is to be compared to the first octet of the payload prior to encryption following the MAC header.

The length of the Filter Value and Filter Mask subfields is (*Length* – 5)/2, where *Length* is the value in the Length field of the TCLAS element.

The Filter Value subfield is an octet string that is compared to the MSDU or MMDPU content, beginning at the octet indicated by the Filter Offset.

The Filter Mask subfield is an octet string that is used to indicate which bits in the Filter Value subfield are compared. The length of the Filter Mask subfield is equal to the length of the Filter Value subfield. A bit in the Filter Value subfield is only compared if the matching bit in the Filter Mask subfield is 1.

For Classifier Type 4, frame classifier is defined for both IPv4 and IPv6, shown in Figure 8-206 (with the Classifier Type subfield set to 4) and Figure 8-207, and distinguished by the Version subfield. The classifier parameters represent corresponding values in a received IPv4 or IPv6 frame and are defined in Table 8-112. The subfields in the classifier parameters are represented and transmitted in big-endian format.

Table 8-112—Classifier Parameters for Classifier Type 4

Subfield	Included in IPv4	IPv4 field length (octets)	Included in IPv6	IPv6 field length (octets)
Version	Yes	1	Yes	1
Source IP Address	Yes	4	Yes	16
Destination IP Address	Yes	4	Yes	16
Source Port	Yes	2	Yes	2
Destination Port	Yes	2	Yes	2
DSCP	Yes	1	Yes	1
Protocol	Yes	1	No	n/a
Next Header	No	n/a	Yes	1
Flow Label	No	n/a	Yes	3

The Frame Classifier subfield of Classifier Type 4 for traffic over IPv4 is shown in Figure 8-206.

	Classifier Type (4)	Classifier Mask	Version(4)	Source IP Address	Destination IP Address
Octets:	1	1	1	4	4

	Source Port	Destination Port	DSCP	Protocol	Reserved
Octets:	2	2	1	1	1

Figure 8-206—Frame Classifier subfield of Classifier Type 4 for traffic over IPv4

The Frame Classifier subfield of Classifier Type 4 for traffic over IPv6 is shown in Figure 8-207.

	Classifier Type (4)	Classifier Mask	Version(6)	Source IP Address	Destination IP Address
Octets:	1	1	1	16	16

	Source Port	Destination Port	DSCP	Next Header	Flow Label
Octets:	2	2	1	1	3

Figure 8-207—Frame Classifier subfield of Classifier Type 4 for traffic over IPv6

NOTE—Frame classification when extension headers are used is supported only if the TCLAS does not classify on ports (Classifier Mask has the Source and Destination Port bits set to 0).

The value in the Version subfield is the value specified in IETF RFC 791-1981 or IETF RFC 2460-1998.

The DSCP subfield contains the value as described in IETF RFC 2474-1998 in the 6 least significant bits. The 2 most significant bits are reserved.

The Next Header subfield contains the next encapsulated protocol and is compatible with the values specified for the IPv4 Protocol subfield. In the presence of options in the IPv6 header, the Next Header subfield specifies the presence of one or more out of six extension headers as defined in IETF RFC 2460-1998.

The Flow Label subfield contains the value in the 20 least significant bits. The 4 most significant bits are reserved.

NOTE—For example, the flow label 0x12345 is represented as the octet sequence 0x01, 0x23, 0x45.

A TCLAS element is valid when the Classifier Mask Version bit is 1.

For Classifier Type 5, the classifier parameters are the following parameters in an IEEE 802.1Q-2003 [B22] tag header: Priority Code Point (PCP; equivalent to IEEE 802.1D-2004 [B20] User Priority), Canonical Format Indicator (CFI), and VLAN ID (VID).

The Frame Classifier field for Classifier Type 5 is defined in Figure 8-208.

	Classifier Type (4)	Classifier Mask	802.1Q PCP	802.1Q CFI	802.1Q VID
Octets:	1	1	1	1	2

Figure 8-208—Frame Classifier field of Classifier Type 5

The subfields in the classifier parameters are represented and transmitted in big-endian format.

The PCP subfield contains the value in the 4 LSBs; the 4 MSBs are reserved.

The CFI subfield contains the value in the LSB; the 7 MSBs are reserved.

The VID subfield contains the value in the 12 LSBs; the 4 MSBs are reserved.

8.4.2.34 TS Delay element

The TS Delay element is used in the ADDTS Response frame transmitted by the HC and indicates the time after which the ADDTS may be retried. The TS Delay element is defined in Figure 8-209.

	Element ID	Length (4)	Delay
Octets:	1	1	4

Figure 8-209—TS Delay element

The Delay field is 4 octets long and specifies the amount of time, in TUs, a STA should wait before it reinitiates setup of a TS.

The Delay field is set to 0 when an AP does not expect to serve any TSPECs for an indeterminate time and does not know this time a priori.

8.4.2.35 TCLAS Processing element

The TCLAS Processing element is present in the ADDTS Request, ADDTS Response, FMS Request, DMS Request, and TFS Request frames if there are multiple TCLASs associated with the request. It indicates how an MSDU received from higher layers should be processed by the classifier. The TCLAS Processing element is defined in Figure 8-210.

	Element ID	Length (1)	Processing
Octets:	1	1	1

Figure 8-210—TCLAS Processing element

The Processing subfield is 1 octet long. The encoding of the Processing subfield is shown in Table 8-113.

Table 8-113—Encoding of Processing subfield

Processing subfield value	Meaning
0	Incoming MSDU's higher layer parameters have to match to the parameters in all the associated TCLAS elements.
1	Incoming MSDU's higher layer parameters have to match to at least one of the associated TCLAS elements.
2	Incoming MSDUs that do not belong to any other TS are classified to the TS for which this TCLAS Processing element is used. In this case, there are not any associated TCLAS elements.
3–255	Reserved

8.4.2.36 Schedule element

The Schedule element is transmitted by the HC to a STA to announce the schedule that the HC/AP follows for admitted streams originating from or destined to that STA in the future. The information in this element may be used by the STA for power management, internal scheduling, or any other purpose. The element information format is shown in Figure 8-211.

	Element ID	Length (12)	Schedule Info	Service Start Time	Service Interval	Specification Interval
Octets:	1	1	2	4	4	2

Figure 8-211—Schedule element

The Schedule Info field is shown in Figure 8-212.

	B0	B1 B4	B5 B6	B7 B15
	Aggregation	TSID	Direction	Reserved
Bits:	1	4	2	9

Figure 8-212—Schedule Info field

The Aggregation subfield is set to 1 if the schedule is an aggregate schedule for all TSIDs associated with the STA to which the frame is directed. It is set to 0 otherwise. The TSID subfield is as defined in 8.2.4.5.2 and indicates the TSID for which this schedule applies. The Direction subfield is as defined in 8.4.2.32 and defines the direction of the TSPEC associated with the schedule. The TSID and Direction subfields are valid only when the Aggregation subfield is 0. If the Aggregation subfield is 1, the TSID and Direction subfields are reserved.

The Service Start Time field is 4 octets and indicates the anticipated time, expressed in microseconds, when service starts and represents the lower order 4 octets of the TSF timer value at the start of the first SP. The AP uses this field to confirm or modify the service start time indicated in the TSPEC request.

The Service Interval field is 4 octets and indicates the time, expressed in microseconds, between two successive SPs and represents the measured time from the start of one SP to the start of the next SP.

The Specification Interval field is 2 octets long and contains an unsigned integer that specifies the time interval, in TUs, to verify schedule conformance.

The HC may set the Service Start Time field and the Service Interval field to 0 (unspecified) for nonpowersaving STAs.

8.4.2.37 QoS Capability element

The QoS Capability element contains a number of subfields that are used to advertise optional QoS capabilities at a QoS STA. The QoS Capability element is present in Beacon frames that do not contain the EDCA Parameter Set element and in (Re)Association Request frames. The QoS Capability element is defined in Figure 8-213.

	Element ID	Length (1)	QoS Info
Octets:	1	1	1

Figure 8-213—QoS Capability element format

The QoS Info field is 1 octet in length and is defined in 8.4.1.17.

8.4.2.38 AP Channel Report element

The AP Channel Report element contains a list of channels where a STA is likely to find an AP. The format of the AP Channel Report element is shown in Figure 8-214. See 10.11.6 for details.

	Element ID	Length	Operating Class	Channel List
Octets:	1	1	1	variable

Figure 8-214—AP Channel Report element format

The Element ID field is equal to the AP Channel Report value in Table 8-54.

The Length field in octets is variable and depends on the number of channels reported in the Channel List. The minimum value of the Length field is 1 (based on a minimum length for the channel list field of 0 octets).

Operating Class contains an enumerated value from Annex E, specifying the operating class in which the Channel List is valid. An AP Channel Report only reports channels for a single operating class. Multiple AP Channel Report elements are present when reporting channels in more than one operating class.

The Channel List contains a variable number of octets, where each octet describes a single channel number. Channel numbering is dependent on Operating Class according to Annex E.

8.4.2.39 Neighbor Report element

The format of the Neighbor Report element is shown in Figure 8-215.

	Element ID	Length	BSSID	BSSID Information	Operating Class	Channel Number	PHY Type	Optional Subelements
Octets:	1	1	6	4	1	1	1	variable

Figure 8-215—Neighbor Report element format

The Element ID field is equal to the Neighbor Report value in Table 8-54.

The Length field in octets is variable and depends on the number and length of optional subelements. Each Report element describes an AP and consists of BSSID, BSSID Information, Channel Number, Operating Class, PHY Type, and optionally includes optional subelements. The minimum value of the Length field is 13 (i.e., with no optional subelements in the Neighbor Report element).

The BSSID is the BSSID of the BSS being reported. The subsequent fields in the Neighbor Report element pertain to this BSS.

The BSSID Information field can be used to help determine neighbor service set transition candidates. It is 4 octets in length and contains the subfields as shown in Figure 8-216.

	B0 B1	B2	B3	B4 B9	B10	B11	B12 B31
	AP Reachability	Security	Key Scope	Capabilities	Mobility Domain	High Throughput	Reserved
Bits:	2	1	1	6	1	1	20

Figure 8-216—BSSID Information field

The AP Reachability field indicates whether the AP identified by this BSSID is reachable by the STA that requested the neighbor report. For example, the AP identified by this BSSID is reachable for the exchange of preauthentication frames as described in 11.5.9.2. The values are shown in Table 8-114.

Table 8-114—Reachability field

Value	Reachability	Usage
0	Reserved	Not used.
1	Not Reachable	A station sending a preauthentication frame to the BSSID will not receive a response even if the AP indicated by the BSSID is capable of preauthentication.
2	Unknown	The AP is unable to determine if the value Reachable or Not Reachable is to be returned.
3	Reachable	The station sending a preauthentication frame to the BSSID can receive a response from an AP that is capable of preauthentication.

The Security bit, if 1, indicates that the AP identified by this BSSID supports the same security provisioning as used by the STA in its current association. If the bit is 0, it indicates either that the AP does not support the same security provisioning or that the security information is not available at this time.

The Key Scope bit, when set, indicates the AP indicated by this BSSID has the same authenticator as the AP sending the report. If this bit is 0, it indicates a distinct authenticator or the information is not available.

The Capabilities Subfield contains selected capability information for the AP indicated by this BSSID. The bit fields within this subfield have the same meaning and are set to the equivalent bits within the Capability Information field (see 8.4.1.4) being sent in the beacons by the AP being reported. The format of the Capabilities subfield is as in Figure 8-217.

	B4	B5	B6	B7	B8	B9
	Spectrum Management	QoS	APSD	Radio Measurement	Delayed Block Ack	Immediate Block Ack
Bits:	1	1	1	1	1	1

Figure 8-217—Capabilities subfield

The Mobility Domain bit is set to 1 to indicate that the AP represented by this BSSID is including an MDE in its Beacon frames and that the contents of that MDE are identical to the MDE advertised by the AP sending the report.

The High Throughput bit is set to 1 to indicate that the AP represented by this BSSID is an HT AP including the HT Capabilities element in its Beacons, and that the contents of that HT Capabilities element are identical to the HT Capabilities element advertised by the AP sending the report.

Bits 12–31 are reserved.

Operating Class indicates the channel set of the AP indicated by this BSSID. Country, Operating Class, and Channel Number together specify the channel frequency and spacing for the AP indicated by this BSSID. Valid values of Operating Class are shown in Annex E.

Channel Number indicates the last known operating channel of the AP indicated by this BSSID. Channel Number is defined within an Operating Class as shown in Annex E.

The PHY Type field indicates the PHY type of the AP indicated by this BSSID. It is an integer value coded according to the value of the dot11PHYType.

The Optional Subelements field format contains zero or more subelements, each consisting of a 1-octet Subelement ID field, a 1-octet Length field, and a variable-length Data field, as shown in Figure 8-402. Any optional subelements are ordered by nondecreasing subelement ID.

The Subelement ID field values for the defined optional subelements are shown in Table 8-115. A Yes in the Extensible column of a subelement listed in Table 8-115 indicates that the Length of the subelement might be extended in future revisions or amendments of this standard. When the Extensible column of an element is equal to Subelements, then the subelement might be extended in future revisions or amendments of this standard by defining additional subelements within the subelement. See 9.24.9.

Table 8-115—Optional subelement IDs for neighbor report

Subelement ID	Name	Length field (octets)	Extensible
0	Reserved		
1	TSF Information	4	Yes
2	Condensed Country String	2	Yes
3	BSS Transition Candidate Preference	1	
4	BSS Termination Duration	12	
5	Bearing	8	
6–44	Reserved		
45	HT Capabilities subelement	26	Yes
46–60	Reserved		
61	HT Operation subelement	22	Yes
62	Secondary Channel Offset subelement	1	
63–65	Reserved		
66	Measurement Pilot Transmission	1 to 238	Subelements
67–69	Reserved		
70	RM Enabled Capabilities	5	Yes
71	Multiple BSSID	1 to 238	Subelements
72–220	Reserved		
221	Vendor Specific	1 to 238	
222–255	Reserved		

The TSF Information subelement contains TSF Offset and Beacon Interval subfields as shown in Figure 8-218.

	Subelement ID	Length	TSF Offset	Beacon Interval
Octets:	1	1	2	2

Figure 8-218—TSF Information subelement format

The value of the TSF Information subelement Length field in octets is 4.

The TSF Offset subfield is 2 octets long and contains the neighbor AP’s TSF timer offset. This is the time difference, in TU units, between the serving AP and a neighbor AP. This offset is given modulo the neighbor AP’s Beacon Interval and rounded to the nearest TU boundary.

The Beacon Interval field is the beacon interval of the Neighbor AP indicated by this BSSID. This field is defined in 8.4.1.3 and illustrated in Figure 8-37.

The Condensed Country String subelement is set to the first two octets of the value contained in dot11CountryString. This subelement is present only if the country of the neighbor AP indicated by the BSSID differs from the country of the AP that sent this neighbor report.

The Measurement Pilot Transmission subelement has the same format as the Measurement Pilot Transmission element (see 8.4.2.44). The Measurement Pilot Interval subelement is not included if the reported AP is not transmitting Measurement Pilot frames or if the Measurement Pilot Interval of the reported AP is unknown.

The HT Capabilities subelement is the same as the HT Capabilities element as defined in 8.4.2.58.

The HT Operation subelement is the same as the HT Operation element as defined in 8.4.2.59.

The Secondary Channel Offset subelement is the same as the Secondary Channel Offset element as defined in 8.4.2.22.

The RM Enabled Capabilities subelement has the same format as the RM Enabled Capabilities element (see 8.4.2.47).

The Multiple BSSID subelement has the same format as the Multiple BSSID element (see 8.4.2.48). The reference BSSID for the Multiple BSSID subelement is the BSSID field in the Neighbor Report element. This subelement is not present if the neighbor AP is not a member of a Multiple BSSID Set with two or more members or its membership is unknown. (see 10.11.14).

The format of the BSS Transition Candidate Preference subelement field is shown in Figure 8-219.

	Subelement ID	Length	Preference
Octets:	1	1	1

Figure 8-219— BSS Transition Candidate Preference subelement field format

The value of the Length field in the BSS Transition Candidate Information subelement is 1.

The Preference field indicates the network preference for BSS transition to the BSS listed in this BSS Transition Candidate List Entries field in the BSS Transition Management Request frame, BSS Transition Management Query frame, and BSS Transition Management Response frame. The Preference field value is a number ranging from 0 to 255, as defined in Table 8-116, indicating an ordering of preferences for the BSS transition candidates for this STA. Additional details describing use of the Preference field are provided in 10.23.6.

Table 8-116—Preference field values

Preference field value	Description
0	Excluded BSS; reserved when present in the BSS Transition Management Query or BSS Transition Management Response frames.
1–255	Relative values used to indicate the preferred ordering of BSSs, with 255 indicating the most preferred candidate and 1 indicating the least preferred candidate.

The BSS Termination TSF field contained in the BSS Termination Duration subelement is the TSF time of the BSS transmitting the neighbor report that corresponds to the time when termination of the neighbor BSS occurs. How the BSS determines the neighbor BSS termination time is out of scope of the standard. The format of the BSS Termination Duration subelement field is shown in Figure 8-220.

	Subelement ID	Length	BSS Termination TSF	Duration
Octets:	1	1	8	2

Figure 8-220—BSS Termination Duration subelement field format

The value of the Length field in the BSS Termination Duration Information subelement is 10.

The BSS Termination TSF field indicates the value of the TSF counter when BSS termination will occur in the future. A BSS Termination TSF field value of 0 indicates that termination of the BSS will occur imminently. Prior to termination of the BSS, all associated STAs are disassociated by the AP.

The Duration field is an unsigned 2-octet integer that indicates the number of minutes for which the BSS is not present. The Duration field value of 0 is reserved. The Duration field value is 65 535 when the BSS is terminated for a period longer than or equal to 65 535 minutes.

The format of the Bearing subelement field is shown in Figure 8-221.

	Subelement ID	Length	Bearing	Distance	Relative Height
Octets:	1	1	2	4	2

Figure 8-221—Bearing subelement field format

The value of the Bearing Information subelement length field is 4.

The Bearing field specifies the direction that the neighbor, specified by the BSSID field in the Neighbor Report element, is positioned, relative to the reporting BSS and defined in relation to true north, increasing clockwise, measured in degrees from 0 degree to 359 degrees. If the Bearing value is unknown, the subelement is not included.

The Distance field specifies the distance that the neighbor, specified by the BSSID field in the Neighbor Report element, is positioned relative to the reporting BSS as a 4-octet single precision floating point value represented by a binary32 floating point value as defined in IEEE Std 754-2008, with the least significant bit of the fraction occurring in bit 0 of the field, in meters. If the Distance field value is unknown the field is set to 0.

The Relative Height field, defined by a 2-octet signed integer, specifies the relative height in meters that the neighbor is positioned, relative to the reporting BSS. If the Relative height is unknown or at the same height as the reporting BSS, the field is 0.

The Vendor Specific subelement has the same format as the Vendor Specific element (see 8.4.2.28). Multiple Vendor Specific subelements are optionally present in the list of optional subelements.

8.4.2.40 RCPI element

The RCPI element indicates the received frame power level at the receiving STA as shown in Figure 8-222.

	Element ID	Length	RCPI
Octets:	1	1	1

Figure 8-222—RCPI element format

The Element ID field is equal to the RCPI value in Table 8-54.

The Length field in octets is set to 1.

The RCPI field contains an RCPI value as specified for certain PHYs in Clause 16, Clause 18, Clause 17, Clause 19, and Clause 20.

8.4.2.41 BSS Average Access Delay element

The BSS Average Access Delay element contains the AP Average Access Delay, which is a measure of load in the BSS and is available in both QoS APs and non-QoS APs. The format of the BSS Average Access Delay element is defined in Figure 8-223.

	Element ID	Length	AP Average Access Delay
Octets:	1	1	1

Figure 8-223—BSS Average Access Delay element format

The Element ID field is equal to the BSS Average Access Delay value in Table 8-54.

The Length field in octets is set to 1.

The AP Average Access Delay is a scalar indication of the relative level of loading at an AP. A low value indicates more available capacity than a higher value. If the AP is not currently transmitting any DCF or EDCAF traffic, the AP Average Access Delay is set to 255. The values between 1 and 252 are a scaled representation of the average medium access delay for DCF or EDCAF transmitted frames measured from the time the DCF or EDCAF MPDU is ready for transmission (i.e., begins CSMA/CA access) until the actual frame transmission start time. Non-QoS APs average the access delays for all DCF transmitted frames. QoS APs average the access delays for all EDCA transmitted frames of all ACs. The AP Average Access Delay values are scaled as follows:

0:	Access Delay < 8 µs
1:	8 µs ≤ Access Delay < 16 µs
$2 \leq n \leq 14$:	$n \times 8$ µs ≤ Access Delay < $(n + 1) \times 8$ µs
15:	120 µs ≤ Access Delay < 128 µs
16:	128 µs ≤ Access Delay < 144 µs
$17 \leq n \leq 106$:	$(n \times 16) - 128$ µs ≤ Access Delay < $((n + 1) \times 16) - 128$ µs
107:	1584 µs ≤ Access Delay < 1600 µs
108:	1600 µs ≤ Access Delay < 1632 µs
$109 \leq n \leq 246$:	$(n \times 32) - 1856$ µs ≤ Access Delay < $((n + 1) \times 32) - 1856$ µs

247:	6048 µs ≤ Access Delay < 6080 µs
248:	6080 µs ≤ Access Delay < 8192 µs
249:	8192 µs ≤ Access Delay < 12288 µs
250:	12 288 µs ≤ Access Delay < 16384 µs
251:	16 384 µs ≤ Access Delay < 20480 µs
252:	20 480 µs ≤ Access Delay < 24576 µs
253:	24 576 µs ≤ Access Delay
254:	Service unable to access channel
255:	Measurement not available

The values 0–253 indicate Average Access Delay when one or more frames were transmitted during the measurement window. The value 254 indicates that DCF or EDCAF services are currently unable to access the channel due to continuous carrier sense mechanism deferral and that no frames were transmitted during the measurement window. The AP measures and averages the medium access delay for all transmit frames using the DCF or EDCAF over a continuous 30 s measurement window. See 10.11.16 for accuracy requirements.

8.4.2.42 Antenna element

The Antenna element contains the Antenna ID field as shown in Figure 8-224.

	Element ID	Length	Antenna ID
Octets:	1	1	1

Figure 8-224—Antenna element format

The Element ID field is equal to the Antenna value in Table 8-54.

The Length field in octets is set to 1.

The Antenna ID field contains the identifying number for the relevant antenna(s). When included in a Beacon, Probe Response, or Location Track Notification frame, the Antenna ID identifies the antenna(s) used to transmit the Beacon, Probe Response, or Location Track Notification frame. When included in a measurement report, or Location Track Notification frame, the Antenna ID identifies the antenna(s) used for the reported measurement or transmission of the Location Track Notification frame. The valid range for the Antenna ID is 1 to 254. The value 0 indicates that the antenna identifier is unknown. The value 255 indicates that this measurement was made with multiple antennas, i.e., antennas were switched during the measurement duration. In a Beacon Report or Frame Report, the Antenna ID always identifies the antenna used to receive the reported beacon or frame. If during frame reception, different antennas are used to receive the preamble and body, the antenna ID identifies the antenna that receives the frame body. In these cases, the value 255 is not used. The value 1 is the only value used for a STA with only one antenna. STAs with more than one antenna assign Antenna IDs to each antenna and each antenna configuration as consecutive, positive integers starting with 1. Each Antenna ID number represents a unique antenna or unique configuration of multiple antennas characterized by a fixed relative position, a fixed relative direction, and a fixed peak gain for that position and direction.

8.4.2.43 RSNI element

The RSNI element contains an RSNI value, as shown in Figure 8-225.

	Element ID	Length	RSNI
Octets:	1	1	1

Figure 8-225—RSNI element format

The Element ID field is equal to the RSNI value in Table 8-54.

The Length field in octets is set to 1.

RSNI is in steps of 0.5 dB. RSNI is calculated by the ratio of the received signal power (RCPI-ANPI) to the noise plus interference power (ANPI) using the expression:

$$\text{RSNI} = (10 \times \log10((\text{RCPI}_{\text{power}} - \text{ANPI}_{\text{power}}) / \text{ANPI}_{\text{power}})+10) \times 2$$

where $\text{RCPI}_{\text{power}}$ and $\text{ANPI}_{\text{power}}$ indicate power domain values for RCPI and ANPI and not dB domain values. RSNI in dB is scaled in steps of 0.5 dB to obtain 8-bit RSNI values, which cover the range from –10 dB to +117 dB. The value 255 indicates that RSNI is not available. See 10.11.9.4 for details and procedures for measuring ANPI.

8.4.2.44 Measurement Pilot Transmission element

The Measurement Pilot Transmission element contains a Measurement Pilot Transmission field as shown in Figure 8-226.

	Element ID	Length	Measurement Pilot Transmission	Optional Subelements
Octets:	1	1	1	variable

Figure 8-226—Measurement Pilot Transmission element format

The Element ID field is equal to the Measurement Pilot Transmission value in Table 8-54.

The Length field in octets is variable and depends on the number and length of optional subelements.

The Measurement Pilot Transmission field contains the Measurement Pilot Interval as specified in 8.4.1.18.

The Optional Subelements field format contains zero or more subelements, each consisting of a 1-octet Subelement ID field, a 1-octet Length field, and a variable-length Data field, as shown in Figure 8-402. Any optional subelements are ordered by nondecreasing subelement ID.

The Subelement ID field values for the defined optional subelements are shown in Table 8-117. A Yes in the Extensible column of a subelement listed in Table 8-117 indicates that the Length of the subelement might be extended in future revisions or amendments of this standard. When the Extensible column of an element is equal to Subelements, then the subelement might be extended in future revisions or amendments of this standard by defining additional subelements within the subelement. See 9.24.9.

Table 8-117—Optional subelement IDs for Measurement Pilot Transmission

Subelement ID	Name	Length field (octets)	Extensible
0–220	Reserved		
221	Vendor Specific	1 to 255	
222–255	Reserved		

The Vendor Specific subelements have the same format as their corresponding elements (see 8.4.2.28). Multiple Vendor Specific subelements are optionally present in the list of optional subelements.

8.4.2.45 BSS Available Admission Capacity element

The BSS Available Admission Capacity element contains a list of Available Admission Capacity fields at different User Priorities and Access Categories as shown in Figure 8-227.

NOTE—The BSS Available Admission Capacity element is helpful for roaming QoS STAs to select a QoS AP that is likely to accept future admission control requests, but it does not provide an assurance that the HC will admit these requests.

	Element ID	Length	Available Admission Capacity Bitmask	Available Admission Capacity List
Octets:	1	1	2	2 x (total number of nonzero bits in Available Admission Capacity Bitmask)

Figure 8-227—BSS Available Admission Capacity element format

The Element ID field is equal to the BSS Available Admission Capacity value in Table 8-54.

The Length field has units of octets and is set to $2 + 2N_{nz}$, where N_{nz} equals the total number of nonzero bits in Available Admission Capacity Bitmask.

The Available Admission Capacity Bitmask field indicates the UP values that have an Available Admission Capacity specified in the Available Admission Capacity List field. The format of the Available Admission Capacity Bitmask is defined in Table 8-118.

Table 8-118—Available Admission Capacity Bitmask definition

Bit	Available Admission Capacity Reported
0	UP 0
1	UP 1
2	UP 2
3	UP 3
4	UP 4
5	UP 5

Table 8-118—Available Admission Capacity Bitmask definition *(continued)*

Bit	Available Admission Capacity Reported
6	UP 6
7	UP 7
8	AC 0
9	AC 1
10	AC 2
11	AC 3
12–15	Reserved

Each bit in the Available Admission Capacity Bitmask is set to 1 to indicate that the Available Admission Capacity for the corresponding traffic type is present in the Available Admission Capacity List field. The bit is set to 0 to indicate that the Available Admission Capacity for the corresponding traffic type is not present in the Available Admission Capacity List field.

The Available Admission Capacity List comprises a sequence of Available Admission Capacity fields corresponding respectively to the nonzero bits in the Available Admission Capacity Bitmask field.

The Available Admission Capacity field is 2 octets long and contains an unsigned integer that specifies the amount of medium time available using explicit admission control for the corresponding UP or AC traffic, in units of 32 µs per 1 s. See 10.11.17 for furthers details.

8.4.2.46 BSS AC Access Delay element

The BSS AC Access Delay element contains an Access Category Access Delay field, as shown in Figure 8-228.

	Element ID	Length	Access Category Access Delay
Octets:	1	1	4

Figure 8-228—BSS AC Access Delay element format

The Element ID field is equal to the BSS AC Access Delay value in Table 8-54.

The Length field in octets is set to 4.

The AC Access Delay field is formatted as four subfields as shown in Figure 8-229. The AC Access Delay is a scalar indication of the average access delay at a QoS AP for services for each of the indicated Access Categories. If the QoS AP is not currently transmitting any traffic using the indicated AC, the Average Access Delay for that AC is set to 255. The values between 1 and 252 are a scaled representation of the average medium access delay for transmitted frames in the indicated AC measured from the time the EDCA MPDU is ready for transmission (i.e., begins CSMA/CA access) until the actual frame transmission start time. The AC Access Delay values are scaled as follows:

0:	$\text{Access Delay} < 8\ \mu\text{s}$
1:	$8\ \mu\text{s} \le \text{Access Delay} < 16\ \mu\text{s}$
$2 \le n \le 14$:	$n \times 8\ \mu\text{s} \le \text{Access Delay} < (n + 1) \times 8\ \mu\text{s}$
15:	$120\ \mu\text{s} \le \text{Access Delay} < 128\ \mu\text{s}$
16:	$128\ \mu\text{s} \le \text{Access Delay} < 144\ \mu\text{s}$
$17 \le n \le 106$:	$(n \times 16) - 128\ \mu\text{s} \le \text{Access Delay} < ((n + 1) \times 16) - 128\ \mu\text{s}$
107:	$1584\ \mu\text{s} \le \text{Access Delay} < 1600\ \mu\text{s}$
108:	$1600\ \mu\text{s} \le \text{Access Delay} < 1632\ \mu\text{s}$
$109 \le n \le 246$:	$(n \times 32) - 1856\ \mu\text{s} \le \text{Access Delay} < ((n + 1) \times 32) - 1856\ \mu\text{s}$
247:	$6048\ \mu\text{s} \le \text{Access Delay} < 6080\ \mu\text{s}$
248:	$6080\ \mu\text{s} \le \text{Access Delay} < 8192\ \mu\text{s}$
249:	$8192\ \mu\text{s} \le \text{Access Delay} < 12\,288\ \mu\text{s}$
250:	$12\,288\ \mu\text{s} \le \text{Access Delay} < 16\,384\ \mu\text{s}$
251:	$16\,384\ \mu\text{s} \le \text{Access Delay} < 20\,480\ \mu\text{s}$
252:	$20\,480\ \mu\text{s} \le \text{Access Delay} < 24\,576\ \mu\text{s}$
253:	$24\,576\ \mu\text{s} \le \text{Access Delay}$
254:	Service unable to access channel
255:	Measurement not available

The values 0–253 indicate Average Access Delay when one or more frames were transmitted during the measurement window. The value 254 indicates that EDCA services are currently unable to access the channel due to continuous carrier sense mechanism deferral to higher priority AC transmissions and that no frames were transmitted during the measurement window. The QoS AP measures and averages the medium access delay for all transmit frames of the indicated AC using EDCA mechanism over a continuous 30 s measurement window. See 10.11.16 for accuracy requirements.

	Average Access Delay for Best Effort (AC_BE)	Average Access Delay for Background (AC_BK)	Average Access Delay for Video (AC_VI)	Average Access Delay for Voice (AC_VO)
Octets:	1	1	1	1

Figure 8-229—Access Category Access Delay subfields

8.4.2.47 RM Enabled Capabilities element

The RM Enabled Capabilities element signals support for radio measurements in a device. The element is shown in Figure 8-230.

	Element ID	Length	RM Enabled Capabilities
Octets:	1	1	5

Figure 8-230—RM Enabled Capabilities element format

The Element ID field is equal to the RM Enabled Capabilities value in Table 8-54.

The Length field in octets is set to 5.

The RM Enabled Capabilities field is an octet string. Each subfield of this field indicates whether the corresponding capability listed in Table 8-119 is enabled.

Table 8-119—RM Enabled Capabilities definition

Bit position in the RM Enabled Capabilities field	Field name	Notes
0	Link Measurement capability enabled	A STA sets Link Measurement capability enabled bit to 1 when dot11RMLinkMeasurementActivated is true, and is set to 0 otherwise.
1	Neighbor Report capability enabled	A STA sets Neighbor Report capability enabled bit to 1 when dot11RMNeighborReportActivated is true, and sets it to 0 otherwise.
2	Parallel Measurements capability enabled	A STA sets Parallel Measurements capability enabled bit to 1 when dot11RMParallelMeasurementsActivated is true, and sets it to 0 otherwise.
3	Repeated Measurements capability enabled	A STA sets Repeated Measurements capability enabled bit to 1 when dot11RMRepeatedMeasurementsActivated is true, and sets it to 0 otherwise.
4	Beacon Passive Measurement capability enabled	A STA sets Beacon Passive Measurement capability enabled bit to 1 when dot11RMBeaconPassiveMeasurementActivated is true, and sets it to 0 otherwise.
5	Beacon Active Measurement capability enabled	A STA sets Beacon Active Measurement capability enabled bit to 1 when dot11RMBeaconActiveMeasurementActivated is true, and sets it to 0 otherwise.
6	Beacon Table Measurement capability enabled	A STA sets Beacon Table Measurement capability enabled bit to 1 when dot11RMBeaconTableMeasurementActivated is true, and sets it to 0 otherwise.
7	Beacon Measurement Reporting Conditions capability enabled	A STA sets Beacon Measurement Reporting Conditions capability enabled bit to 1 when dot11RMBeaconMeasurementReportingConditionsActivated is true, and sets it to 0 otherwise.
8	Frame Measurement capability enabled	A STA sets Frame Measurement capability enabled bit to 1 when dot11RMFrameMeasurementActivated is true, and sets it to 0 otherwise.

Table 8-119—RM Enabled Capabilities definition *(continued)*

Bit position in the RM Enabled Capabilities field	Field name	Notes
9	Channel Load Measurement capability enabled	A STA sets Channel Load Measurement capability enabled bit to 1 when dot11RMChannelLoadMeasurementActivated is true, and sets it to 0 otherwise.
10	Noise Histogram Measurement capability enabled	A STA sets Noise Histogram Measurement capability enabled bit to 1 when dot11RMNoiseHistogramMeasurementActivated is true, and sets it to 0 otherwise.
11	Statistics Measurement capability enabled	A STA sets Statistics Measurement capability enabled bit to 1 when dot11RMStatisticsMeasurementActivated is true, and sets it to 0 otherwise.
12	LCI Measurement capability enabled	A STA sets LCI Measurement capability enabled bit to 1 when dot11RMLCIMeasurementActivated is true, and sets it to 0 otherwise.
13	LCI Azimuth capability enabled	A STA sets LCI Azimuth capability enabled bit to 1 when dot11RMLCIAzimuthActivated is true, and sets it to 0 otherwise.
14	Transmit Stream/ Category Measurement capability enabled	A STA sets Transmit Stream/Category Measurement capability enabled bit to 1 when dot11RMTransmitStreamCategoryMeasurementActivated is true, and sets it to 0 otherwise.
15	Triggered Transmit Stream/Category Measurement capability enabled	A STA sets Triggered Transmit Stream/Category Measurement capability enabled bit to 1 when dot11RMTriggeredTransmitStreamCategoryMeasurementActivated is true, and sets it to 0 otherwise.
16	AP Channel Report capability enabled	A STA sets AP Channel Report capability enabled bit to 1 when dot11RMAPChannelReportActivated is true, and sets it to 0 otherwise.
17	RM MIB capability enabled	A STA sets RM MIB capability enabled bit to 1 when dot11RMMIBActivated is true, and sets it to 0 otherwise.
18–20	Operating Channel Max Measurement Duration	A STA sets Operating Channel Max Measurement Duration to equal the value of dot11RMMaxMeasurementDuration. See 10.11.4.
21–23	Nonoperating Channel Max Measurement Duration	A STA sets Nonoperating Channel Max Measurement Duration to equal the value of dot11RMNonOperatingChannelMaxMeasurementDuration. See 10.11.4.
24–26	Measurement Pilot capability	A STA sets Measurement Pilot capability to equal the value of dot11RMMeasurementPilotActivated. See Table 10-7 in 10.11.15.
27	Measurement Pilot Transmission Information capability enabled	A STA sets Measurement Pilot Transmission capability enabled bit to 1 when dot11RMMeasurementPilotTransmissionInformationActivated is true, and sets it to 0 otherwise.
28	Neighbor Report TSF Offset capability enabled	A STA sets Neighbor Report TSF Offset capability enabled bit to 1 when dot11RMNeighborReportTSFOffsetActivated is true, and sets it to 0 otherwise.
29	RCPI Measurement capability enabled	A STA sets RCPI Measurement capability enabled bit to 1 when dot11RMRCPIMeasurementActivated is true, and sets it to 0 otherwise.

Table 8-119—RM Enabled Capabilities definition *(continued)*

Bit position in the RM Enabled Capabilities field	Field name	Notes
30	RSNI Measurement capability enabled	A STA sets RSNI Measurement capability enabled bit to 1 when dot11RMRSNIMeasurementActivated is true, and sets it to 0 otherwise.
31	BSS Average Access Delay capability enabled	A STA sets BSS Average Access Delay capability enabled bit to 1 when dot11RMBSSAverageAccessDelayActivated is true, and sets it to 0 otherwise (see NOTE).
32	BSS Available Admission Capacity capability enabled	A STA sets BSS Available Admission Capacity capability enabled bit to 1 when dot11RMBSSAvailableAdmissionCapacityActivated is true, and sets it to 0 otherwise.
33	Antenna capability enabled	A STA sets Antenna capability enabled bit to 1 when dot11RMAntennaInformationActivated is true, and sets it to 0 otherwise.
34–39	Reserved	
NOTE—At a QoS AP dot11RMBSSAverageAccessDelayActivated is true indicates that the AP BSS AC Access Delay capability is also enabled.		

8.4.2.48 Multiple BSSID element

The format of the Multiple BSSID element is shown in Figure 8-231.

	Element ID	Length	MaxBSSID Indicator	Optional Subelements
Octets:	1	1	1	variable

Figure 8-231—Multiple BSSID element format

The Element ID field is equal to the Multiple BSSID value in Table 8-54.

The value of the Length field is 1 plus the length of the extensions in units of octets.

The Max BSSID Indicator field contains a value assigned to *n*, where 2^n is the maximum number of BSSIDs in the multiple BSSID set, including the reference BSSID (see 10.11.14). The actual number of BSSIDs in the multiple BSSID set is not explicitly signalled. The BSSID(i) value corresponding to the i^{th} BSSID in the multiple BSSID set is derived from a reference BSSID (REF_BSSID) as follows:

BSSID(i) = BSSID_A | BSSID_B

where

BSSID_A is a BSSID with (48–n) MSBs equal to the (48–n) MSBs of the REF_BSSID and n LSBs equal to 0

BSSID_B is a BSSID with (48–n) MSBs equal to 0 and n LSBs equal to [(n LSBs of REF_BSSID) +i] mod 2^n

| indicates the OR operation

When the Multiple BSSID element is transmitted in a Beacon or Probe Response frame, the reference BSSID is the BSSID of the frame. More than one Multiple BSSID element may be included in a Beacon frame. The AP determines the number of Multiple BSSID elements. The AP does not fragment a nontransmitted BSSID profile subelement for a single BSSID across two Multiple BSSID elements unless the length of the nontransmitted BSSID profile subelement exceeds 255 octets. When the Multiple BSSID element is transmitted as a subelement in a Neighbor Report element, the reference BSSID is the BSSID field in the Neighbor Report element.

The Optional Subelements field format contains zero or more subelements, each consisting of a 1-octet Subelement ID field, a 1-octet Length field, and a variable-length Data field, as shown in Figure 8-402. Any optional subelements are ordered by nondecreasing subelement ID.

The Subelement ID field values for the defined optional subelements are shown in Table 8-120. A Yes in the Extensible column of a subelement listed in Table 8-120 indicates that the Length of the subelement might be extended in future revisions or amendments of this standard. When the Extensible column of an element is equal to Subelements, then the subelement might be extended in future revisions or amendments of this standard by defining additional subelements within the subelement. See 9.24.9.

Table 8-120—Optional subelement IDs for Multiple BSSID

Subelement ID	Name	Length field (octets)	Extensible
0	Nontransmitted BSSID Profile	Variable	
1–220	Reserved		
221	Vendor Specific	1 to 255	
222–255	Reserved		

The Nontransmitted BSSID Profile subelement contains a list of elements for one or more APs that have nontransmitted BSSIDs, and is defined as follows:

- For each nontransmitted BSSID, the Nontransmitted BSSID Capability element (see 8.4.2.74) is the first element included, followed by a variable number of elements, in the order defined in 8-20.
- The SSID and multiple BSSID-index subelements are included in the Nontransmitted BSSID Profile subelement.
- The FMS Descriptor element is included in the Nontransmitted BSSID Profile subelement if the Multiple BSSID element is included in a Beacon frame and if the TIM field indicates there are buffered group addressed frames for this nontransmitted BSSID.
- The Timestamp and Beacon Interval fields, DSSS Parameter Set, FH Parameter Set, IBSS Parameter Set, Country, FH Parameters, FH Pattern Table, Channel Switch Assignment, Extended Channel Switch Announcement, Supported Operating Classes, IBSS DFS, ERP Information, HT Capabilities and HT Operation elements are not included in the Nontransmitted BSSID Profile field; the values of these elements for each nontransmitted BSSID are always the same as the corresponding transmitted BSSID element values.

The Vendor Specific subelements have the same format as their corresponding elements (see 8.4.2.28). Multiple Vendor Specific subelements are optionally present in the list of optional subelements.

The Multiple BSSID element is included in Beacon frames, as described in 8.3.3.2, and Probe Response frames, as described in 8.3.3.10. The use of the Multiple BSSID element is described 10.11.14. Nontransmitted BSSID Advertisement procedures are described in 10.1.3.6.

8.4.2.49 Mobility Domain element (MDE)

The MDE contains the Mobility Domain Identifier (MDID) and the FT Capability and Policy field. The AP uses the MDE to advertise that it is included in the group of APs that constitute a mobility domain, to advertise its support for FT capability, and to advertise its FT policy information. The format for this element is given in Figure 8-232.

	Element ID	Length	MDID	FT Capability and Policy
Octets:	1	1	2	1

Figure 8-232—MDE format

The Length field is set to 3.

The MDID field is a 2-octet value that follows the ordering conventions defined in 8.2.2.

The FT Capability and Policy field is 1 octet. The FT Capability and Policy field is defined in Figure 8-233.

	B0	B1	B2 B7
	Fast BSS Transition over DS	Resource Request Protocol Capability	Reserved
Bits:	1	1	6

Figure 8-233—FT Capability and Policy field

Bits 0–1 of the FT Capability and Policy field control the behavior of STAs performing fast BSS transitions (see 12.3). The STA might use information from the MDE to determine the transition methods recommended by the AP and protocols supported by the AP. The choice of executing any specific transition method is outside the scope of this standard.

If Resource Request Protocol Capability subfield is 1, then the STA may perform the FT Resource Request Protocol of 12.6.

When sent by a STA to a target AP, the FT Capability and Policy field matches the value advertised by that target AP. See 12.8.

8.4.2.50 Fast BSS Transition element (FTE)

The FTE includes information needed to perform the FT authentication sequence during a fast BSS transition in an RSN. This element is shown in Figure 8-234.

	Element ID	Length	MIC Control	MIC	ANonce	SNonce	Optional Parameter(s)
Octets:	1	1	2	16	32	32	variable

Figure 8-234—FTE format

The Length field for this element indicates the length of the Information field.

The MIC Control field is two octets and is defined in Figure 8-235.

	B0 B7	B8 B15
	Reserved	Element Count
Bits:	8	8

Figure 8-235—MIC Control field

The Element Count subfield of the MIC Control field contains the number of elements that are included in the message integrity code (MIC) calculation. A value of 0 indicates no MIC is present.

The MIC field contains a MIC that is calculated using the algorithm specified in 12.8.4 and 12.8.5.

The ANonce field contains a value chosen by the R1KH. It is encoded following the conventions in 8.2.2.

The SNonce field contains a value chosen by the S1KH. It is encoded following the conventions in 8.2.2.

The format of the Optional Parameter(s) field is shown in Figure 8-236.

	Subelement ID	Length	Data
Octets:	1	1	variable

Figure 8-236—Optional Parameter(s) field

The Subelement ID field contains one of the values from Table 8-121:

Table 8-121—Subelement IDs

Value	Contents of Data field	Length (in octets)
0	Reserved	
1	PMK-R1 key holder identifier (R1KH-ID)	6
2	GTK subelement	35–51
3	PMK-R0 key holder identifier (R0KH-ID)	1–48
4	IGTK	Variable
5–255	Reserved	

R1KH-ID indicates the identity of the R1KH, which is used by the S0KH and the R0KH for deriving the PMK-R1s. It is encoded following the conventions in 8.2.2.

The GTK subelement contains the group temporal key, which is encrypted (see procedures in 12.8.5) and is defined in Figure 8-237.

	Subelement ID	Length	Key Info	Key Length	RSC	Wrapped Key
Octets:	1	1	2	1	8	24–40

Figure 8-237—GTK subelement format

The GTK subelement Key Info subfield is defined in Figure 8-238.

	B0 B1	B2 B15
	Key ID	Reserved
Bits:	2	14

Figure 8-238—GTK subelement's Key Info subfield

Key Length field is the length of the Key field in octets, not including any padding (see 12.8.5).

RSC field contains the receive sequence counter (RSC) for the GTK being installed. Delivery of the RSC field value allows a STA to identify replayed MPDUs. If the RSC field value is less than 8 octets in length, the remaining octets are set to 0. The least significant octet of the transmit sequence counter (TSC) or packet number (PN) is in the first octet of the RSC field. See Table 11-5.

For WEP, the RSC value is set to 0 on transmit and is not used at the receiver.

The Wrapped Key field contains the encrypted GTK as described in 12.8.5.

When sent by a non-AP STA, the R0KH-ID indicates the R0KH with which the S0KH negotiated the PMK-R0 it is using for this transition. When sent by an AP, the R0KH-ID indicates the R0KH that the S0KH will be using to generate a PMK-R0 security association. It is encoded following the conventions from 8.2.2.

The IGTK field contains the Integrity GTK, used for protecting robust management frames. The IGTK subelement format is shown in Figure 8-239.

	Subelement ID	Length	KeyID	IPN	Key Length	Wrapped Key
Octets:	1	1	2	6	1	24

Figure 8-239—IGTK subelement format

The KeyID field indicates the value of the BIP key ID.

The IPN field indicates the receive sequence counter for the IGTK being installed. The PN field gives the current message number for the IGTK, to allow a STA to identify replayed MPDUs.

The Key Length field is the length of IGTK in octets, not including any padding (see 12.8.5).

The Wrapped Key field contains the wrapped IGTK being distributed. The length of the resulting AES-Key-wrapped IGTK in the Wrapped Key field is Key Length + 8 octets.

8.4.2.51 Timeout Interval element (TIE)

The TIE specifies time intervals and timeouts. Figure 8-240 shows this element.

	Element ID	Length	Timeout Interval Type	Timeout Interval Value
Octets:	1	1	1	4

Figure 8-240—TIE format

The Length field is set to 5.

The Timeout Interval Type field contains one of the values from Table 8-122.

Table 8-122—Timeout Interval Type field value

Timeout Interval Type	Meaning	Units
0	Reserved	
1	Reassociation deadline interval	Time units (TUs)
2	Key lifetime interval	Seconds
3	Association Comeback time	Time units (TUs)
4–255	Reserved	

The Timeout Interval Value field contains an unsigned 32-bit integer. It is encoded according to the conventions in 8.2.2.

A reassociation deadline interval value of 0 indicates no deadline exists. A key lifetime interval value of 0 is reserved.

8.4.2.52 RIC Data element (RDE)

The RIC refers to a collection of elements that are used to express a resource request and to convey responses to the corresponding requests.

A RIC is a sequence of one or more Resource Requests, or a sequence of one or more Resource Responses. Each Resource Request or Response consists of an RDE, followed by one or more elements that describe that resource. See 12.11 for examples and procedures.

The RDE format is shown in Figure 8-241.

	Element ID	Length	RDE Identifier	Resource Descriptor Count	Status Code
Octets:	1	1	1	1	2

Figure 8-241—RDE format

The Length field is set to 4.

The RDE Identifier field has an arbitrary 8-bit value, chosen by the resource requestor to uniquely identify the RDE within the RIC.

The Resource Descriptor Count field indicates the number of alternative Resource Descriptors that follow this RDE.

The Status Code field is used in Resource Responses to indicate the result of the request. Valid values for the Status Code field are given in 8.4.1.9. When an RDE is included in a Resource Request, the Status Code field is reserved.

8.4.2.53 RIC Descriptor element

The RIC Descriptor element is used with an RDE during a fast BSS transition to negotiate resources that are not otherwise described by elements. See 12.11 for procedures for including this element in a RIC.

Figure 8-242 shows the format of this element.

	Element ID	Length	Resource Type	Variable parameters
Octets:	1	1	1	variable

Figure 8-242—RIC Descriptor element format

The Length field is set to the number of octets in this element (variable).

The Resource Type field contains one of the values given in Table 8-123.

Table 8-123—Resource type code in RIC Descriptor element

Resource type value	Meaning	Variable parameters
1	Block Ack	Block Ack parameter set as defined in 8.4.1.14, Block Ack timeout value as defined in 8.4.1.15, and Block Ack starting sequence control as defined in 8.3.1.8.
0, 2–255	Reserved	

Variable parameters contain any additional data based on the resource type.

8.4.2.54 DSE Registered Location element

A DSE Registered Location element includes DSE location configuration information (LCI), which contains latitude, longitude, and altitude information. The DSE Registered Location element format is shown in Figure 8-243.

	Element ID	Length	DSE Registered Location element body fields
Octets:	1	1	20

Figure 8-243—DSE Registered Location element format

The Length field is set to 20.

The structure and information fields are little endian, per conventions defined in 8.2.2, and are based on the LCI format described in IETF RFC 3825.

The DSE Registered Location element body fields are shown in Figure 8-244.

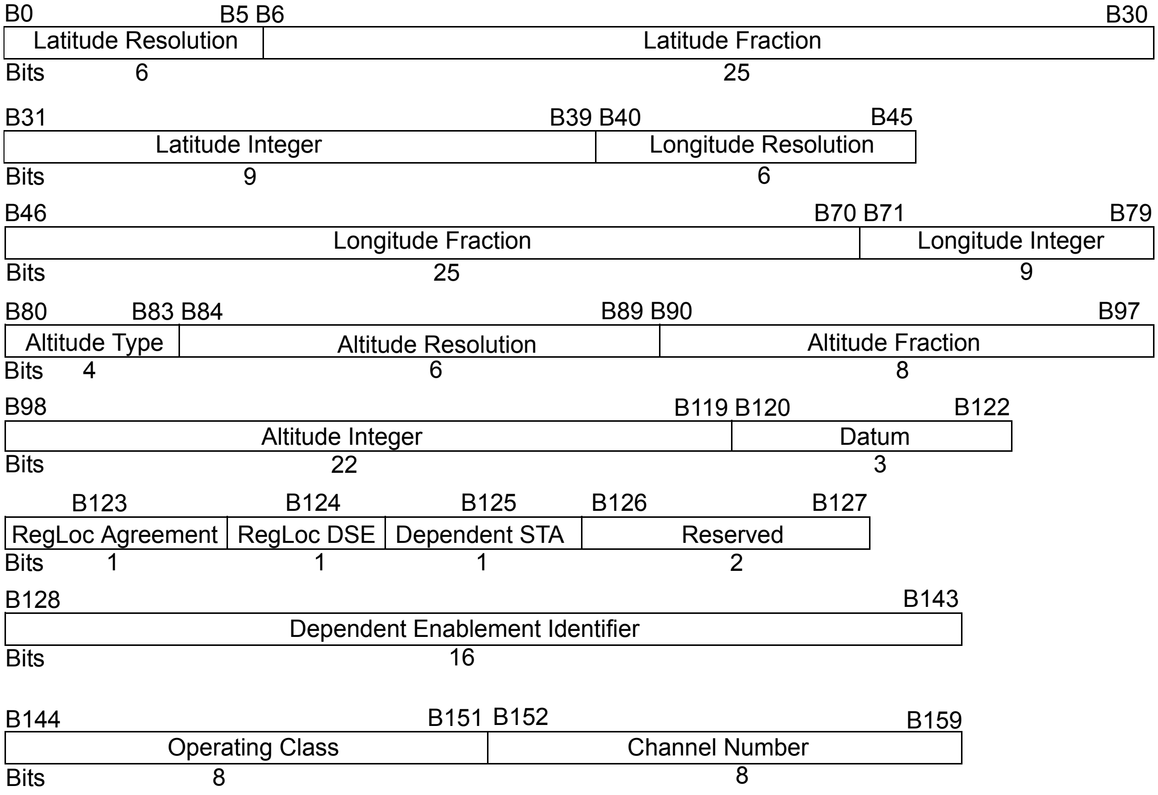


Figure 8-244—DSE registered location element body fields format

The definition of fields within the DSE Registered Location element body is as defined in Section 2.1 of IETF RFC 3825 (July 2004) except as defined in this standard.

With an Altitude Type field value of 3 (i.e., height above ground is in meters), the altitude is defined to be in meters and is formatted in twos-complement, fixed-point, 22-bit integer part with 8-bit fraction.

The Datum field is a 3-bit field, rather than the 8-bit field defined in IETF RFC 3825, and the codes used are as defined in IETF RFC 3825.

The RegLoc Agreement bit field is set to 1 to report that the STA is operating within a national policy area or an international agreement area near a national border (see 10.12.3); otherwise, it is 0.

The RegLoc DSE bit field is set to 1 to report that the enabling STA is enabling the operation of STAs with DSE; otherwise, it is 0.

The Dependent STA bit field is set to 1 to report that the STA is operating with the enablement of the enabling STA whose LCI is being reported; otherwise, it is 0.

The Dependent Enablement Identifier field is a 16-bit field with a value set by the enabling STA via the DSE Enablement frame; otherwise, it is set to 0.

The Operating Class field indicates the channel set for which the enablement request, report, or announcement applies. The Operating Class and Channel Number fields together specify the channel frequency and channel bandwidth for which the report applies. Valid values for the Operating Class field are shown in Annex E.

The Channel Number field indicates the channel number for which the enablement request, report, or announcement applies. The channel number is defined within an operating class as shown in Annex E.

NOTE—An example of fixed/fractional notation, using the longitude of the Sears Tower from p. 13 of IETF RFC 3825 (July 2004) is shown below:

Longitude 87.63602 degrees West (or –87.63602 degrees),

Using twos-complement, 34-bit fixed point, 25-bit fraction,

Longitude = 0xf50ba5b97,

Longitude = 1101010000101110100101101110010111 (big endian)

DSE registered location expression for a Longitude resolution of 34 bits:

Bits 40–45 Longitude resolution = (bit 40) 0 1 0 0 0 1 (bit 45)

Bits 46–70 Longitude fraction = (bit 46) 1 1 1 0 1 0 0 1 1 1 0 1 1 0 1 0 0 1 0 1 1 1 0 1 0 (bit 70)

Bits 71–79 Longitude integer = (bit 71) 0 0 0 1 0 1 0 1 1 (bit 79)

The octets in transmission order = E2 E5 96 2E D4.

8.4.2.55 Extended Channel Switch Announcement element

The Extended Channel Switch Announcement element is used by an AP in an infrastructure BSS, a STA in an IBSS, or a mesh STA in an MBSS to advertise when the BSS is changing to a new channel or a new channel in a new operating class. The announcement includes both the operating class and the channel number of the new channel. The element is present only when an extended channel switch is pending. The format of the Extended Channel Switch Announcement element is shown in Figure 8-245.

	Element ID	Length	Channel Switch Mode	New Operating Class	New Channel Number	Channel Switch Count
Octets:	1	1	1	1	1	1

Figure 8-245—Extended Channel Switch Announcement element format

The Length field is set to 4.

The Channel Switch Mode field indicates any restrictions on transmission until a channel switch. An AP in an infrastructure BSS or a STA in an IBSS sets the Channel Switch Mode field to either 0 or 1 on transmission as specified in 10.9.8.2 and 10.9.8.3. The Channel Switch Mode field is reserved in an MBSS.

The New Operating Class field is set to the number of the operating class after the channel switch, as defined in Annex E.

The New Channel Number field is set to the number of the channel after the channel switch. The channel number is a channel from the STA's new operating class as defined in Annex E.

For nonmesh STAs, the Channel Switch Count field indicates either the number of target beacon transmission times (TBTTs) until the STA sending the Extended Channel Switch Announcement element switches to the new channel or a value of 0. A value of 1 indicates that the switch occurs immediately before the next TBTT. A value of 0 indicates that the switch occurs anytime after the frame containing the element is transmitted.

For mesh STAs, the Channel Switch Count field is encoded as an octet with bits 6 to 0 set to the time, in units of 2TU when the MSB (bit 7) is 0, or in units of 100TU when the MSB (bit 7) is 1, until the mesh STA sending the Channel Switch Announcement element switches to the new channel. A value of 0 for bits 6 to 0 indicates that the switch occurs at any time after the frame containing the element is transmitted. For example, a 200 TU channel switch time is encoded as X'82' and a 10TU channel switch time is encoded as X'05'.

8.4.2.56 Supported Operating Classes element

The Supported Operating Classes element is used by a STA to advertise the operating classes that it is capable of operating with in this country. The format of the Supported Operating Classes element is shown in Figure 8-246.

	Element ID	Length	Current Operating Class	Operating Classes
Octets:	1	1	1	Length–1

Figure 8-246—Supported Operating Classes element format

The value of the Length field of the Supported Operating Classes element is between 2 and 253. The Current Operating Class octet indicates the operating class in use for transmission and reception. The Operating Classes field lists in ascending order all operating classes that the STA is capable of operating with in this country. The use of this element is described in 10.10.1 and 10.11.9.1.

8.4.2.57 Management MIC element

The Management MIC element (MME) provides message integrity and protects group addressed robust management frames from forgery and replay. Figure 8-247 shows the MME format.

	Element ID	Length	KeyID	IPN	MIC
Octets:	1	1	2	6	8

Figure 8-247—Management MIC element format

The value of the Element ID field is 76 decimal (4c hex).

The Length field is set to 16.

The Key ID field identifies the IGTK used to compute the MIC. Bits 0–11 define a value in the range 0–4095. Bits 12–15 are reserved. The IGTK Key ID is either 4 or 5. The remaining Key IDs are reserved.

The IPN field contains a 6 octet value, interpreted as a 48-bit unsigned integer and used to detect replay of protected group addressed robust management frames.

The MIC field contains a message integrity code calculated over the robust management frame as specified in 11.4.4.5 and 11.4.4.6.

8.4.2.58 HT Capabilities element

8.4.2.58.1 HT Capabilities element structure

An HT STA declares that it is an HT STA by transmitting the HT Capabilities element.

The HT Capabilities element contains a number of fields that are used to advertise optional HT capabilities of an HT STA. The HT Capabilities element is present in Beacon, Association Request, Association Response, Reassociation Request, Reassociation Response, Probe Request, Probe Response, Mesh Peering Open, and Mesh Peering Close frames. The HT Capabilities element is defined in Figure 8-248.

	Element ID	Length	HT Capabilities Info	A-MPDU Parameters	Supported MCS Set	HT Extended Capabilities	Transmit Beamforming Capabilities	ASEL Capabilities
Octets:	1	1	2	1	16	2	4	1

Figure 8-248—HT Capabilities element format

The Element ID field is set to the value for HT Capabilities element defined in Table 8-54.

The Length field of the HT Capabilities element is set to 26.

8.4.2.58.2 HT Capabilities Info field

The HT Capabilities Info field of the HT Capabilities element is 2 octets in length, and contains capability information bits. The structure of this field is defined in Figure 8-249.

	B0	B1	B2 B3	B4	B5	B6	B7	B8 B9
	LDPC Coding Capability	Supported Channel Width Set	SM Power Save	HT-Greenfield	Short GI for 20 MHz	Short GI for 40 MHz	Tx STBC	Rx STBC
Bits:	1	1	2	1	1	1	1	2

B10	B11	B12	B13	B14	B15
HT-Delayed Block Ack	Maximum A-MSDU Length	DSSS/CCK Mode in 40 MHz	Reserved	Forty MHz Intolerant	L-SIG TXOP Protection Support
1	1	1	1	1	1

Figure 8-249—HT Capabilities Info field

The subfields of the HT Capabilities Info field are defined in Table 8-124.

Table 8-124—Subfields of the HT Capabilities Info field

Subfield	Definition	Encoding
LDPC Coding Capability	Indicates support for receiving LDPC coded packets	Set to 0 if not supported Set to 1 if supported
Supported Channel Width Set	Indicates the channel widths supported by the STA. See 10.15.	Set to 0 if only 20 MHz operation is supported Set to 1 if both 20 MHz and 40 MHz operation is supported This field is reserved when the transmitting or receiving STA is operating in an operating class that does not include a value of 13 or 14 in the behavior limits as specified in Annex E.
SM Power Save	Indicates the spatial multiplexing power save mode. See 10.2.4.	Set to 0 for static SM power save mode Set to 1 for dynamic SM power save mode Set to 3 for SM Power Save disabled The value 2 is reserved
HT-Greenfield	Indicates support for the reception of PPDUs with HT-greenfield format. See 20.1.4.	Set to 0 if not supported Set to 1 if supported
Short GI for 20 MHz	Indicates short GI support for the reception of packets transmitted with TXVECTOR parameter CH_BANDWIDTH equal to HT_CBW20	Set to 0 if not supported Set to 1 if supported
Short GI for 40 MHz	Indicates short GI support for the reception of packets transmitted with TXVECTOR parameter CH_BANDWIDTH equal to HT_CBW40	Set to 0 if not supported Set to 1 if supported
Tx STBC	Indicates support for the transmission of PPDUs using STBC	Set to 0 if not supported Set to 1 if supported
Rx STBC	Indicates support for the reception of PPDUs using STBC	Set to 0 for no support Set to 1 for support of one spatial stream Set to 2 for support of one and two spatial streams Set to 3 for support of one, two and three spatial streams
HT-Delayed Block Ack	Indicates support for HT-delayed Block Ack operation. See 9.21.8.	Set to 0 if not supported Set to 1 if supported Support indicates that the STA is able to accept an ADDBA request for HT-delayed Block Ack
Maximum A-MSDU Length	Indicates maximum A-MSDU length. See 9.11.	Set to 0 for 3839 octets Set to 1 for 7935 octets

Table 8-124—Subfields of the HT Capabilities Info field *(continued)*

Subfield	Definition	Encoding
DSSS/CCK Mode in 40 MHz	Indicates use of DSSS/CCK mode in a 20/40 MHz BSS. See 10.15.	In Beacon and Probe Response frames: Set to 0 if the BSS does not allow use of DSSS/CCK in 40 MHz Set to 1 if the BSS does allow use of DSSS/CCK in 40 MHz Otherwise: Set to 0 if the STA does not use DSSS/CCK in 40 MHz Set to 1 if the STA uses DSSS/CCK in 40 MHz See 10.15.8 for operating rules
Forty MHz Intolerant	Indicates whether APs receiving this information or reports of this information are required to prohibit 40 MHz transmissions (see 10.15.12).	Set to 1 by an HT STA to prohibit a receiving AP from operating that AP's BSS as a 20/40 MHz BSS; otherwise, set to 0.
L-SIG TXOP Protection Support	Indicates support for the L-SIG TXOP protection mechanism (see 9.23.5)	Set to 0 if not supported Set to 1 if supported

The following subfields are reserved for a mesh STA: Tx STBC, Rx STBC, PSMP Support.

8.4.2.58.3 A-MPDU Parameters field

The structure of the A-MPDU Parameters field of the HT Capabilities element is shown in Figure 8-250.

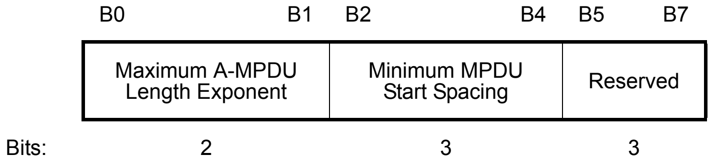


Figure 8-250—A-MPDU Parameters field

The subfields of the A-MPDU Parameters field are defined in Table 8-125.

Table 8-125—Subfields of the A-MPDU Parameters field

Subfield	Definition	Encoding
Maximum A-MPDU Length Exponent	Indicates the maximum length of A-MPDU that the STA can receive.	This field is an integer in the range 0 to 3. The length defined by this field is equal to $2^{(13 + \text{Maximum A-MPDU Length Exponent})} - 1$ octets.
Minimum MPDU Start Spacing	Determines the minimum time between the start of adjacent MPDUs within an A-MPDU that the STA can receive, measured at the PHY-SAP. See 9.12.3.	Set to 0 for no restriction Set to 1 for 1/4 µs Set to 2 for 1/2 µs Set to 3 for 1 µs Set to 4 for 2 µs Set to 5 for 4 µs Set to 6 for 8 µs Set to 7 for 16 µs

8.4.2.58.4 Supported MCS Set field

The Supported MCS Set field of the HT Capabilities element indicates which MCSs a STA supports.

An MCS is identified by an MCS index, which is represented by an integer in the range 0 to 76. The interpretation of the MCS index (i.e., the mapping from MCS to data rate) is PHY dependent. For the HT PHY, see 20.6.

The structure of the MCS Set field is defined in Figure 8-251.

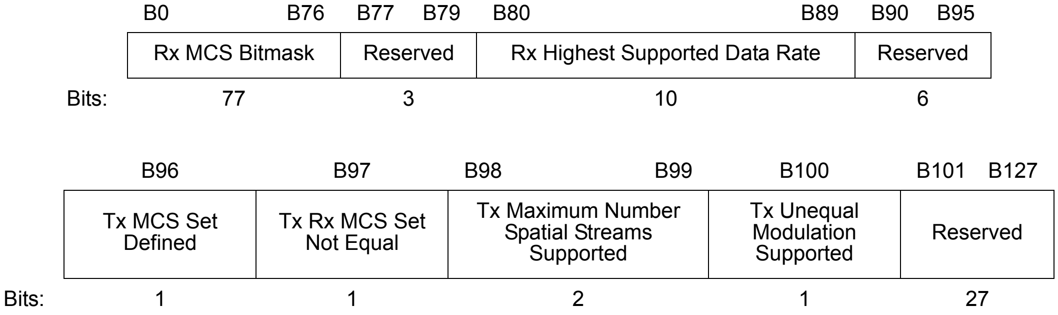


Figure 8-251—Supported MCS Set field

The Rx MCS Bitmask subfield of the Supported MCS Set field defines a set of MCS index values, where bit B0 corresponds to MCS 0 and bit B76 corresponds to MCS 76.

NOTE—An HT STA includes the mandatory MCS values defined in 20.1 in the Rx MCS Bitmask subfield.

The Rx Highest Supported Data Rate subfield of the Supported MCS Set field defines the highest data rate that the STA is able to receive, in units of 1 Mb/s, where 1 represents 1 Mb/s, and incrementing by 1 Mb/s steps to the value 1023, which represents 1023 Mb/s. If the maximum data rate expressed in Mb/s is not an integer, then the value is rounded up to the next integer. The value 0 indicates that this subfield does not specify the highest data rate that the STA is able to receive; see 9.7.6.5.3.

The Tx MCS Set Defined, Tx Rx MCS Set Not Equal, Tx Maximum Number Spatial Streams Supported, and Tx Unequal Modulation Supported subfields of the Supported MCS Set field indicate the transmit-supported MCS set, as defined in Table 8-126.

Table 8-126—Transmit MCS Set

Condition	Tx MCS Set Defined	Tx Rx MCS Set Not Equal	Tx Maximum Number Spatial Streams Supported	Tx Unequal Modulation Supported
No Tx MCS set is defined	0	0	0	0
The Tx MCS set is defined to be equal to the Rx MCS set	1	0	0	0
The Tx MCS set may differ from the Rx MCS set	1	1	Indicates the maximum number of spatial streams supported when transmitting: Set to 0 for 1 spatial stream Set to 1 for 2 spatial streams Set to 2 for 3 spatial streams Set to 3 for 4 spatial streams	Indicates whether transmit unequal modulation (UEQM) is supported: Set to 0 for UEQM not supported Set to 1 for UEQM supported

8.4.2.58.5 HT Extended Capabilities field

The structure of the HT Extended Capabilities field of the HT Capabilities element is defined in Figure 8-252.

	B0	B1 B2	B3 B7	B8 B9	B10	B11	B12 B15
	PCO	PCO Transition Time	Reserved	MCS Feedback	+HTC Support	RD Responder	Reserved
Bits:	1	2	5	2	1	1	4

Figure 8-252—HT Extended Capabilities field

The subfields of the HT Extended Capabilities field are defined in Table 8-127.

Table 8-127—Subfields of the HT Extended Capabilities field

Subfield	Definition	Encoding
PCO	Indicates support for PCO. When transmitted by an AP: indicates whether the AP can operate its BSS as a PCO BSS. When transmitted by a non-AP STA: indicates whether the STA can operate as a PCO active STA when the Transition Time subfield in its HT Extended Capabilities field meets the intended transition time of the PCO capable AP.	Set to 0 if not supported Set to 1 if supported
PCO Transition Time	When transmitted by a non-AP STA: indicates that the STA can switch between 20 MHz channel width and 40 MHz channel width within the specified time. When transmitted by an AP: indicates the PCO Transition Time to be used during PCO operation. The value contained in this field is dynamic when transmitted by an AP, i.e., the value of this field may change at any time during the lifetime of the association of a STA with the AP. See 10.16.3.	If the PCO subfield is equal to 0, this field is reserved. Otherwise: Set to 1 for 400 µs Set to 2 for 1.5 ms Set to 3 for 5 ms Set to 0 for no transition. In this case, the PCO active STA does not change its operating channel width and is able to receive 40 MHz PPDUs during the 20 MHz phase (see 10.16).
MCS Feedback	Indicates whether the STA can provide MFB	Set to 0 (No Feedback) if the STA does not provide MFB Set to 2 (Unsolicited) if the STA provides only unsolicited MFB Set to 3 (Both) if the STA can provide MFB in response to MRQ (either Delayed or Immediate, see 9.28.1) as well as unsolicited MFB The value 1 is reserved
+HTC Support	Indicates support of the HT Control field. See 9.9	Set to 0 if not supported Set to 1 if supported
RD Responder	Indicates support for acting as a reverse direction responder, i.e., the STA may use an offered RDG to transmit data to an RD initiator using the Reverse Direction Protocol described in 9.25.	Set to 0 if not supported Set to 1 if supported

The following subfield is reserved for a mesh STA: PCO.

8.4.2.58.6 Transmit Beamforming Capabilities

The structure of the Transmit Beamforming Capabilities field is defined in Figure 8-253.

B0	B1	B2	B3	B4	B5	B6 B7	B8
Implicit Transmit Beamforming Receiving Capable	Receive Staggered Sounding Capable	Transmit Staggered Sounding Capable	Receive NDP Capable	Transmit NDP Capable	Implicit Transmit Beamforming Capable	Calibration	Explicit CSI Transmit Beamforming Capable
Bits: 1	1	1	1	1	1	2	1

B9	B10	B11 B12	B13 B14	B15 B16	B17 B18	B19 B20
Explicit Noncompressed Steering Capable	Explicit Compressed Steering Capable	Explicit Transmit Beamforming CSI Feedback	Explicit Noncompressed Beamforming Feedback Capable	Explicit Compressed Beamforming Feedback Capable	Minimal Grouping	CSI Number of Beamformer Antennas Supported
1	1	2	2	2	2	2

B21 B22	B23 B24	B25 B26	B27 B28	B29 B31
Noncompressed Steering Number of Beamformer Antennas Supported	Compressed Steering Number of Beamformer Antennas Supported	CSI Max Number of Rows Beamformer Supported	Channel Estimation Capability	Reserved
2	2	2	2	3

Figure 8-253—Transmit Beamforming Capabilities field

The subfields of the Transmit Beamforming Capabilities field are defined in Table 8-128.

Table 8-128—Subfields of the Transmit Beamforming Capabilities field

Subfield	Definition	Encoding
Implicit Transmit Beamforming Receiving Capable	Indicates whether this STA can receive Transmit Beamforming steered frames using implicit feedback	Set to 0 if not supported Set to 1 if supported
Receive Staggered Sounding Capable	Indicates whether this STA can receive staggered sounding frames.	Set to 0 if not supported Set to 1 if supported
Transmit Staggered Sounding Capable	Indicates whether this STA can transmit staggered sounding frames.	Set to 0 if not supported Set to 1 if supported
Receive NDP Capable	Indicates whether this receiver can interpret null data packets as sounding frames.	Set to 0 if not supported Set to 1 if supported
Transmit NDP Capable	Indicates whether this STA can transmit null data packets as sounding frames.	Set to 0 if not supported Set to 1 if supported
Implicit Transmit Beamforming Capable	Indicates whether this STA can apply implicit transmit beamforming.	Set to 0 if not supported Set to 1 if supported

Table 8-128—Subfields of the Transmit Beamforming Capabilities field *(continued)*

Subfield	Definition	Encoding
Calibration	Indicates whether the STA can participate in a calibration procedure initiated by another STA that is capable of generating an immediate response sounding PPDU and can provide a CSI report in response to the receipt of a sounding PPDU.	Set to 0 if not supported Set to 1 if the STA can respond to a calibration request using the CSI report, but cannot initiate calibration The value 2 is reserved Set to 3 if the STA can both initiate and respond to a calibration request
Explicit CSI Transmit Beamforming Capable	Indicates whether this STA can apply transmit beamforming using CSI explicit feedback in its transmission	Set to 0 if not supported Set to 1 if supported
Explicit Noncompressed Steering Capable	Indicates whether this STA can apply transmit beamforming using noncompressed beamforming feedback matrix explicit feedback in its transmission	Set to 0 if not supported Set to 1 if supported
Explicit Compressed Steering Capable	Indicates whether this STA can apply transmit beamforming using compressed beamforming feedback matrix explicit feedback in its transmission	Set to 0 if not supported Set to 1 if supported
Explicit Transmit Beamforming CSI Feedback	Indicates whether this receiver can return CSI explicit feedback.	Set to 0 if not supported Set to 1 for delayed feedback Set to 2 for immediate feedback Set to 3 for delayed and immediate feedback
Explicit Noncompressed Beamforming Feedback Capable	Indicates whether this receiver can return noncompressed beamforming feedback matrix explicit feedback.	Set to 0 if not supported Set to 1 for delayed feedback Set to 2 for immediate feedback Set to 3 for delayed and immediate feedback
Explicit Compressed Beamforming Feedback Capable	Indicates whether this receiver can return compressed beamforming feedback matrix explicit feedback.	Set to 0 if not supported Set to 1 for delayed feedback Set to 2 for immediate feedback Set to 3 for delayed and immediate feedback
Minimal Grouping	Indicates the minimal grouping used for explicit feedback reports	Set to 0 if the STA supports groups of 1 (no grouping) Set to 1 indicates groups of 1, 2 Set to 2 indicates groups of 1, 4 Set to 3 indicates groups of 1, 2, 4
CSI Number of Beamformer Antennas Supported	Indicates the maximum number of beamformer antennas the beamformee can support when CSI feedback is required	Set to 0 for single Tx antenna sounding Set to 1 for 2 Tx antenna sounding Set to 2 for 3 Tx antenna sounding Set to 3 for 4 Tx antenna sounding
Noncompressed Steering Number of Beamformer Antennas Supported	Indicates the maximum number of beamformer antennas the beamformee can support when noncompressed beamforming feedback matrix is required	Set to 0 for single Tx antenna sounding Set to 1 for 2 Tx antenna sounding Set to 2 for 3 Tx antenna sounding Set to 3 for 4 Tx antenna sounding

Table 8-128—Subfields of the Transmit Beamforming Capabilities field *(continued)*

Subfield	Definition	Encoding
Compressed Steering Number of Beamformer Antennas Supported	Indicates the maximum number of beamformer antennas the beamformee can support when compressed beamforming feedback matrix is required	Set to 0 for single Tx antenna sounding Set to 1 for 2 Tx antenna sounding Set to 2 for 3 Tx antenna sounding Set to 3 for 4 Tx antenna sounding
CSI Max Number of Rows Beamformer Supported	Indicates the maximum number of rows of CSI explicit feedback from the beamformee or calibration responder or transmit ASEL responder that a beamformer or calibration initiator or transmit ASEL initiator can support when CSI feedback is required.	Set to 0 for a single row of CSI Set to 1 for 2 rows of CSI Set to 2 for 3 rows of CSI Set to 3 for 4 rows of CSI
Channel Estimation Capability	Indicates the maximum number of space-time streams (columns of the MIMO channel matrix) for which channel dimensions can be simultaneously estimated when receiving an NDP sounding PPDU or the extension portion of the HT Long Training fields (HT-LTFs) in a staggered sounding PPDU. See NOTE.	Set 0 for 1 space-time stream Set 1 for 2 space-time streams Set 2 for 3 space-time streams Set 3 for 4 space-time streams
NOTE—The maximum number of space-time streams for which channel coefficients can be simultaneously estimated using the HT-LTFs corresponding to the data portion of the packet is limited by the Rx MCS Bitmask subfield of the Supported MCS Set field and by the Rx STBC subfield of the HT Capabilities Info field. Both fields are part of the HT Capabilities element.		

8.4.2.58.7 ASEL Capability field

The structure of the ASEL Capability field of the HT Capabilities element is defined in Figure 8-254.

	B0	B1	B2	B3	B4	B5	B6	B7
	Antenna Selection Capable	Explicit CSI Feedback Based Transmit ASEL Capable	Antenna Indices Feedback Based Transmit ASEL Capable	Explicit CSI Feed-back Capable	Antenna Indices Feedback Capable	Receive ASEL Capable	Transmit Sounding PPDUs Capable	Reserved
Bits:	1	1	1	1	1	1	1	1

Figure 8-254—ASEL Capability field

The subfields of the ASEL Capability field are defined in Table 8-129.

Table 8-129—ASEL Capability field subfields

Subfield	Definition	Encoding
Antenna Selection Capable	Indicates whether this STA supports ASEL	Set to 0 if not supported Set to 1 if supported
Explicit CSI Feedback Based Transmit ASEL Capable	Indicates whether this STA supports transmit ASEL based on explicit CSI feedback	Set to 0 if not supported Set to 1 if supported
Antenna Indices Feedback Based Transmit ASEL Capable	Indicates whether this STA supports transmit ASEL based on antenna indices feedback	Set to 0 if not supported Set to 1 if supported
Explicit CSI Feedback Capable	Indicates whether this STA can compute CSI and provide CSI feedback in support of ASEL	Set to 0 if not supported Set to 1 if supported
Antenna Indices Feedback Capable	Indicates whether this STA can compute an antenna indices selection and return an antenna indices selection in support of ASEL	Set to 0 if not supported Set to 1 if supported
Receive ASEL Capable	Indicates whether this STA supports receive ASEL	Set to 0 if not supported Set to 1 if supported
Transmit Sounding PPDUs Capable	Indicates whether this STA can transmit sounding PPDUs for ASEL training on request	Set to 0 if not supported Set to 1 if supported

8.4.2.59 HT Operation element

The operation of HT STAs in the BSS is controlled by the HT Operation element. The structure of this element is defined in Figure 8-255.

	Element ID	Length	Primary Channel	HT Operation Information	Basic MCS Set
Octets:	1	1	1	5	16

Figure 8-255—HT Operation element format

The Element ID field is set to the value for HT Operation element defined in Table 8-54.

The structure of the HT Operation Information field is shown in Figure 8-256.

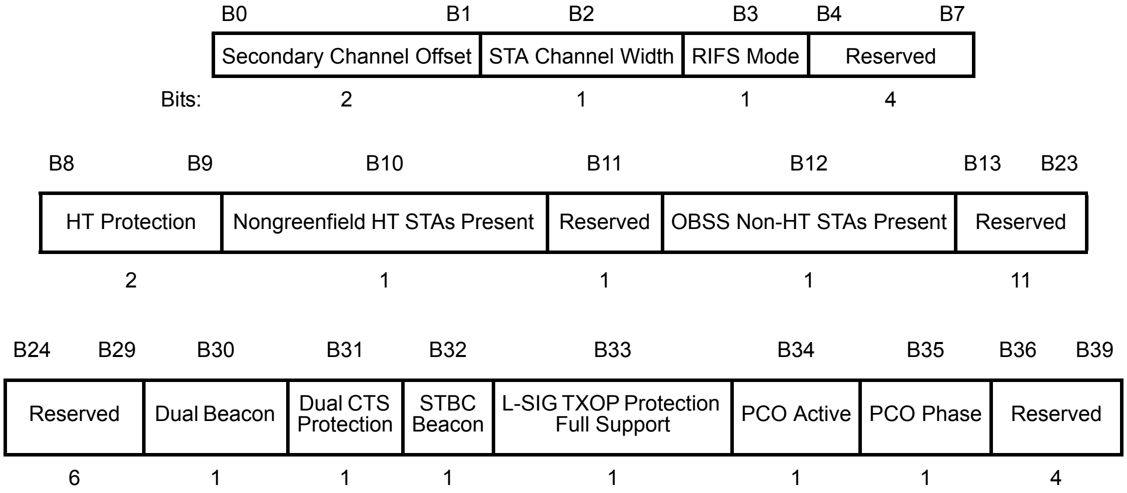


Figure 8-256—HT Operation Information field

The Primary Channel field, subfields of the HT Operation Information field, and the Basic MCS Set field are defined in Table 8-130. The "Reserved in IBSS?" column indicates whether each field is reserved (Y) or not reserved (N) when this element is present in a frame transmitted within an IBSS. The "Reserved in MBSS?" column indicates whether each field is reserved (Y) or not reserved (N) when this element is present in a frame transmitted within an MBSS.

Table 8-130—HT Operation element fields and subfields

Field	Definition	Encoding	Reserved in IBSS?	Reserved in MBSS?
Primary Channel	Indicates the channel number of the primary channel. See 10.15.2.	Channel number of the primary channel	N	N
Secondary Channel Offset	Indicates the offset of the secondary channel relative to the primary channel.	Set to 1 (SCA) if the secondary channel is above the primary channel Set to 3 (SCB) if the secondary channel is below the primary channel Set to 0 (SCN) if no secondary channel is present The value 2 is reserved	N	N
STA Channel Width	Defines the channel widths that may be used to transmit to the STA. See 10.15.12	Set to 0 for a 20 MHz channel width Set to 1 allows use of any channel width in the Supported channel width set This field is reserved when the transmitting or receiving STA is operating in an operating class that does not include a value of 13 or 14 in the behavior limits as specified in Annex E. See NOTE 1.	N	N

Table 8-130—HT Operation element fields and subfields *(continued)*

Field	Definition	Encoding	Reserved in IBSS?	Reserved in MBSS?
RIFS Mode	Indicates whether the use of reduced interframe space is permitted within the BSS. See 9.3.2.3.2 and 9.23.3.3	Set to 0 if use of RIFS is prohibited Set to 1 if use of RIFS is permitted	Y	Y
HT Protection	Indicates protection requirements of HT transmissions. See 9.23.3.	Set to 0 for no protection mode Set to 1 for nonmember protection mode Set to 2 for 20 MHz protection mode Set to 3 for non-HT mixed mode	Y	N
Nongreenfield HT STAs Present	AP indicates if any HT STAs that are not HT-greenfield capable have associated. Mesh STA indicates if it establishes a mesh peering with an HT STA that is not HT-greenfield capable. Determines when a non-AP STA should use HT-greenfield protection. Present in Beacon and Probe response frames transmitted by an AP or mesh STA. Otherwise reserved. See 9.23.3.1.	Set to 0 if all HT STAs that are associated are HT-greenfield capable or all HT peer mesh STAs are HT-greenfield capable Set to 1 if one or more HT STAs that are not HT-greenfield capable are associated or one or more HT peer mesh STAs are not HT-greenfield capable	Y	N
OBSS Non-HT STAs Present	Indicates if the use of protection for non-HT STAs by overlapping BSSs is determined to be desirable. If the BSS is operating in an operating class for which the behavior limits set listed in Annex E includes the value 16, this field indicates if there exist any non-HT OBSSs and whether HT-greenfield transmissions are allowed. Present in Beacon and Probe response frames transmitted by an AP. Otherwise reserved. See 9.23.3.4 and 10.9.8.5.	If not operating in an operating class for which the behavior limits set listed in Annex E includes the value 16: Set to 1 if the use of protection for non-HT STAs by OBSSs is determined to be desirable. See NOTE 2. Set to 0 otherwise. If operating in an operating class for which the behavior limits set listed in Annex E includes the value 16: Set to 1 if there exists one or more non-HT OBSSs. Indicates that HT-greenfield transmissions are disallowed in the BSS. Set to 0 otherwise.	Y	Y
Dual Beacon	Indicates whether the AP transmits an STBC beacon.	Set to 0 if no STBC beacon is transmitted Set to 1 if an STBC beacon is transmitted by the AP	Y	Y

Table 8-130—HT Operation element fields and subfields *(continued)*

Field	Definition	Encoding	Reserved in IBSS?	Reserved in MBSS?
Dual CTS Protection	Dual CTS protection is used by the AP to set a NAV at STAs that do not support STBC and at STAs that can associate solely through the STBC beacon. See 9.3.2.7.	Set to 0 if dual CTS protection is not required Set to 1 if dual CTS protection is required	Y	Y
STBC Beacon	Indicates whether the beacon containing this element is a primary or an STBC beacon. The STBC beacon has half a beacon period shift relative to the primary beacon. Defined only in a Beacon transmitted by an AP. Otherwise reserved. See 10.1.3.2.	Set to 0 in a primary beacon Set to 1 in an STBC beacon	Y	Y
L-SIG TXOP Protection Full Support	Indicates whether all HT STA in the BSS support L-SIG TXOP protection. See 9.23.5.	Set to 0 if one or more HT STA in the BSS do not support L-SIG TXOP protection Set to 1 if all HT STA in the BSS support L-SIG TXOP protection	Y	Y
PCO Active	Indicates whether PCO is active in the BSS Present in Beacon/Probe Response frames transmitted by an AP. Otherwise reserved. Non-PCO STAs regard the BSS as a 20/40 MHz BSS and may associate with the BSS without regard to this field. See 10.16.	Set to 0 if PCO is not active in the BSS Set to 1 if PCO is active in the BSS	Y	Y
PCO Phase	Indicates the PCO phase of operation Defined only in a Beacon and Probe Response frames when PCO Active is 1. Otherwise reserved. See 10.16.	Set to 0 indicates switch to or continue 20 MHz phase Set to 1 indicates switch to or continue 40 MHz phase	Y	Y

Table 8-130—HT Operation element fields and subfields *(continued)*

Field	Definition	Encoding	Reserved in IBSS?	Reserved in MBSS?
Basic MCS Set	Indicates the MCS values that are supported by all HT STAs in the BSS. Present in Beacon, Probe Response, Mesh Peering Open and Mesh Peering Confirm frames. Otherwise reserved.	The Basic MCS Set is a bitmap of size 128 bits. Bit 0 corresponds to MCS 0. A bit is set to 1 to indicate support for that MCS and 0 otherwise. MCS values are defined in 8.4.2.58.4.	N	N

NOTE 1—Any change of STA Channel Width field value does not impact the value of the HT Protection field.

NOTE 2—Examples of when this bit may be set to 1 include, but are not limited to, the following situations:

- One or more non-HT STAs are associated
- A non-HT BSS is overlapping (a non-HT BSS may be detected by the reception of a Beacon where the supported rates contain only Clause 16, Clause 18, Clause 17, or Clause 19 rates)
- A management frame (excluding a Probe Request) is received where the supported rate set includes only Clause 16, Clause 18, Clause 17, and Clause 19 rates

8.4.2.60 20/40 BSS Intolerant Channel Report element

The 20/40 BSS Intolerant Channel Report element contains a list of channels on which a STA has found conditions that disallow the use of a 20/40 MHz BSS. The format of the 20/40 BSS Intolerant Channel Report element is shown in Figure 8-257.

	Element ID	Length	Operating Class	Channel List
Octets:	1	1	1	Variable

Figure 8-257—20/40 BSS Intolerant Channel Report element format

The Element ID field is set to the value of 20/40 BSS Intolerant Channel Report element defined in Table 8-54.

The Length field of the 20/40 BSS Intolerant Channel Report element is variable and depends on the number of channels reported in the Channel List field. The minimum value of the Length field is 1 (based on a minimum length for the Channel List field of 0 octets).

Operating Class field of the 20/40 MHz BSS Intolerant Channel Report element contains an enumerated value from Annex E, encoded as an unsigned integer, specifying the operating class in which the channel list is valid. A 20/40 BSS Intolerant Channel Report only reports channels for a single operating class. Multiple 20/40 BSS Intolerant Channel Report elements are used to report channels in more than one operating class.

The Channel List field of the 20/40 MHz BSS Intolerant Channel Report element a variable number of octets, where each octet describes a single channel number. Channel numbering shall be dependent on operating class according to Annex E.

A 20/40 BSS Intolerant Channel Report element includes only channels that are valid for the regulatory domain in which the STA transmitting the element is operating and that are consistent with the Country element transmitted by the AP of the BSS of which it is a member.

8.4.2.61 Overlapping BSS Scan Parameters element

The Overlapping BSS Scan Parameters element is used by an AP in a BSS to indicate the values to be used by BSS members when performing OBSS scan operations. The format of the Overlapping BSS Scan Parameters element is shown in Figure 8-258.

	Element ID	Length	OBSS Scan Passive Dwell	OBSS Scan Active Dwell	BSS Channel Width Trigger Scan Interval	OBSS Scan Passive Total Per Channel	OBSS Scan Active Total Per Channel	BSS Width Channel Transition Delay Factor	OBSS Scan Activity Threshold
Octets:	1	1	2	2	2	2	2	2	2

Figure 8-258—Overlapping BSS Scan Parameters element format

The Element ID field value is equal to the Overlapping BSS Scan Parameters element value in Table 8-54.

The Length field is set to 14.

The OBSS Scan Passive Dwell field contains a value in TUs, encoded as an unsigned integer, that a receiving STA uses to set dot11OBSSScanPassiveDwell as described in 10.15.5.

The OBSS Scan Active Dwell field contains a value in TUs, encoded as an unsigned integer, that a receiving STA uses to set dot11OBSSScanActiveDwell as described in 10.15.5.

The BSS Channel Width Trigger Scan Interval field contains a value in seconds, encoded as an unsigned integer, that a receiving STA uses to set dot11BSSWidthTriggerScanInterval as described in 10.15.5.

The OBSS Scan Passive Total Per Channel field contains a value in TUs, encoded as an unsigned integer, that a receiving STA uses to set dot11OBSSScanPassiveTotalPerChannel as described in 10.15.5.

The OBSS Scan Active Total Per Channel field contains a value in TUs, encoded as an unsigned integer, that a receiving STA uses to set dot11OBSSScanActiveTotalPerChannel as described in 10.15.5.

The BSS Width Channel Transition Delay Factor field contains an integer value that a receiving STA uses to set dot11BSSWidthChannelTransitionDelayFactor as described in 10.15.5.

The OBSS Scan Activity Threshold field contains a value in hundredths of percent, encoded as an unsigned integer, that a receiving STA uses to set dot11OBSSScanActivityThreshold as described in 10.15.5.

The use of each of these parameters is described in 10.15.5.

8.4.2.62 20/40 BSS Coexistence element

The 20/40 BSS Coexistence element is used by STAs to exchange information that affects 20/40 BSS coexistence.

The 20/40 BSS Coexistence element is formatted as shown in Figure 8-259.

	Element ID	Length	20/40 BSS Coexistence Information field
Octets:	1	1	1

Figure 8-259—20/40 BSS Coexistence element format

The Element ID field is set to the value for 20/40 BSS Coexistence element defined in Table 8-54.

The structure of the 20/40 BSS Coexistence Information field is shown in Figure 8-260.

	B0	B1	B2	B3	B4	B5 B7
	Informatio n Request	Forty MHz Intolerant	20 MHz BSS Width Request	OBSS Scanning Exemption Request	OBSS Scanning Exemption Grant	Reserved
Bits:	1	1	1	1	1	3

Figure 8-260—20/40 BSS Coexistence Information field

The Information Request field is used to indicate that a transmitting STA is requesting the recipient to transmit a 20/40 BSS Coexistence Management frame with the transmitting STA as the recipient.

The Forty MHz Intolerant field is set to 1 to prohibit an AP that receives this information or reports of this information from operating a 20/40 MHz BSS. When equal to 0, it does not prohibit a receiving AP from operating a 20/40 MHz BSS. This field is used for inter-BSS communication. The definition of this field is the same as the definition of the Forty MHz Intolerant field in the HT Capabilities element (see 8.4.2.58), and its operation is described in 10.15.11.

The 20 MHz BSS Width Request field is set to 1 to prohibit a receiving AP from operating its BSS as a 20/40 MHz BSS. Otherwise, it is set to 0. This field is used for intra-BSS communication. The operation of this field is described in 10.15.12.

The OBSS Scanning Exemption Request field is set to 1 to indicate that the transmitting non-AP STA is requesting the BSS to allow the STA to be exempt from OBSS scanning. Otherwise, it is set to 0. The OBSS Scanning Exemption Request field is reserved when transmitted by an AP. The OBSS Scanning Exemption Request field is reserved when a 20/40 BSS Coexistence element is included in a group addressed frame.

The OBSS Scanning Exemption Grant field is set to 1 by an AP to indicate that the receiving STA is exempted from performing OBSS Scanning. Otherwise, it is set to 0. The OBSS Scanning Exemption Grant field is reserved when transmitted by a non-AP STA. The OBSS Scanning Exemption Grant field is reserved when a 20/40 BSS Coexistence element is included in a group addressed frame.

8.4.2.63 Time Advertisement element

The Time Advertisement element, shown in Figure 8-261, specifies fields describing the source of time corresponding to a time standard, an external clock (external time source), an estimate of the offset between that time standard and the TSF timer, and an estimate of the standard deviation of the error in the offset estimate. This information is used by a receiving STA to align its own estimate of the time standard based on that of another STA.

	Element ID	Length	Timing Capabilities	Time Value (optional)	Time Error (optional)	Time Update Counter (optional)
Octets:	1	1	1	0 or 10	0 or 5	0 or 1

Figure 8-261—Time Advertisement element format

The Timing Capabilities field specifies the STA's source and encoding of the Time Value field. The encoding of the Timing Capabilities field is specified in Table 8-131.

Table 8-131—Encoding of the Timing Capabilities field

Value	Usage
0	No standardized external time source
1	Timestamp offset based on UTC [see ITU-R Recommendation TF.460-4(2002) [B48]]. The Timestamp offset value in nanoseconds is defined to be 0 at the beginning of the first nanosecond of the first day of the year 1958.
2	UTC time at which the TSF timer is 0.
3–255	Reserved

When the value of the Timing Capabilities field is 0, only the Element ID, Length, and Timing Capabilities fields are included in the Time Advertisement element.

When the value of the Timing Capabilities is 1, the following additional fields are included in the Time Advertisement element:

— Time Value field, a two's complement integer in nanoseconds that, when added to the Timestamp present in the same transmitted frame, gives the receiving STA an estimate of the time standard at the time the frame was transmitted. The Timestamp is derived from the TSF Timer as defined in 10.21.

— Time Error field, which is set to an unsigned integer in nanoseconds that defines the standard deviation of the error in the Time Value estimate. The value of all 1s is used to indicate that the error is unknown.

When the Timing Capabilities field is 2, the following fields are included in the Time Advertisement element:

— The Time Value field is the UTC time at which the TSF Timer is 0, given that the TSF Timer units are 1 microsecond units as defined in 10.1.3. The format, including all subfields is shown in Table 8-132. For any subfield not known in the Time Value field, the subfield value is 0.

— The Time Error field is an unsigned integer in milliseconds that defines the standard deviation of the error in the Time Value estimate.

— The Time Update Counter field is a modulo 256 counter that increments each time the AP updates the Time Value UTC at which the TSF Timer is 0.

Table 8-132—Time Value field format when Timing Capabilities is 2

Octet	Description
0–1	Year (0–65 534)
2	Month (0–12)
3	Day of month (0–31)
4	Hours (0–23)
5	Minutes (0–59)
6	Seconds (0–59)
7–8	Milliseconds (0–999)
9	Reserved

8.4.2.64 Link Identifier element

The Link Identifier element contains information that identifies a TDLS direct link. The element information format is defined in Figure 8-262.

	Element ID	Length	BSSID	TDLS initiator STA Address	TDLS responder STA Address
Octets:	1	1	6	6	6

Figure 8-262—Link Identifier element format

The Element ID field is defined in Table 8-54.

The Length field is set to 18.

The BSSID field is set to the BSSID of the BSS to which the TDLS initiator STA is associated.

The TDLS initiator STA Address field is set to the TDLS initiator STA’s MAC address.

The TDLS responder STA Address field is set to the TDLS responder STA’s MAC address.

8.4.2.65 Wakeup Schedule element

The Wakeup Schedule element contains information regarding the periodic wakeup schedule for TDLS Peer Power Save Mode. The element format is defined in Figure 8-263.

	Element ID	Length	Offset	Interval	Awake Window Slots	Maximum Awake Window Duration	Idle Count
Octets:	1	1	4	4	4	4	2

Figure 8-263—Wakeup Schedule element format

The Element ID field is defined in Table 8-54.

The Length field is set to 18.

The Offset field is the time in microseconds between TSF 0 and the start of a first Awake Window. See 10.2.1.14.

The Interval field is set to the time in microseconds between the start of two successive Awake Windows. See 10.2.1.14.

The Awake Window Slots field is set to the duration of the Awake Window in units of backoff slots (see 9.19.2.3). See 10.2.1.14.

The Maximum Awake Window Duration field is set to the maximum duration of the Awake Window, in units of microseconds. See 10.2.1.14.

The Idle Count field is set to the number of consecutive Awake Windows during which no individually addressed frame is received from the TDLS peer STA before a TDLS peer STA deletes the wakeup schedule. See 10.2.1.14.

8.4.2.66 Channel Switch Timing element

The Channel Switch Timing element contains information regarding the channel switch timing. The element is defined in Figure 8-264.

	Element ID	Length	SwitchTime	Switch Timeout
Octets:	1	1	2	2

Figure 8-264—Channel Switch Timing element format

The Element ID field is defined in Table 8-54

The Length field is set to 4.

The Switch Time field is set to the time it takes for a STA sending the Channel Switch Timing element to switch channels, in units of microseconds.

The Switch Timeout field is set to a time in units of microseconds. The STA sending the Channel Switch Timing element waits for the first data frame exchange on the off-channel for Switch Timeout microseconds before switching back to base channel. The time is measured from the end of the last symbol of the ACK frame that is transmitted in response to TDLS Channel Switch Response frame, as seen at the air interface.

8.4.2.67 PTI Control element

The PTI Control element contains information regarding the traffic buffered at the TPU buffer STA for the TPU sleep STA at the time a TDLS Peer Traffic Indication frame is transmitted by the TPU buffer STA. The element is optionally included in the TDLS Peer Traffic Indication frame. The element is defined in Figure 8-265.

	Element ID	Length	TID	Sequence Control
Octets:	1	1	1	2

Figure 8-265—PTI Control element format

The Element ID field is defined in Table 8-54.

The Length field is set to 3.

The TID field contained in the PTI Control element is set to the TID of the latest MPDU that has been transmitted over the TDLS direct link to the TPU sleep STA that is the destination of the TDLS Peer Traffic Indication frame that contains the PTI Control element. See 10.2.1.15.

The Sequence Control field is defined in 8.2.4.4. The Sequence Control field contained in the PTI Control element is set to the sequence number of the latest MPDU that has been transmitted over the TDLS direct link to the TPU sleep STA that is the destination of the TDLS Peer Traffic Indication frame that contains the PTI Control element. See 10.2.1.15.

8.4.2.68 TPU Buffer Status element

The TPU Buffer Status element contains information regarding the traffic buffered at the TPU buffer STA for the TPU sleep STA at the time a TDLS Peer Traffic Indication frame is transmitted by the TPU buffer STA. The element is included in the TDLS Peer Traffic Indication frame. The element is defined in Figure 8-266.

			B0	B1	B2	B3	B4 B7
	Element ID	Length	AC_BK traffic available	AC_BE traffic available	AC_VI traffic available	AC_VO traffic available	Reserved
Octets:	1	1	1				

Figure 8-266—TPU Buffer Status element format

The Element ID field is defined in Table 8-54.

The Length field is set to 1.

The AC_BK traffic available field is one bit in size and is set to 1 if AC_BK contains traffic buffered for the TPU sleep STA to which the TPU Buffer Status element will be transmitted, and is set to 0 otherwise.

The AC_BE traffic available field is one bit in size and is set to 1 if AC_BE contains traffic buffered for the TPU sleep STA to which the TPU Buffer Status element will be transmitted, and is set to 0 otherwise.

The AC_VI traffic available field is one bit in size and is set to 1 if AC_VI contains traffic buffered for the TPU sleep STA to which the TPU Buffer Status element will be transmitted, and is set to 0 otherwise.

The AC_VO traffic available field is one bit in size and is set to 1 if AC_VO contains traffic buffered for the TPU sleep STA to which the TPU Buffer Status element will be transmitted, and is set to 0 otherwise.

8.4.2.69 Event Request element

8.4.2.69.1 Event Request definition

The Event Request element contains a request to the receiving STA to perform the specified event action. The format of the Event Request element is shown in Figure 8-267.

	Element ID	Length	Event Token	Event Type	Event Response Limit	Event Request
Octets:	1	1	1	1	1	variable

Figure 8-267—Event Request element format

The Element ID field is equal to the Event Request value in Table 8-54.

The value of the Length field is 3 plus the length of the Event Request field.

The Event Token field is a nonzero number that is unique among the Event Request elements sent to each destination MAC address for which a corresponding Event Report element has not been received.

The Event Type field is a number that identifies the type of event request. The Event Types are shown in Table 8-133.

Table 8-133—Event Type definitions for event requests and reports

Name	Event Type
Transition	0
RSNA	1
Peer-to-Peer Link	2
WNM Log	3
Reserved	4–220
Vendor Specific	221
Reserved	222–255

The Event Response Limit field contains the maximum number of requested Event Reports to be included in the Event Report element. A value of 0 indicates that no limit is set on the number of Event Reports to be included in the Event Report element.

The Event Request field contains the event request corresponding to the Event Type, as described in 8.4.2.69.2 through 8.4.2.69.4. The Event Request field is not present when requesting a WNM Log report.

The Event Request element is included in an Event Request frame, as described in 8.5.14.2. The use of the Event Request element and Event Request frame is described in 10.23.2.

8.4.2.69.2 Transition event request

The Event Request field corresponding to the Transition event request contains zero or more Transition Event Request subelements. A transition event is a STA movement or attempted movement from one BSS (the source BSS) in one ESS to another BSS (the target BSS) within the same ESS.

The Transition Event subelements specify the conditions in which a Transition Event Report is sent by a STA. The set of valid Transition Event Request subelements is defined in Table 8-134.

Table 8-134—Transition Event Request subelement

Order	Transition Event Request subelement	Subelement ID
1	Transition Target BSSID	0
2	Transition Source BSSID	1
3	Transition Time Threshold	2
4	Transition Result	3
5	Frequent Transition	4
—	Reserved	5–255

The Transition Target BSSID subelement is used to request that a Transition Event Report includes the transition event entry when the target BSSID is equal to the specific BSSID in the Target BSSID field. Excluding this subelement from the Event Request element indicates a request for transition events for all target BSSIDs. The format of the Transition Target BSSID subelement is shown in Figure 8-268.

	Subelement ID	Length	Target BSSID
Octets:	1	1	6

Figure 8-268—Transition Target BSSID subelement format

The Subelement ID field is equal to the Transition Target BSSID value in Table 8-134.

The value of the Length field is 6.

The Target BSSID field contains a 6-octet BSSID.

The Transition Source BSSID subelement is used to request that a Transition Event Report includes the transition event entry when the source BSSID is equal to the BSSID specified in the Source BSSID field. Excluding this subelement from the Event Request element indicates a request for transition events for all source BSSIDs. The format of the Transition Source BSSID subelement is shown in Figure 8-269.

	Subelement ID	Length	Source BSSID
Octets:	1	1	6

Figure 8-269—Transition Source BSSID subelement format

The Subelement ID field is equal to the Transition Source BSSID value in Table 8-134.

The value of the Length field is 6.

The Source BSSID field contains a 6-octet BSSID.

The Transition Time Threshold subelement is used to request that a Transition Event Report includes the transition event entry when the Transition Time is greater than or equal to the Transition Time Threshold value. The format of the Transition Time subelement is shown in Figure 8-270.

	Subelement ID	Length	Transition Time Threshold
Octets:	1	1	2

Figure 8-270—Transition Time Threshold subelement format

The Subelement ID field is equal to the Transition Time Threshold value in Table 8-134.

The value of the Length field is 2.

The Transition Time Threshold field contains a value representing the Transition Time to be used as the threshold value for the Transition Time condition in TUs. The Transition Time is defined in 10.23.2.2.

The Transition Result subelement is used to request that a Transition Event Report includes the transition event entry that matches the transition result defined by this subelement. The format of Transition Result subelement is shown in Figure 8-271.

	Subelement ID	Length	Match Value
Octets:	1	1	1

Figure 8-271—Transition Result subelement format

The Subelement ID field is equal to the Transition Result value in Table 8-134.

The value of the Length field is 1.

The Match Value field is set with each bit as defined in Figure 8-272 to request that the specified transition results that match the bit descriptions are included in the Transition Event Report.

	B0	B1	B2-B7
	Include Successful Transitions	Include Failed Transitions	Reserved
Bits	1	1	6

Figure 8-272—Match Value field definitions

The Frequent Transition subelement is used to request that an alerting Transition Event report be generated when the total transition count during the specified time period is equal to or greater than the value given in Frequent Transition Count Threshold field. The format of the Frequent Transition subelement is shown in Figure 8-273.

	Subelement ID	Length	Frequent Transition Count Threshold	Time Interval
Octets:	1	1	1	2

Figure 8-273—Frequent Transition subelement format

The Subelement ID field is equal to the Frequent Transition value in Table 8-134.

The value of the Length field is 3.

The Frequent Transition Count Threshold field is a 1-octet field containing the number of transitions in the measurement duration after which a Transition Event Report is generated.

The Time Interval field is the time interval in TUs during which the STA determines if the Frequent Transition Count Threshold is exceeded.

8.4.2.69.3 RSNA event request

The Event Request field corresponding to an RSNA event request contains zero or more RSNA Event Request subelements.

The RSNA Event Request subelements are defined to have a common format consisting of a 1 octet Subelement ID field, a 1 octet Length field, and a variable-length subelement specific information field. See Figure 8-402. The set of valid RSNA Event Request subelements is defined in Table 8-135.

Table 8-135—RSNA Event Request subelement

Order	RSNA Event Request subelement	Subelement ID
1	RSNA Target BSSID	0
2	Authentication Type	1
3	EAP Method	2
4	RSNA Result	3
—	Reserved	4–255

The RSNA subelements specify reporting conditions for RSNA Event Reports.

The RSNA Target BSSID subelement identifies the BSS at which an RSNA Event establishment was attempted. Excluding this subelement from the Event Request element indicates a request for transition events for all source BSSIDs. The format of the RSNA Target BSSID subelement is shown in Figure 8-274.

	Subelement ID	Length	Target BSSID
Octets:	1	1	6

Figure 8-274—RSNA Target BSSID subelement format

The Subelement ID field is equal to the RSNA Target BSSID value in Table 8-135.

The value of the Length field is 6.

The Target BSSID field contains a 6-octet BSSID.

The Authentication Type subelement is used to request that an RSNA Event Report includes the RSNA event entry when the Authentication Type is equal to the authentication type specified in the Authentication Type field. The format of the Authentication Type subelement is shown in Figure 8-275.

	Subelement ID	Length	Authentication Type
Octets:	1	1	4

Figure 8-275—Authentication Type subelement format

The Subelement ID field is equal to the Authentication Type value in Table 8-135.

The value of the Length field is 4.

The Authentication Type field contains one of the AKM suite selectors defined in Table 8-101 in 8.4.2.27.3.

The EAP Method subelement is used to request that an RSNA Event Report includes the RSNA event entry when the EAP Method is equal to the EAP method specified in the EAP Method field. The format of the EAP Method subelement is shown in Figure 8-276.

	Subelement ID	Length	EAP Type	EAP Vendor ID (optional)	EAP Vendor Type (optional)
Octets:	1	1	1	0 or 3	0 or 4

Figure 8-276—EAP Method subelement format

The Subelement ID field is equal to the EAP Method value in Table 8-135.

The value of the Length field is 1 or 8.

The EAP Type field contains a value that identifies a single EAP method and is any valid IANA assigned EAP type.

The EAP Vendor ID field contains a value that identifies the EAP Vendor. The EAP Vendor ID field is included when EAP Type field is 254, and is excluded otherwise.

The EAP Vendor Type field contains a value that identifies the EAP Type as defined by the vendor. The EAP Vendor Type field is included when EAP Type field is 254, and is excluded otherwise.

The RSNA Result subelement is used to request that an RSNA Event Report includes the RSNA event entry that matches the transition result defined by this subelement. The format of RSNA Result subelement is shown in Figure 8-277.

	Subelement ID	Length	Match Value
Octets:	1	1	1

Figure 8-277—RSNA Result subelement format

The Subelement ID field is equal to the RSNA Result value in Table 8-135.

The value of the Length field is 1.

The Match Value field bits are set as defined in Figure 8-278 to request that the specified RSNA results that match that bit descriptions are included in the RSNA Event Report.

	B0	B1	B2-B7
	Include Successful RSNA	Include Failed RSNA	Reserved
Bits	1	1	6

Figure 8-278—Match Value field definitions

8.4.2.69.4 Peer-to-Peer Link event request

The Event Request field corresponding to Peer-to-Peer Link event request contains zero or more Peer-to-Peer Link Event Request subelements.

The Peer-to-Peer Link Event Request subelements are defined to have a common format consisting of a 1 octet Subelement ID field, a 1 octet Length field, and a variable-length subelement specific information field. The set of valid Peer-to-Peer Link Event Request subelements is defined in Table 8-136.

Table 8-136—Peer-to-Peer Link Event Request subelement

Order	Peer-to-Peer Link Event Request subelement	Subelement ID
1	Peer Address	0
2	Channel Number	1
—	Reserved	2–255

The Peer-to-Peer Link subelements specify reporting conditions for Peer-to-Peer Link Event Reporting.

The Peer Address subelement identifies the peer STA, BSS, or IBSS of the Peer-to-Peer links to be reported. Excluding this subelement from the Event Request element indicates a request for Peer-to-Peer Link events for any peer STA, any BSS, and any IBSS. The format of the Peer Address subelement is shown in Figure 8-279.

	Subelement ID	Length	Peer STA/ BSSID Address
Octets:	1	1	6

Figure 8-279—Peer Address subelement format

The Subelement ID field is equal to the Peer Address value in Table 8-136.

The value of the Length field is 6.

The Peer STA/BSSID Address field contains a 6-octet MAC address of a peer STA or a BSSID for Peer-to-Peer links in an IBSS. If the indicated address matches the Address 1 field of the MAC Header contents (see Table 8-19), then the address is a peer STA address for a TDLS or IBSS. If the indicated address matches the Address 3 field of the MAC Header contents, then the address is a BSSID for the Direct Link in an infrastructure BSS or for the IBSS.

The Channel Number subelement identifies the channel for the Peer-to-Peer links to be reported. Excluding this subelement from the Event Request element indicates a request for Peer-to-Peer Link events for any channel. The format of the Channel Number subelement is shown in Figure 8-280.

	Subelement ID	Length	Operating Class	Channel Number
Octets:	1	1	1	1

Figure 8-280—Channel Number subelement format

The Subelement ID field is equal to the Channel Number value in Table 8-136.

The value of the Length field is 2.

The Operating Class field indicates the channel set of the Peer-to-Peer link to be used for the Peer-to-Peer Link event report. Operating Classes are defined in Annex E.

The Channel Number field indicates the channel number of the Peer-to-Peer Link events requested and included in the Peer-to-Peer Link event report. A Channel Number of 0 indicates a request to report any Peer-to-Peer Link event for any supported channel in the specified filtering Operating Class.

8.4.2.69.5 Vendor Specific event request

The Event Request field corresponding to Vendor Specific event request contains zero or more Vendor Specific subelements. The Vendor Specific subelement has the same format as the Vendor Specific element (see 8.4.2.28).

8.4.2.70 Event Report element

8.4.2.70.1 Event Report Definition

The Event Report element is used by a STA to report an event. The format of the Event Report element is shown in Figure 8-281.

	Element ID	Length	Event Token	Event Type	Event Report Status
Octets:	1	1	1	1	1

	Event TSF (optional)	UTC Offset (optional)	Event Time Error (optional)	Event Report (optional)
Octets:	0 or 8	0 or 10	0 or 5	variable

Figure 8-281—Event Report element format

The Element ID field is equal to the Event Report value in Table 8-54.

The value of the Length field is 3 or 26 plus the length of the Event Report field.

The Event Token field is the Event Token in the corresponding Event Request element. If the Event Report element is being sent autonomously then the Event Token is 0.

The Event Type field is a number that identifies the type of event report. The Event Types are shown in Table 8-133.

The Event Report Status field is a value in Table 8-137, indicating the STA's response to the Event Request.

Table 8-137—Event Report Status

Event Report Status	Description
0	Successful
1	Request failed
2	Request refused
3	Request incapable
4	Detected frequent transition
5–255	Reserved

The Event TSF, UTC Offset, Event Time Error, and Event Report fields are present only when the Event Report Status field is 0.

The Event TSF field is TSF value when the STA logged the event.

The UTC Offset field is the UTC value that corresponds to the UTC time when the TSF timer is equal to 0. If the UTC Offset is unknown, the field is 0.

The Event Time Error field is the UTC standard deviation, as described in 8.4.2.63, that corresponds to the TSF value logged for the event. If the Event Time Error is unknown, the field is 0.

The Event Report field contains the specification of a single event report, as described in 8.4.2.70.2 through 8.4.2.70.5.

The Event Report element is included in an Event Report frame, as described in 8.5.14.3. The use of the Event Report element and frame is described in 10.23.2.

8.4.2.70.2 Transition event report

The format of the Event Report field corresponding to a Transition event report is shown in Figure 8-282.

	Source BSSID	Target BSSID	Transition Time	Transition Reason	Transition Result
Octets:	6	6	2	1	2

	Source RCPI	Source RSNI	Target RCPI	Target RSNI
Octets:	1	1	1	1

Figure 8-282—Event Report format for Transition event

The Source BSSID field contains the 6-octet BSSID address of the associated AP prior to the attempted transition.

The Target BSSID field contains the 6-octet BSSID address of the AP that is the target of the attempted Transition.

The Transition Time field contains the transition time in TUs. The transition time is defined in 10.23.2.2.

The Transition Reason field indicates the reason why a transition attempt occurred and contains one of the values in Table 8-138.

Table 8-138—Transition and Transition Query reasons

Transition Reason value	Description
0	Unspecified
1	Excessive frame loss rates and/or poor conditions
2	Excessive delay for current traffic streams
3	Insufficient QoS capacity for current traffic streams (TSPEC rejected)
4	First association to ESS (the association initiated by an Association Request message instead of a Reassociation Request message)
5	Load balancing
6	Better AP found
7	Deauthenticated or Disassociated from the previous AP
8	AP failed IEEE 802.1X EAP Authentication
9	AP failed 4-Way Handshake

Table 8-138—Transition and Transition Query reasons *(continued)*

Transition Reason value	Description
10	Received too many replay counter failures
11	Received too many data MIC failures
12	Exceeded maximum number of retransmissions
13	Received too many broadcast disassociations
14	Received too many broadcast deauthentications
15	Previous transition failed
16	Low RSSI
17	Roam from a non-IEEE 802.11 system
18	Transition due to received BSS Transition Request frame
19	Preferred BSS transition candidate list included
20	Leaving ESS
21–255	Reserved

The Transition Result field contains the result of the attempted transition and is one of the status codes specified in Table 8-37 in 8.4.1.9.

The Source RCPI field indicates the received channel power of the most recently measured frame from the Source BSSID before the STA reassociates to the Target BSSID. The Source RCPI is a logarithmic function of the received signal power, as defined in the RCPI measurement subclause for the PHY Type.

The Source RSNI field indicates the received signal-to-noise indication of the most recently measured frame from the Source BSSID before the STA reassociates to the Target BSSID. The Source RSNI is a logarithmic function of the signal-to-noise ratio, as defined in 8.4.2.43.

The Source BSSID, Source RCPI, and Source RSNI fields are set to 0 if the transition is initiated by an Association Request frame.

The Target RCPI field indicates the received channel power of the first measured frame just after the STA reassociates to the Target BSSID. If association with the Target BSSID failed, the Target RCPI field indicates the received channel power of the most recently measured frame from the Target BSSID. The Target RCPI is a logarithmic function of the received signal power, as defined in the RCPI measurement subclause for the PHY Type.

The Target RSNI field indicates the received signal-to-noise indication of the first measured frame just after the STA reassociates to the Target BSSID. If association with the Target BSSID failed, the Target RCPI field indicates the received signal-to-noise indication of the most recently measured frame from the Target BSSID. The Target RSNI is a logarithmic function of the signal-to-noise ratio, as defined in 8.4.2.43.

8.4.2.70.3 RSNA event report

The format of the Event Report field corresponding to an RSNA event report is shown in Figure 8-283.

	Target BSSID	Authentication Type	EAP Method	RSNA Result	RSNE
Octets:	6	4	1 or 8	2	variable

Figure 8-283—Event Report format for RSNA event

The Target BSSID field contains the 6-octet BSSID address of the AP accepting the authentication attempt.

The Authentication Type field contains the Authentication type in use at the time of the authentication attempt and is one of the AKM suite selectors defined in Table 8-101 in 8.4.2.27.3.

When the Authentication Type field is the value of either 00-0F-AC:1 (Authentication negotiated over IEEE 802.1X or using PMKSA caching as defined in 11.5.9.3) or 00-0F-AC:3 (AKM suite selector for Fast BSS Transition as defined in 11.5.3), the EAP Method field contains the IANA assigned EAP type. The EAP type contains either the legacy type (1 octet) or the expanded type (1 octet type = 254, 3-octet Vendor ID, 4-octet Vendor-Type). The EAP Method field is 0 otherwise. The EAP Method field is a single octet set to 0 otherwise.

The RSNA Result field contains the result of the RSNA establishment attempt and is one of the status codes specified in Table 8-37 in 8.4.1.9.

The RSNE field contains the entire contents of the negotiated RSNE at the time of the authentication attempt. The maximum length of the RSNE field is less than the maximum length of an RSNE, as defined in 8.4.2.27. If the length of the RSNE included here exceeds the maximum length of the RSNE field, the RSNE shall be truncated to the maximum length allowed for the RSNE field.

8.4.2.70.4 Peer-to-Peer Link event report

The format of the Event Report field corresponding to a Peer-to-Peer Link event report is shown in Figure 8-284.

	Peer STA/ BSSID Address	Operating Class	Channel Number	STA Tx Power	Connection Time	Peer Status
Octets:	6	1	1	1	3	1

Figure 8-284—Event Report format for Peer-to-Peer Link event

A Peer-to-Peer link is defined to be either a Direct Link within a QoS BSS, a TDLS, or a STA-to-STA communication in an IBSS.

The Peer STA/BSSID Address field contains a 6-octet MAC address. If this event is for a Peer-to-Peer link in an infrastructure BSS, this field contains the MAC address of the peer STA. If this event is for a Peer-to-Peer link in an IBSS, this field contains the BSSID of the IBSS.

The Operating Class field indicates the channel set of the Peer-to-Peer link. Valid values of the Operating Class are shown in Annex E.

The Channel Number field indicates the Peer-to-Peer channel number of the Peer-to-Peer link. The Channel Number is defined within a Operating Class as shown in Annex E.

The STA Tx Power field indicates the target transmit power at the antenna in dBm with a tolerance of ± 5 dB of the lowest basic rate of the reporting STA.

The Connection Time field contains the connection time in seconds. If the Peer Status is 0, this field indicates the duration of the Direct Link. If the Peer Status is 1, this field indicates the time difference from the time the Direct Link was established to the time at which the reporting STA generated the event report. If the Peer Status is 2, this field indicates the duration of the IBSS membership. If the Peer Status is 3, this field indicates the time difference from the time the STA joined the IBSS to the time at which the reporting STA generated the event report. See 10.23.2.4.

The Peer Status field indicates the Peer link connection status as indicated in Table 8-139.

Table 8-139—Peer Status definitions

Peer Status	Description
0	Direct Link terminated
1	Direct Link active
2	IBSS membership terminated
3	IBSS membership active
4–255	Reserved

8.4.2.70.5 WNM Log event report

The format of the Event Report field corresponding to a WNM Log event report is shown in Figure 8-285.

	WNM Log Msg
Octets:	variable

Figure 8-285—Event Report format for WNM Log event

The WNM Log Msg field contains the entire syslog message, consisting of the PRI, HEADER, and MSG portion of a WNM Log message as described in IETF RFC 3164-2001. The TAG field of the MSG portion of the message is a 17 octet string containing the ASCII representation of the STA MAC address using hexadecimal notation with colons between octets. The octet containing the individual/group bit occurs last, and that bit is in the least significant position within that octet. See 10.23.2.5.

8.4.2.70.6 Vendor Specific event report

The Event Report field corresponding to Vendor Specific event report contains zero or more Vendor Specific subelements. The Vendor Specific subelement has the same format as the Vendor Specific element (see 8.4.2.28).

8.4.2.71 Diagnostic Request element

8.4.2.71.1 Diagnostic Request definition

The Diagnostic Request element contains a request that the receiving STA undertake the specified diagnostic action. The format of the Diagnostic Request element is shown in Figure 8-286.

	Element ID	Length	Diagnostic Token	Diagnostic Request Type	Diagnostic Timeout	Diagnostic Information Subelements (optional)
Octets:	1	1	1	1	2	variable

Figure 8-286—Diagnostic Request element format

The Element ID field is equal to the Diagnostic Request value in Table 8-54.

The value of the Length field is 4 plus the length of the Diagnostic Information Subelements field. The minimum value of the Length field is 4 (based on a minimum length for the Diagnostic Information Subelements field of 0 octets).

The Diagnostic Token field is a number that is unique among the Diagnostic Request elements sent to each destination MAC address for which a corresponding Diagnostic Report element has not been received.

The Diagnostic Request Type field is a number that identifies a type of diagnostic request. The values of Diagnostic Request Types are shown in Table 8-140.

Table 8-140—Diagnostic Request/Report Type definitions

Name	Diagnostic Type values
Cancel Diagnostic Request	0
Manufacturer Information STA Report	1
Configuration Profile	2
Association Diagnostic	3
IEEE 802.1X Authentication Diagnostic	4
Reserved	5–220
Vendor Specific	221
Reserved	222–255

The Diagnostic Timeout field contains the time, in seconds, after which no response is returned.

The Diagnostic Information Subelements field contains zero or more diagnostic information subelements depending on the specific Diagnostic Request Type, as defined in 8.4.2.71.2 through 8.4.2.71.4.

The Cancel Diagnostic Request, Manufacturer Information STA Report, and Configuration Profile Diagnostic Request elements carry no Diagnostic Information subelements.

The Diagnostic Request element is included in a Diagnostic Request frame, as described in 8.5.14.4. The use of Diagnostic Request element and frames is described in 10.23.3.

8.4.2.71.2 Association Diagnostic request

The Diagnostic Information Subelements field corresponding to an Association Diagnostic request type is shown in Table 8-141. The corresponding Diagnostic Information subelements are defined in 8.4.2.71.5.

Table 8-141—Association Diagnostic request contents

Order	Information subelement
1	AP Descriptor
2	Profile ID

8.4.2.71.3 IEEE 802.1X Authentication Diagnostic request

The Diagnostic Information Subelements field corresponding to an IEEE 802.1X Authentication Diagnostic request type is shown in Table 8-142. The corresponding Diagnostic Information subelements are defined in 8.4.2.71.5.

Table 8-142—IEEE 802.1X Authentication Diagnostic request contents

Order	Information subelement
1	AP Descriptor
2	EAP Method
3	Credential Type
4	Profile ID

8.4.2.71.4 Vendor Specific diagnostic request

The Diagnostic Information Subelements field corresponding to a Diagnostic Request element of type Vendor Specific diagnostic request contains zero or more Vendor Specific subelements. The Vendor Specific subelements have the same format as their corresponding elements (see 8.4.2.28).

8.4.2.71.5 Diagnostic Information subelement descriptions

The following text describes the various subelements that may be included in Diagnostic Information Subelements field of a Diagnostic Request element (8.4.2.71) or a Diagnostic Report element (8.4.2.72). The format of a Diagnostic Information subelement is shown in Figure 8-287.

	Diagnostic Information Subelement ID	Length	Diagnostic Information Subelement Contents
Octets:	1	1	variable

Figure 8-287—Diagnostic Information subelement format

The Diagnostic Information Subelement ID field indicates the Diagnostic Information subelement ID and is any allocated value in Figure 8-143.

Table 8-143—Diagnostic Information subelement ID values

Identifier	Subelement name	Length (in octets)
0	Credential Type	3
1	AKM Suite	6
2	AP Descriptor	10
3	Antenna Type	4 to 251
4	Cipher Suite	6
5	Collocated Radio Type	3
6	Device Type	3
7	EAP Method	3 to 10
8	Firmware Version	3 to 251
9	MAC Address	8
10	Manufacturer ID String	3 to 251
11	Manufacturer Model String	3 to 251
12	Manufacturer OI	5 or 7
13	Manufacturer Serial Number String	3 to 251
14	Power Save Mode	6
15	Profile ID	3
16	Supported Operating Classes	3 to 251
17	Status Code	4
18	SSID	4 to 36
19	Tx Power Capability	3 to 251
20	Certificate ID	3 to 251
21–220	Reserved	
221	Vendor Specific	3 to 251
221–255	Reserved	

The Length field is the length in octets of the Diagnostic Information Subelement Contents field.

The values of the Diagnostic Information Subelement Contents field are described in the following paragraphs.

The format for the Credential Type subelement is shown in Figure 8-288.

	Subelement ID	Length	Credential Values
Octets:	1	1	variable

Figure 8-288—Credential Type subelement format

The Credentials Values field indicates one or more types of credential. Each value is chosen from those shown in Table 8-144.

Table 8-144—Credentials values

Value	Description
0	None
1	Preshared key
2	Username and password
3	X.509 certificate
4	Other certificate
5	One time password
6	Token
7–255	Reserved

The format for the AKM Suite subelement is shown in Figure 8-289.

	Subelement ID	Length	OUI	AKM Suite
Octets:	1	1	3	1

Figure 8-289—AKM Suite subelement format

The OUI and AKM Suite fields identify the authentication method, and the AKM suite selector is one of the values in Table 8-101 in 8.4.2.27.3.

The format of the AP descriptor subelement is described in Figure 8-290.

	Subelement ID	Length	BSSID	Operating Class	Channel Number
Octets:	1	1	6	1	1

Figure 8-290—AP Descriptor subelement format

The BSSID field is a 6-octet field, as described in 8.2.4.3.4, that identifies the BSS indicated in the AP Descriptor subelement.

The Operating Class field contains an enumerated value from Annex E specifying the frequency band in which the Channel Number is valid.

The Channel Number field indicates the current operating channel of the AP identified by the BSSID in the AP Descriptor.

The format for the Antenna Type subelement is shown in Figure 8-291.

	Subelement ID	Length	Antenna Count	Antenna Gain	Antenna Type
Octets:	1	1	1	1	variable

Figure 8-291—Antenna Type subelement format

The Antenna Count field contains a 1-octet field indicating the number of antennas of the indicated Antenna Type and Antenna Gain pair. The Antenna Count field value 0 is reserved.

The Antenna Gain field contains the peak gain, in dBi, of the antenna.

The Antenna Type field contains an ASCII string (truncated to 251 octets if required) describing the manufacturer's type information (i.e., a series of letters/numbers) of the antenna(s) connected to the wireless adapter. The Antenna Type field does not change based on different modes of operation of the antenna(s), as may be identified by the Antenna ID field (see 8.4.2.42). This string is not null terminated.

NOTE—Beamforming antennas might have several Antenna IDs, depending on antenna bearing.

The format for the Cipher Suite subelement is shown in Figure 8-292.

	Subelement ID	Length	OUI	Suite Type
Octets:	1	1	3	1

Figure 8-292—Cipher Suite subelement format

The OUI and Suite Type fields identify the cipher suite, and the cipher suite selector is one of the values in Table 8-99.

The format for the Collocated Radio Type subelement is shown in Figure 8-293.

	Subelement ID	Length	Collocated Radio Type
Octets:	1	1	1

Figure 8-293—Collocated Radio Type subelement format

The Collocated Radio Type subelement contains a 1-octet field indicating the type of collocated radio, and is one of the values in Table 8-145. The method that a STA uses to obtain the information on the Collocated Radio is out of the scope of this standard.

Table 8-145—Collocated Radio Type

Collocated Radio Type	Value
Reserved	0
Cellular	1
Cordless	2
GPS	3
IEEE 802.11	4
IEEE 802.15	5
IEEE 802.16	6
IEEE 802.20	7
IEEE 802.22	8
Digital Audio Broadcasting	9
Digital Video Broadcasting	10
Reserved	11–255

The Device Type subelement reports the type of device in which the IEEE 802.11 STA resides. The format of the Device Type subelement is shown in Figure 8-294.

	Subelement ID	Length	Device Type
Octets:	1	1	1

Figure 8-294—Device Type subelement format

The Device Type field is a 1-octet field indicating the category of device.[25] The numerical assignment to each device type category is defined in Table 8-146.

Table 8-146—Device Type definitions

Device Type	Value
Reserved	0
Reference Design	1
Access Point or Wireless Router for Home or Small Office	2
Enterprise Access Point	3
Cable, DSL or Other Broadband Gateway	4

[25] The category of device is based on the Wi-Fi Alliance® category definitions found at http://www.wi-fi.org/knowledge_center/insist-on-wifi-certified.

Table 8-146—Device Type definitions *(continued)*

Device Type	Value
Digital Still Camera	5
Portable Video Camera	6
Networked Web Camera	7
Digital Audio—Stationary	8
Digital Audio—Portable	9
Set-Top Box, Media Extender, Media Server (includes players & recorders)	10
Display Device (television, monitor, picture frame)	11
Game Console or Game Console Adapter	12
Gaming Device —Portable	13
Media Server or Media Adapter	14
Network Storage Device	15
External Card	16
Internal Card	17
Ultra-Mobile PC	18
Notebook Computer	19
PDA (Personal Digital Assistant)	20
Printer or Print Server (includes scanner and/or fax capability)	21
Phone—Dual-Mode	22
Phone—Single-Mode	23
Smartphone—Dual-Mode	24
Smartphone—Single-Mode	25
Reserved	26-220
Other devices	221
Reserved	222–255

The format for the EAP Method subelement is shown in Figure 8-295.

	Subelement ID	Length	EAP Type	EAP Vendor ID (optional)	EAP Vendor Type (optional)
Octets:	1	1	1	0 or 3	0 or 4

Figure 8-295—EAP Method subelement format

The EAP Type field contains a value that identifies a single EAP method and is any valid IANA assigned EAP type.

The EAP Vendor ID field contains a value that identifies the EAP Vendor. The EAP Vendor ID field is included when EAP Type field is 254, and is excluded otherwise.

The EAP Vendor Type field contains a value that identifies the EAP Type as defined by the vendor. The EAP Vendor Type field is included when EAP Type field is 254, and is excluded otherwise.

The format for the Firmware Version subelement is shown in Figure 8-296.

	Subelement ID	Length	Firmware Version
Octets:	1	1	variable

Figure 8-296—Firmware Version subelement format

This Firmware Version field contains an ASCII string identifying the version of firmware currently installed on the wireless network adaptor. This string is not null terminated.

The format for the MAC Address subelement is shown in Figure 8-297.

	Subelement ID	Length	MAC Address
Octets:	1	1	6

Figure 8-297—MAC Address subelement format

This MAC Address field contains the 6-octet IEEE 802 MAC address of the STA.

The format for the Manufacturer ID String subelement is shown in Figure 8-298.

	Subelement ID	Length	ID
Octets:	1	1	variable

Figure 8-298—Manufacturer ID String subelement format

The ID field contains an ASCII string indicating the manufacturer identifier of the wireless network adaptor. This string is not null terminated.

The format for the Manufacturer Model String subelement is shown in Figure 8-299.

	Subelement ID	Length	Model
Octets:	1	1	variable

Figure 8-299—Manufacturer Model String subelement format

The Model field contains an ASCII string indicating the model of the wireless network adaptor. This string is not null terminated.

The format for the Manufacturer OI subelement is shown in Figure 8-300.

	Subelement ID	Length	OI
Octets:	1	1	3 or 5

Figure 8-300—Manufacturer OI subelement format

The OI field contains an organizationally unique identifier, as defined in 8.4.1.31.

The format for the Manufacturer Serial Number String subelement is shown in Figure 8-301.

	Subelement ID	Length	Serial Number
Octets:	1	1	variable

Figure 8-301—Manufacturer Serial Number String subelement format

The Serial Number field contains an ASCII string indicating the serial number of the wireless network adaptor. This string is not null terminated.

The format for the Power Save Mode subelement is shown in Figure 8-302.

	Subelement ID	Length	Power Save Mode
Octets:	1	1	4

Figure 8-302—Power Save Mode subelement format

The Power Save Mode field is a bitmap field that indicates the power save mode(s) in use by the STA, as defined in Table 8-147.

Table 8-147—Power Save Mode definition

Power Save Mode	Bit
Unknown	0
None	1
PS mode (ReceiveDTIMs=1, see 10.2.1)	2
PS mode (ReceiveDTIMs=0, see 10.2.1)	3
U-APSD (see 10.2.1.5)	4
S-APSD (see 10.2.1.5)	5
U-PSMP (see 9.26	6
S-PSMP (see 9.26)	7
SM Power Save (see 10.2.4)	8
WNM-Sleep Mode (see 10.2.1.18)	9

Table 8-147—Power Save Mode definition *(continued)*

Power Save Mode	Bit
FMS (see 10.2.1.16)	10
TIM Broadcast (see 10.2.1.17)	11
TFS (see 10.23.11)	12
TDLS Peer U-APSD (see 10.2.1.15)	13
TDLS Peer PSM (see 10.2.1.14)	14
Reserved	15–31

The format for the Profile ID subelement is shown in Figure 8-303.

	Subelement ID	Length	Profile ID
Octets:	1	1	1

Figure 8-303— Profile ID subelement format

The Profile ID field contains a unique identifier for referencing a configuration profile available on a device. The value of the identifier is uniquely associated to a single configuration profile on the device sending the identifier.

The format for the Supported Operating Classes subelement is shown in Figure 8-304.

	Subelement ID	Length	Supported Operating Classes Element
Octets:	1	1	variable

Figure 8-304—Supported Operating Classes subelement format

The Supported Operating Classes Element field contains the Supported Operating Classes element, as defined in 8.4.2.56.

The format for the Status Code subelement is shown in Figure 8-305.

	Subelement ID	Length	Status Code
Octets:	1	1	2

Figure 8-305—Status Code subelement format

The Status Code field contains the final IEEE 802.11 Status code, as defined in Table 8-37 in 8.4.1.9, received at the end of the applicable operation.

The format for the SSID subelement is shown in Figure 8-306.

	Subelement ID	Length	SSID
Octets:	1	1	0 to 32

Figure 8-306—SSID subelement format

The SSID field contains a Service Set Identifier with a maximum length of 32 octets, as defined in 8.4.2.2.

The format for the Tx Power Capability subelement is shown in Figure 8-307.

	Subelement ID	Length	Tx Power Mode	Tx Power
Octets:	1	1	1	variable

Figure 8-307—Tx Power Capability subelement format

The Tx Power Mode field identifies the transmit power mode of the non-AP STA and is one of the values in Table 8-148.

Table 8-148—Tx Power Modes

Tx Power Mode	Value
Discrete	0
Range	1
Reserved	2–255

The Tx Power field indicates the target transmit power level(s) at the antenna(s), where the actual power is within ±5 dB to the target. Each transmit power level is encoded in a single octet as a twos complement value in dBm, rounded to the nearest integer. If the Tx Power Mode field is 0 then the Tx Power field contains one or more transmit power levels in increasing numerical order. If the Tx Power Mode field is 1, the Tx Power field contains the STA's minimum and nonzero maximum transmit power levels, in that order.

The format for the Certificate ID subelement is shown in Figure 8-308.

	Subelement ID	Length	Certificate ID
Octets:	1	1	variable

Figure 8-308—Certificate ID subelement format

The Certificate ID field contains an UTF-8 string indicating the assigned identifier for the STA. This string is not null terminated. The Certificate ID typically takes the form of "WFA3991" and might be used by a receiving STA to look up the certificate assigned to that ID using a web lookup url such as http://certifications.wi-fi.org/pdf_certificate.php?cid= WFA3991.

The Vendor Specific subelements have the same format as their corresponding elements (see 8.4.2.28).

8.4.2.72 Diagnostic Report element

8.4.2.72.1 Diagnostic Report definition

The Diagnostic Report element contains a Diagnostic report. The format of the Diagnostic Report element is shown in Figure 8-309.

	Element ID	Length	Diagnostic Token	Diagnostic Report Type	Diagnostic Status	Diagnostic Information Subelements
Octets:	1	1	1	1	1	variable

Figure 8-309—Diagnostic Report element format

The Element ID field is equal to the Diagnostic Report value in Table 8-54.

The value of the Length field is 3 plus the length of the Diagnostic Information Subelements field. The minimum value of the Length field is 3.

The Diagnostic Token field is the Diagnostic Token in the corresponding Diagnostic Request element.

The Diagnostic Report Type field is a number that identifies the Diagnostic report. The Diagnostic Report Types are defined in Table 8-140.

The Diagnostic Status field is a value in Table 8-137 (see 8.4.2.70), indicating the STA's response to the Diagnostic Request indicated by the Diagnostic Token.

The Diagnostic Information Subelements field contains the results of the diagnostic request.

The Diagnostic Information Subelements field contains the diagnostic subelement values, defined in 8.4.2.71.5 corresponding to the Diagnostic Report Type field value.

The Diagnostic Report element is included in a Diagnostic Report frame, as described in 8.5.14.5. The use of Diagnostic Report element and frames is described in 10.23.3.

8.4.2.72.2 Manufacturer Information STA Report

The contents of the Diagnostic Information Subelements field of a Diagnostic Report element of type Manufacturer Information STA Report is shown in Table 8-149. The corresponding Diagnostic Information subelements are defined in 8.4.2.71.5.

Table 8-149—Manufacturer Information STA Report contents

Order	Information subelement
1	Manufacturer OI
2	Manufacturer ID string
3	Manufacturer model string
4	Manufacturer serial number string
6	Firmware Version

Table 8-149—Manufacturer Information STA Report contents *(continued)*

Order	Information subelement
7	Antenna Type
8	Collocated Radio Type
9	Device Type
10	Certificate ID

8.4.2.72.3 Configuration Profile report

The contents of the Diagnostic Information Subelements field of a Diagnostic Report element of type Configuration Profile is shown in Table 8-150. The corresponding Diagnostic Information subelements are defined in 8.4.2.71.5.

Table 8-150—Configuration Profile report contents

Order	Information subelement
1	Profile ID
2	Supported Operating Classes
3	Tx Power
4	Cipher Suite
5	AKM Suite
6	EAP Method
7	Credential Type
8	SSID
9	Power Save Mode

8.4.2.72.4 Association Diagnostic report

The contents of the Diagnostic Information Subelements field of a Diagnostic Report element of type Association Diagnostic is shown in Table 8-151. The corresponding Diagnostic Information subelements are defined in 8.4.2.71.5.

Table 8-151—Association Diagnostic report contents

Order	Information subelement
1	AP Descriptor
2	Status Code

8.4.2.72.5 IEEE 802.1X Authentication Diagnostic report

The contents of the Diagnostic Information Subelements field of a Diagnostic Report element of type IEEE 802.1X Authentication Diagnostic is shown in Table 8-152. The corresponding Diagnostic Information subelements are defined in 8.4.2.71.5.

Table 8-152—IEEE 802.1X Authentication Diagnostic report contents

Order	Information subelement
1	AP Descriptor
2	EAP Method
3	Credential Type
4	Status Code

8.4.2.72.6 Vendor Specific diagnostic report

The contents of the Diagnostic Information subelements field of a Diagnostic Report element of type Vendor Specific contains zero or more Vendor Specific subelements that have the same formats as Vendor Specific elements in 8.4.2.28.

8.4.2.73 Location Parameters element

8.4.2.73.1 Location Parameters definition

The Location Parameters element is used for location services. The format of this element is shown in Figure 8-310.

	Element ID	Length	Location Subelements
Octets:	1	1	variable

Figure 8-310—Location Parameters element format

The Element ID field is equal to the Location Parameters value in Table 8-54.

The value of the Length field is the length of the Location Subelements field.

The Location Subelements field contains one or more Location subelements described in Table 8-153.

Table 8-153—Location subelements

Identifier	Subelement name	Length (in octets)
1	Location Indication Parameters (see 8.4.2.73.2)	18
2	Location Indication Channels (see 8.4.2.73.3)	4–254
3	Location Status (see 8.4.2.73.4)	4

Table 8-153—Location subelements *(continued)*

Identifier	Subelement name	Length (in octets)
4	Radio Information (see 8.4.2.73.5)	7
5	Motion (see 8.4.2.73.6)	10
6	Location Indication Broadcast Data Rate (see 8.4.2.73.7)	4
7	Time of Departure (see 8.4.2.73.8)	9
8	Location Indication Options (see 8.4.2.73.9)	4
9–220	Reserved	
221	Vendor Specific (see 8.4.2.28)	5–254
222–255	Reserved	

The Location Parameters element is included in Location Configuration Request frames, as described in 8.5.14.6; Location Configuration Response frames, as described in 8.5.14.7; and Location Track Notification frames, as described in 8.5.8.17. The use of the Location Parameters element is described in 10.23.4.

The Vendor Specific subelements have the same format as their corresponding elements (see 8.4.2.28). Multiple Vendor Specific subelements are optionally present in the list of Optional subelements.

8.4.2.73.2 Location Indication Parameters subelement

The Location Indication Parameters subelement contains STA location reporting characteristics. The format of the Location Indication Parameters subelement is shown in Figure 8-311.

	Subelement ID	Length	Indication Multicast Address	Report Interval Units	Normal Report Interval	Normal Number of Frames per Channel
Octets:	1	1	6	1	2	1

	In-Motion Report Interval	In-Motion Number of Frames per Channel	Burst Inter-frame Interval	Tracking Duration	ESS Detection Interval
Octets:	2	1	1	1	1

Figure 8-311—Location Indication Parameters subelement

The Subelement ID field contains the value for Location Indication Parameters as defined in Table 8-153.

The Length field is 16.

The Indication Multicast Address field specifies the destination address to which the Location Track Notification frames are sent to in a non-IBSS network. The value of this field is a locally administered group address formed according to the procedure defined in 10.23.4.1. The field is reserved when Location Track Notifications are transmitted in an IBSS.

The Report Interval Units field contains the units used for the Normal Report Interval field and In-Motion Report Interval field, as indicated in Table 8-154.

Table 8-154—Report Interval Units field

Report Interval Units	Description
0	Hours
1	Minutes
2	Seconds
3	Milliseconds
4–255	Reserved

The Normal Report Interval is the time interval, expressed in the units indicated in the Report Interval Units field, at which the reporting STA is expected to transmit one or more Location Track Notification frames if either dot11MgmtOptionMotionDetectionActivated is false or the STA is stationary. The STA does not transmit Location Track Notification frames when the Normal Report Interval is 0.

The Normal Number of Frames per Channel is the number of Location Track Notification frames per channel sent or expected to be sent by the STA at each Normal Report Interval.

Motion is the act or process of moving, or a particular action or movement relative to the point at which the STA is configured to send Location Track Notification frames. Motion might be detected using one of the following criteria:

- Detection of speed that is greater or equal to 0.5 m/sec.
- Detection of movement or vibration, for example by a ball-in-tube sensor or accelerometer or other means.

The exact criteria and mechanism to detect motion is out of scope for this standard.

The In-Motion Report Interval is the time interval, expressed in the units indicated in the Report Interval Units field, at which the STA reports its location by sending a Location Track Notification frame when the reporting STA is in motion. If dot11MgmtOptionMotionDetectionActivated is false, this field is 0.

The In-Motion Number of Frames per Channel is the number of Location Track Notification frames per channel sent or expected to be sent by the STA at each In-Motion Report Interval. If dot11MgmtOptionMotionDetectionActivated is false, this field is 0.

The Burst Inter-frame Interval is the target time interval, expressed in milliseconds, between the transmissions of each of the Normal or In-Motion frames on the same channel. The Burst Inter-frame interval value is 0 to indicate that frames are transmitted with no target inter-frame delay.

The Tracking Duration is the amount of time, in minutes, that a STA sends the Location Track Notification frames. The duration starts as soon as the STA sends a Location Configuration Response frame with a Location Status value of Success. If the Tracking Duration value is a nonzero value the STA sends Location Track Notification Frames, based on the Normal and In-Motion Report Interval field values, until the duration ends. If the Tracking Duration is 0 the STA continuously sends Location Track Notification frames as defined by Normal and In-Motion Report Interval field values until transmission is terminated based on the procedures detailed in 10.23.4.2.

The ESS Detection Interval is the periodicity, in minutes, that a STA checks for beacons transmitted by one or more APs belonging to the same ESS that configured the STA. If no beacons from the ESS are received for this period, the STA terminates transmission of Location Track Notification frames, as described in the procedures detailed in 10.23.4.2. The ESS Detection Interval field is not used when the ESS Detection Interval field value is 0.

8.4.2.73.3 Location Indication Channels subelement

The Location Indication Channels subelement contains location reporting channel information. The format of the Location Indication Channels subelement format is shown in Figure 8-312.

			one or more entries
	Subelement ID	Length	Channel Entry
Octets:	1	1	2xn

Figure 8-312—Location Indication Channels subelement

The Subelement ID field contains the value for Location Indication Channels as defined in Table 8-153.

The Length field is 2*n*, where *n* indicates the total number of Channel Entry subelements contained in the element.

The Channel Entry field includes one or more Operating Class and Channel pair. The format Channel Entry field is shown in Figure 8-313.

	Operating Class	Channel
Octets:	1	1

Figure 8-313—Channel Entry field format

The Operating Class field each indicates the frequency band on which a STA transmits Location Track Notification frames. All Operating Class field values are for the country specified in the Beacon frame. Valid values of the Operating Class field are defined in Annex E.

The Channel field includes the channel numbers on which a STA sends or an ESS expects to receive Location Track Notification frames. Valid values of the Channel field are defined in Annex E.

8.4.2.73.4 Location Status subelement

The Location Status subelement provides the result of a Location Request or Location Configuration Request frame. The format of the Location Status subelement is shown in Figure 8-314.

	Subelement ID	Length	Config Subelement ID	Status
Octets:	1	1	1	1

Figure 8-314—Location Status subelement

The Subelement ID field contains the value for Location Status as defined in Table 8-153.

The Length field is 2.

The Config Subelement ID field is a specific Location Parameters subelement ID transmitted in a Location Configuration Request frame as defined in Table 8-153.

A Location Status subelement is included for each configuration subelement in the Location Configuration Request frame, except when all status values are the same. When all status values are the same, one Location Status subelement is included with the Config subelement ID set to 0 in the Location Configuration Response frame.

The Status field identifies the result of the Location Request frame and is one of the values in Table 8-137.

8.4.2.73.5 Radio Information subelement

The Radio Information subelement contains radio information. The format of the Radio Information subelement is shown in Figure 8-315.

	Subelement ID	Length	Transmit Power	Antenna ID	Antenna Gain	RSNI	RCPI
Octets:	1	1	1	1	1	1	1

Figure 8-315—Radio Information subelement

The Subelement ID field contains the value for Radio Information as defined in Table 8-153.

The Length field is 5.

The Transmit Power field is the transmit power used to transmit the current Location Track Notification frame containing the Location Parameters element with the Radio Information subelement and is a signed integer, 1 octet in length, reported in dBm. A value of –128 indicates that the transmit power is unknown. The tolerance for the transmit power value reported in the Radio Information subelement is ± 5 dB. This tolerance is defined as the maximum possible difference, in decibels, between the reported power value and the total transmitted power across all antennas of the STA, which are measured when transmitting Location Request frames.

The Antenna ID field is the identifying number for the antenna used to transmit the Location Request frame. The Antenna ID is defined in 8.4.2.42.

The Antenna Gain field is the antenna gain of the antenna (or group of antennas) over which the Location Track Notification frame is transmitted and is a signed integer, 1 octet in length, reported in dB. A value of –128 indicates that the antenna gain is unknown.

The RSNI field contains the RSNI value measured against the most recently received Location Configuration Request frame requesting that a Radio Information subelement be included in the Location

Track Notification frame. The RSNI value is defined in 8.4.2.43. A value of 255 indicates that the RSNI value is unknown or is not used.

The RCPI field contains the RCPI value measured against the most recently received Location Configuration Request frame requesting that a Radio Information subelement be included in the Location Track Notification frame. The RCPI value is defined in 8.4.2.40. A value of 255 indicates that the RCPI value is unknown or is not used.

8.4.2.73.6 Motion subelement

The Motion subelement contains motion information. The format of the Motion subelement is shown in Figure 8-316.

	Subelement ID	Length	Motion Indicator	Bearing	Speed Units	Horizontal Speed	Vertical Speed
Octets:	1	1	1	2	1	2	2

Figure 8-316—Motion subelement

The Subelement ID field contains the value for Motion as defined in Table 8-153.

The Length field is 8.

The Motion Indicator field is defined in Table 8-155. The mechanism that a STA uses to determine the value or transitions from one value to another of the Motion Indicator field is beyond the scope of the standard.

Table 8-155—Motion Indicator field

Motion Indicator value	Description
0	Stationary: the device is stationary and not in motion.
1	Start of motion: the device was stationary and is now in motion.
2	In motion: the device is and has been in motion.
3	End of motion: the device was in motion and is now stationary.
4	Unknown: information related to motion is unknown.
5–255	Reserved

The Bearing field, defined by a 2-octet unsigned integer, specifies the direction that the STA is traveling with relation to true north, increasing clockwise, measured in degrees from 0 degree to 359 degrees. If the Bearing value is unknown, the field is 65 535.

The Speed Units field contains the units for both Horizontal and Vertical Speed field, as defined in Table 8-156.

Table 8-156—Speed Units

Speed Units Value	Description
0	Centimeters per second
1	Meters per second
2–255	Reserved

The Horizontal Speed field contains the horizontal speed of the STA expressed in the units indicated in the Speed Units field. If the Horizontal Speed value is unknown, the field is 65 535.

The Vertical Speed field is a twos complement signed integer indicating the vertical speed of the STA expressed in the units indicated in the Speed Units field. If the Vertical Speed value is unknown or greater than 32 766, the field is 32 767. If the Vertical Speed value is less than –32 767, the field is –32 768.

The Motion subelement field values are valid at the time of transmission of the Location Track Notification frame containing the subelement.

8.4.2.73.7 Location Indication Broadcast Data Rate subelement

The Location Indication Broadcast Data Rate subelement contains location reporting transmission rate information. The format of the Location Indication Broadcast Data Rate subelement format is shown in Figure 8-317.

	Subelement ID	Length	Broadcast Target Data Rate
Octets:	1	1	4

Figure 8-317—Location Indication Broadcast Data Rate subelement

The Subelement ID field contains the value for Location Indication Broadcast Data Rate as defined in Table 8-153.

The value of the Length field is 4.

The Broadcast Target Data Rate field specifies the target data rate at which the STA transmits Location Track Notification frames. The Broadcast Target Data Rate field format is specified by the Rate Identification field defined in 8.4.1.32. A value of 0 indicates the STA transmits Location Track Notification frames at a rate chosen by the STA transmitting the Location Track Notification frames.

8.4.2.73.8 Time of Departure subelement

The Time of Departure subelement contains time of departure information for the Location Track Notification frame including the subelement. The format of the Time of Departure subelement is shown in Figure 8-318.

	Subelement ID	Length	TOD Timestamp	TOD RMS	TOD Clock Rate
Octets:	1	1	4	2	2

Figure 8-318—Time of Departure subelement

The Subelement ID field contains the value for Time of Departure as defined in Table 8-153.

The value of the Length field is 8.

The TOD Timestamp field carried within a frame specifies when the first frame energy is sent by the transmitting port in units equal to 1/TOD Clock Rate, where the TOD Clock Rate is specified in the TOD Clock Rate field. The reported TOD timestamp value is determined from the TIME_OF_DEPARTURE parameter within the PHY-TXSTART.confirm primitive.

The TOD RMS field specifies the RMS time of departure error in units equal to 1/TOD Clock Rate, where the TOD Clock Rate is specified in the TOD Clock Rate field, where the time of departure error equals the difference between the TOD Timestamp field and the time of departure measured by a reference entity using a clock synchronized to the start time and mean frequency of the local PHY entity's clock. TOD RMS field is determined from aTxPmdTxStartRMS in units equal to 1/TOD Clock Rate, where the TOD Clock Rate is specified in the TOD Clock Rate field.

The TOD Clock Rate field contains the clock rate used to generate the TOD timestamp value reported in the TOD Timestamp field, and it is specified in units of MHz.

8.4.2.73.9 Location Indication Options subelement

The Location Indication Options subelement contains the options for the STA when transmitting the Location Track Notification frame. The format of the Location Indication Options subelement is shown in Figure 8-319.

	Subelement ID	Length	Options Used	Indication Parameters
Octets:	1	1	1	variable

Figure 8-319—Location Indication Options subelement

The Subelement ID field contains the value for Location Indication Options as defined in Table 8-153.

The Length field is 1 plus the length of each Indication Parameter included.

The Options Used field specifies which Indication Parameter fields in the Location Indication Options subelement are used. The format of the Options Used field is shown in Figure 8-320.

	B0	B1 B7
	Beacon Measurement Mode Used	Reserved
Bits:	1	7

Figure 8-320—Options Used field format

The Indication Parameters field defines a sequence of optional fields that are included in the Location Indication Options subelement based on the Options Used field value. The value of the Indication Parameters field is defined in Table 8-157.

Table 8-157—Indication Parameter values

Order	Field length	Field	Description
1	1	Beacon Measurement Mode	The Beacon Measurement Mode field is the mode of beacon measurement, as defined in Table 8-64. The results of the beacon measurement are included in the Location Track Notification frame, as described in 8.5.8.17 and 10.23.4.2.
2–8		Reserved	

8.4.2.74 Nontransmitted BSSID Capability element

The format of the Nontransmitted BSSID Capability element is shown in Figure 8-321.

	Element ID	Length	Nontransmitted BSSID Capability
Octets:	1	1	2

Figure 8-321—Nontransmitted BSSID Capability element format

The Element ID field is equal to the Nontransmitted BSSID Capability value in Table 8-54.

The value of the Length field is 2.

The Nontransmitted BSSID Capability field contains the Capability information field of the BSS.

The Nontransmitted BSSID Capability element is included in the Nontransmitted BSSID profile subelement of the Multiple BSSID element defined in 8.4.2.48. The use of the Multiple BSSID element is described in 10.11.14 and Nontransmitted BSSID Advertisement procedures are described in 10.1.3.6.

8.4.2.75 SSID List element

The format of the SSID List element is shown in Figure 8-322.

	Element ID	Length	SSID List
Octets:	1	1	variable

Figure 8-322—SSID List element format

The Element ID field is equal to the SSID List value in Table 8-54.

The value of the Length field is the length of the SSID list (variable) in octets.

The SSID List field is a list of SSID elements, each including the element ID, length field and SSID information field (see 8.4.2.2) for which the STA is requesting information.

The SSID List element is included in Probe Request frames, as described in 8.3.3.9. The use of the SSID List element and frames is described in 10.1.4.

8.4.2.76 Multiple BSSID-Index element

The format of the Multiple BSSID-Index element is shown in Figure 8-323.

	Element ID	Length	BSSID Index	DTIM Period (optional)	DTIM Count (optional)
Octets:	1	1	1	0 or 1	0 or 1

Figure 8-323—Multiple BSSID-Index element format

The Element ID field is equal to the Multiple BSSID-Index value in Table 8-54.

The value of the Length field is 1 octet when the Multiple BSSID-Index element is included in the Probe Response frame and otherwise is three octets.

The BSSID Index field is a value between 1 and $2^n – 1$ that identifies the nontransmitted BSSID, where *n* is a nonzero, positive integer value.

The DTIM Period field is the DTIM period field for the BSSID. This field is not present when the Multiple BSSID-Index element is included in the Probe Response frame.

The DTIM Count field is the DTIM count field for the BSSID. This field is not present when the Multiple BSSID-Index element is included in the Probe Response frame.

The Multiple BSSID-index element is included in the nontransmitted BSSID profile element, as described in 10.1.3.6. The use of the Multiple BSSID element and frames is described in 10.11.14.

8.4.2.77 FMS Descriptor element

The FMS Descriptor element defines information about group addressed BUs buffered at the AP. It is present in the Beacon frames when dot11MgmtOptionFMSActivated is true. The format of the FMS Descriptor element is shown in Figure 8-324.

				zero or more FMS Counters	zero or more FMSIDs
	Element ID	Length	Number of FMS Counters	FMS Counters	FMSIDs
Octets:	1	1	1	n	m

Figure 8-324—FMS Descriptor element format

The Element ID field is equal to the FMS Descriptor value in Table 8-54.

The Length field is 1 if no FMS streams are accepted at the AP or is 1 + *n* + *m*, where *n* is the number of FMS Counters present and *m* indicates the number of 1-octet FMSIDs present in the element.

The Number of FMS Counters field defines the number of FMS Counters fields that are contained in the FMS Descriptor element.

The FMS Counters field contains zero or more FMS Counters. The format of the FMS Counter is shown in Figure 8-325. When one or more FMS streams are accepted at the AP, at least one FMS counter is present in the FMS Descriptor element. A maximum of eight FMS counters are permitted. The FMS counters are used by the non-AP STA to identify the DTIM beacon after which group addressed BUs assigned to a particular delivery interval are transmitted. A single FMS Counter is shared by all FMS streams that use the same delivery interval.

	B0 B2	B3 B7
	FMS Counter ID	Current Count
Bits:	3	5

Figure 8-325—FMS Counter format

The FMS Counter ID field is a 3- bit value that represents the counter ID assigned by the AP for a particular FMS stream.

The Current Count field indicates how many DTIM Beacon frames (including the current one) appear before the next DTIM Beacon frame after which the group addressed BUs assigned to a particular delivery interval are scheduled to be transmitted. The Current Count field is 0 on transmission and ignored upon reception when the FMS Counter field is included in the FMS Status subelement.

The FMSIDs field contains zero or more FMSIDs. Each FMSID is a 1-octet identifier assigned by the AP.

Inclusion of an FMSID indicates the AP has buffered BUs for the corresponding group addressed stream that is scheduled for transmission immediately after the DTIM Beacon frame.

The FMS Descriptor element is included in Beacon frames, as described in 8.3.3.2. The use of the FMS Descriptor element and frames is described in 10.2.1.16.

8.4.2.78 FMS Request element

The FMS Request element defines information about the group addressed frames being requested by the non-AP STA. The format of FMS Request element is shown in Figure 8-326.

				One or more Request subelements
	Element ID	Length	FMS Token	Request subelements
Octets:	1	1	1	variable

Figure 8-326—FMS Request element format

The Element ID field is the FMS Request value in Table 8-54.

The value of the Length field is 1 + *n*, where *n* indicates the total length of all FMS subelements contained in the element.

The FMS Token field contains a unique identifier for the FMS Stream Set that is the set of FMS subelements specified in the request. If this is a new request, then the FMS Token value is 0. Otherwise, the FMS Token value is the value assigned by the AP in the FMS Response element. The FMS Token is fixed for the lifetime of the FMS Stream Set.

The Request Subelements field contains one or more Request subelements described in Table 8-158.

Table 8-158—Request subelements

Identifier	Subelement name	Length (in octets)
0	Reserved	
1	FMS subelement	6 to 254
2–220	Reserved	
221	Vendor Specific subelement	5 to 254
222–255	Reserved	

The format of the FMS subelement is shown in Figure 8-327.

					One or more TCLAS Elements	
	Subelement ID	Length	Delivery Interval	Max Delivery Interval	Rate Identification	TCLAS Elements	TCLAS Processing Element (optional)
Octets:	1	1	1	1	4	variable	0 or 3

Figure 8-327—FMS Subelement format

The Subelement ID field is 1 to uniquely identify this subelement as the FMS subelement.

The value of the Length field is the sum of the lengths of the TCLAS element(s) plus 6 and the optional TCLAS Processing element, if present.

The Delivery Interval field defines the periodicity of stream transmission in units of DTIMs. The default value is 1. The value set to 0 indicates that requesting non-AP STA does not use the FMS stream identified by the TCLAS elements anymore.

The Max Delivery Interval field defines the maximum delivery interval the non-AP STA supports for the requested stream in units of DTIMs. The value set to 0 indicates that the non-AP STA is willing to accept any maximum delivery interval supported by the AP.

The Rate Identification field specifies the data rate as described in 8.4.1.32, at which the STA requests to receive group addressed frames. If the STA does not request a particular multicast rate, the Rate Identification field is 0.

The TCLAS Elements field contains one or more TCLAS elements to specify the traffic filter as defined in 8.4.2.33. The number of TCLAS elements is limited and the total size of the FMS Request element is less than or equal to 255 octets.

The TCLAS Processing Element field is optionally present and defines how multiple TCLAS elements are processed as defined in 8.4.2.35.

The FMS Request element is included in FMS Request frames, as described in 8.5.14.11. The use of the FMS Request element and frames is described in 10.2.1.16.

The Vendor Specific subelements have the same format as their corresponding elements (see 8.4.2.28).

8.4.2.79 FMS Response element

The FMS Response element provides information about the delivery of group addressed frames. The format of the FMS Response element is shown in Figure 8-328.

				One or more Status Subelements
	Element ID	Length	FMS Token	Status Subelements
Octets:	1	1	1	variable

Figure 8-328—FMS Response element format

The Element ID field is the FMS Response value in Table 8-54.

The Length field is 1 + *n*, where *n* indicates the total length of all FMS Status subelements contained in the element.

The FMS Token field is assigned by the AP for the set of FMS streams that share the counter identified by the FMS Counter ID maintained in the AP.

The Status Subelements field contains one or more Status subelements described in Table 8-159.

Table 8-159—Status subelements

Identifier	Subelement name	Length (in octets)
0	Reserved	
1	FMS Status subelement	15 to 254
2	TCLAS Status subelement	6 to 254
3–220	Reserved	
221	Vendor Specific subelement	5 to 254
222–255	Reserved	

The FMS Status Subelements field contains one or more FMS Status subelements. The format of the FMS Status subelement is shown in Figure 8-329.

	Subelement ID	Length	Element Status	Delivery Interval	Max Delivery Interval	FMS ID	FMS Counter	Rate Identification	Multicast Address
Octets:	1	1	1	1	1	1	1	4	6

Figure 8-329—FMS Status Subelement format

The Subelement ID field is 1 to uniquely identify this subelement as the FMS Status subelement.

The value of the Length field is 15.

The Element Status field indicates the status of STA's requested delivery interval, as indicated in Table 8-160, provided by the AP.

Table 8-160—FMS Element Status and TFS Response Status definition

Value	Description
0	Accept
1	Deny, due to request format error or ambiguous classifier.
2	Deny, due to lack of resources on AP.
3	Deny, due to requested classifier(s) matching 2 or more existing streams on different intervals.
4	Deny, by policy, requested stream or filter is not permitted to participate in the service.
5	Deny, reason unspecified.
6	Alternate Preferred, due to existing stream with different delivery interval.
7	Alternate Preferred, due to policy limits on AP.
8	Alternate Preferred, due to AP changed the delivery interval.
9	Alternate Preferred, due to AP multicast rate policy.
10	Terminate, due to AP policy change.
11	Terminate, due to lack of resources of AP.
12	Terminate, due to other FMS stream with higher priority.
13	Alternate Preferred, due to AP changed the maximum delivery interval.
14	Alternate Preferred, due to AP unable to provide requested TCLAS-based classifiers.
15–255	Reserved

The Delivery Interval field defines the minimum integer of DTIM periods between successive transmissions of frames for the stream corresponding to that FMSID.

The Max Delivery Interval field defines the maximum delivery interval the AP uses for the stream corresponding to FMSID. The value set to 0 indicates that the AP has no maximum delivery interval for the stream identified by FMSID.

The FMSID field is assigned by the AP and provides a unique identifier for this stream within the BSS.

The format of the FMS Counter field is shown in Figure 8-325.

The Rate Identification field specifies the data rate as described in 8.4.1.32 to be used for the multicast service. If the value of the Rate Identification field is 0 then the data rate is undefined.

The Multicast MAC Address field contains the MAC address of the multicast traffic to which this FMS response relates.

The format of the TCLAS Status subelement is shown in Figure 8-330.

				one or more TCLAS Elements	
	Subelement ID	Length	FMSID	TCLAS Elements	TCLAS Processing Element (optional)
Octets:	1	1	1	variable	0 or 3

Figure 8-330—TCLAS Status Subelement format

The Subelement ID field is 2 to uniquely identify this subelement as the TCLAS Status subelement.

The value of the Length field is 1 plus the sum of the lengths of the TCLAS element(s) plus the optional TCLAS Processing element, if present.

The FMSID field is assigned by the AP and provides a unique identifier for this stream within the BSS.

The TCLAS Elements field contains one or more TCLAS elements to specify the traffic filter as defined in 8.4.2.33. The number of TCLAS elements is limited and the total size of the FMS Response element is less than or equal to 255 octets.

The TCLAS Processing Element field is optionally present and defines how multiple TCLAS elements are processed as defined in 8.4.2.35.

The FMS Response element is included in FMS Response frames, as described in 8.5.14.12. The use of the FMS Response element and frames is described in 10.2.1.16.

The Vendor Specific subelements have the same format as their corresponding elements (see 8.4.2.28).

8.4.2.80 QoS Traffic Capability element

The QoS Traffic Capability element provides information about types of traffic generated by a non-AP QoS STA, and is used by a QoS AP to indicate the access categories of associated non-AP QoS STAs. The format of the QoS Traffic Capability element is shown in Figure 8-331.

	Element ID	Length	QoS Traffic Capability Bitmask/Flags	AC STA Count List (optional)
Octets:	1	1	1	Total number of nonzero bits in Bits 0–1 of QoS Traffic Capability Bitmask/Flags field

Figure 8-331—QoS Traffic Capability Element format

The Element ID field is the QoS Traffic Capability value in Table 8-54.

The value of the Length field is 1 + *n*, where *n* equals the total number of nonzero bits in Bits 0–1 of the QoS Traffic Capability Bitmask/Flags field.

The format of QoS Traffic Capability Bitmask/Flags field is defined in Table 8-161.

Table 8-161—QoS Traffic Capability Bitmask/Flags definition

Bit	Description
0	AC_VO
1	AC_VI
2	Reserved
3	Reserved
4	UP 4 Traffic
5	UP 5 Traffic
6	UP 6 Traffic
7	Reserved

Bits 0–1 serve as QoS Traffic Capability Bitmask. The bitmask indicates the AC values that have the station count specified in the following AC STA Count List. An AP sets the bit to 1 to indicate that the station count for the corresponding AC is present in the AC STA Count List field. An AP sets the bit to 0 to indicate that the station count for the corresponding AC is not present in the AC STA Count List field. A non-AP STA always sets Bits 0–1 to 0. An AP ignores Bits 0–1 on reception.

Bits 4–6 serve as QoS Traffic Capability Flags. Each of Bits 4–6 serves as a flag for a non-AP STA to indicate application requirements about the user priorities of the traffic it generates. A non-AP STA sets the bit to 1 to indicate the existence of an application that requires generation of traffic belonging to the corresponding user priority (UP). A non-AP STA sets the bit to 0 to indicate that such application does not exist. An AP always sets Bits 4–6 to 0. A non-AP STA ignores Bits 4–6 on reception.

Bits 2–3 and Bit 7 are reserved.

The AC STA Count List comprises a sequence of STA Count fields corresponding to the nonzero bits in the Bits 0–1 of the QoS Traffic Capability Bitmask/Flags field. The STA Count field is 1 octet long and contains an unsigned integer, encoded according to 8.2.2. The STA Count field specifies the number of associated QoS STAs that have indicated QoS Traffic Capability of the corresponding AC. If the number of STAs is greater than 255, the STA Count field is 255. The AC STA Count List field is present only when the QoS AP transmits the QoS Traffic Capability element.

The QoS Traffic Capability element is included in Beacon frames, as described in 8.3.3.2; Probe Response frames, as described in 8.3.3.10; Association Request frames, as described in 8.3.3.5; and Reassociation Request frames, as described in 8.3.3.7.

8.4.2.81 BSS Max Idle Period element

The BSS Max Idle Period element contains the time period a non-AP STA can refrain from transmitting frames to the AP before the AP disassociates the STA due to inactivity. The format of the BSS Max Idle Period element is shown in Figure 8-332.

	Element ID	Length	Max Idle Period	Idle Options
Octets:	1	1	2	1

Figure 8-332—BSS Max Idle Period element format

The Element ID field is the BSS Max Idle Period value in Table 8-54.

The value of the Length field is 3.

The Max Idle Period field indicates the time period during which a STA can refrain from transmitting frames to its associated AP without being disassociated. The Max Idle Period field is a 16-bit unsigned integer. The time period is specified in units of 1000 TUs. The value of 0 is reserved. A non-AP STA is considered inactive if the AP has not received a data frame or management frame of a frame exchange sequence initiated by the STA for a time period equal to or greater than the time specified by the Max Idle Period field value.

The Idle Options field indicates the options associated with the BSS Idle capability. The Idle Options field is shown in Figure 8-333.

	B0	B1-B7
	Protected Keep-Alive Required	Reserved
Bits:	1	7

Figure 8-333—Idle Options field

The Protected Keep-Alive Required bit set to 1 indicates that the STA sends an RSN protected frame to the AP to reset the Idle Timer at the AP for the STA, as defined in 10.23.12. If the Protected Keep-Alive Required bit is 0, the STA sends either an unprotected or a protected frame to the AP to reset the Idle Timer at the AP.

The BSS Max Idle Period element is included in Association Response frames, as described in 8.3.3.6, and Reassociation Response frames, as described in 8.3.3.8. The use of the BSS Max Idle Period element and frames is described in 10.23.12.

8.4.2.82 TFS Request element

The TFS Request element defines information about the traffic filters that are enabled at the AP for the requesting non-AP STA. The format of the TFS Request element is defined in Figure 8-334.

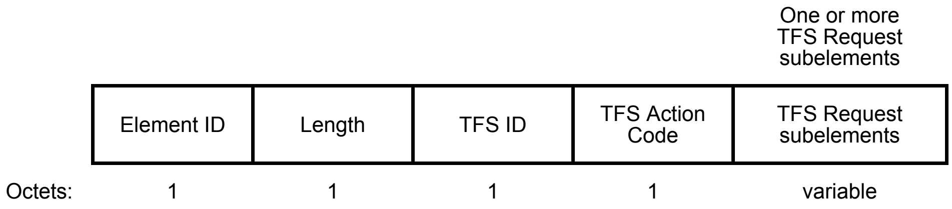


Figure 8-334—TFS Request element format

The Element ID field is equal to the TFS Request value in Table 8-54.

The Length field is 2 + *n*, where *n* indicates the total length of all TFS subelements contained in the element.

The TFS ID field is assigned by the STA and provides a unique identifier for the set of traffic filters specified in the TFS subelements.

The TFS Action Code field defines the actions taken at the AP when a frame matches a traffic filter. The functions of the bits in this field are shown in Table 8-162.

Table 8-162—TFS Action Code field values

Bit(s)	Information	Notes
0	Delete after match	Setting this field to 1 indicates the traffic filter is to be deleted when a frame matches the traffic filter. A value of 0 for this field indicates no deletion of the traffic filter.
1	Notify	Setting this field to 1 indicates the STA is to be sent a TFS Notify frame when a frame matches the traffic filter. Setting this field to 0 indicates the AP does not send a TFS Notify frame to the requesting STA.
2–7	Reserved	All other bits are reserved, and are set to 0 on transmission and ignored on reception.

The TFS Request Subelements field contains one or more TFS Request subelements described in Table 8-163.

Table 8-163—TFS Request subelements

Identifier	Subelement name	Length (in octets)
1	TFS subelement	5 to 254
221	Vendor Specific subelement	5 to 254
0, 2 to 220, 222 to 255	Reserved	

The format of the TFS subelement is shown in Figure 8-335.

			One or more TCLAS Elements	
	Subelement ID	Length	TCLAS Elements	TCLAS Processing Element (optional)
Octets:	1	1	variable	0 or 3

Figure 8-335—TFS Subelement format

The Subelement ID field uniquely identifies this subelement to be the TFS subelement. The value of this field is 1.

The value of the Length field is the sum of the lengths of the TCLAS element(s) plus the optional TCLAS Processing element, if present.

The TCLAS Elements field contains one or more TCLAS elements to specify the traffic filter as defined in 8.4.2.33.

The TCLAS Elements field contains one or more TCLAS elements to specify the traffic filter as defined in 8.4.2.33. The number of TCLAS elements is limited and the total size of the TFS Request element is less than 255 octets.

The TCLAS Processing Element field is optionally present and defines how multiple TCLAS elements are processed as defined in 8.4.2.35.

The TFS Request element is included in TFS Request frames, as described in 8.5.14.15, and WNM-Sleep Mode Request frames, as described in 8.5.14.18. The use of the TFS Request element and frames is described in 10.23.11.

8.4.2.83 TFS Response element

The TFS Response element defines information about the status of the requested traffic filter. The format of the TFS Response element is defined in Figure 8-336.

			One or more Status subelements
	Element ID	Length	Status subelements
Octets:	1	1	4*n*

Figure 8-336—TFS Response element format

The Element ID field is equal to the TFS Response value in Table 8-54.

The Length field is 4*n*, where *n* indicates the total number of TFS Status subelements contained in the element.

The Status Subelement field contains one or more Status subelements described in Table 8-164.

Table 8-164—Status subelements

Identifier	Subelement name	Length (in octets)
1	TFS Status subelement	4 to 254
2	TFS subelement	5 to 254
221	Vendor Specific subelement	5 to 254
0, 3 to 220, 222 to 255	Reserved	

The TFS Status Subelement field contains the information as defined in Figure 8-337.

	Subelement ID	Length	TFS Response Status	TFS ID
Octets:	1	1	1	1

Figure 8-337—TFS Status Subelement format

The Subelement ID field uniquely identifies this subelement to be the TFS Status subelement. The value of this field is 1.

The value of the Length field is 2.

The TFS Response Status field indicates the status returned by the AP responding to the STA's requested traffic filter, as indicated in Table 8-160.

The TFS ID field indicates the unique ID for the TFS traffic filter set.

The TFS Response element is included in TFS Response frames, as described in 8.5.14.16, and WNM-Sleep Mode Response frames, as described in 8.5.14.19. The use of the TFS Response element and frames is described in 10.23.11.

8.4.2.84 WNM-Sleep Mode element

The WNM-Sleep Mode element is used to enter and exit the WNM-Sleep mode. The format of the WNM-Sleep Mode element is shown in Figure 8-338.

	Element ID	Length	Action Type	WNM-Sleep Mode Response Status	WNM-Sleep Interval
Octets:	1	1	1	1	2

Figure 8-338—WNM-Sleep Mode element format

The Element ID field is equal to the WNM-Sleep Mode value in Table 8-54.

The value of the Length field is 4.

The Action Type field is a number that identifies the type of WNM-Sleep mode request and response. The Action Types are shown in Table 8-165.

Table 8-165—Action Type definitions

Name	Action Type value
Enter WNM-Sleep Mode	0
Exit WNM-Sleep Mode	1
Reserved	2–255

The WNM-Sleep Mode Response Status field indicates the status returned by the AP responding to the non-AP STA's WNM-Sleep mode request as defined in Table 8-166. This field is valid only in the WNM Sleep Mode element in a WNM-Sleep Mode Response frame and is reserved otherwise.

Table 8-166—WNM-Sleep Mode Response Status definition

Value	Description
0	Enter/Exit WNM-Sleep Mode Accept.
1	Exit WNM-Sleep Mode Accept, GTK/IGTK update required.
2	Denied. The AP is unable to perform the requested action.
3	Denied temporarily. The AP is unable to perform the requested action at the current time. The request can be submitted again at a later time.
4	Denied. Due to the pending key expiration.
5	Denied. The requested action was not granted due to other WNM services in use by the requesting STA.
6–255	Reserved

The WNM-Sleep Interval field is reserved if the Action Type field is 1.

The WNM-Sleep Interval field indicates to the AP how often a STA in WNM-Sleep Mode wakes to receive Beacon frames, defined as the number of DTIM intervals. The value set to 0 indicates that the requesting non-AP STA does not wake up at any specific interval.

The WNM-Sleep Mode element is included in WNM-Sleep Mode Request frames, as described in 8.5.14.18, and WNM-Sleep Mode Response frames, as described in 8.5.14.19. The use of the WNM-Sleep Mode element and frames is described in 10.2.1.18.

8.4.2.85 TIM Broadcast Request element

The TIM Broadcast Request element contains information about the periodic TIM broadcast being requested by the non-AP STA. The format of the TIM Broadcast Request element is shown in Figure 8-339.

	Element ID	Length	TIM Broadcast Interval
Octets:	1	1	1

Figure 8-339—TIM Broadcast Request element format

The Element ID field is equal to the TIM Broadcast Request value in Table 8-54.

The value of the Length field is 1.

The TIM Broadcast Interval field is the number of beacon periods between TIM frame transmissions. A value of 0 terminates the use of TIM Broadcast for the requesting station.

The TIM Broadcast Request element is included in TIM Broadcast Request frames, as described in 8.5.14.20; Association Request frames, as described in 8.3.3.5; and Reassociation Request frames, as described in 8.3.3.7. The use of the TIM Broadcast Request element and frames is described in 10.2.1.17.

8.4.2.86 TIM Broadcast Response element

The TIM Broadcast Response element contains information about the periodic TIM broadcast by the AP. The format of the TIM Broadcast Response element is shown in Figure 8-340.

	Element ID	Length	Status	TIM Broadcast Interval (optional)	TIM Broadcast Offset (optional)	High Rate TIM Rate (optional)	Low Rate TIM Rate (optional)
Octets:	1	1	1	0 or 1	0 or 4	0 or 2	0 or 2

Figure 8-340—TIM Broadcast Response element format

The Element ID field is equal to the TIM Broadcast Response value in Table 8-54.

The value of the Length field is 1 or 10, depending on the presence of a TIM Broadcast schedule (TIM Broadcast Interval, TIM Broadcast Offset, High Rate TIM Rate, and Low Rate TIM Rate).

The Status field indicates the status of the AP responding to the STA's requested delivery interval, as indicated in Table 8-167.

When the Status field is 0, 1, or 3, the TIM Broadcast Interval field, TIM Broadcast Offset field, High Rate TIM Rate field, and Low Rate TIM Rate field are included in the TIM Broadcast Response element.

Table 8-167—Status field values

Field value	Description
0	Accept
1	Accept, valid timestamp present in TIM frames
2	Denied
3	Overridden
4–255	Reserved

The TIM Broadcast Interval field contains the number of beacon periods between scheduled TIM frame transmissions.

The TIM Broadcast Offset field contains the offset in microseconds with a tolerance of ± 4 microseconds relative to the TBTT for which a TIM frame is scheduled for transmission. The field contains a signed integer.

The High Rate TIM Rate field provides an indication of the rate that is used to transmit the high data rate TIM frame, in units of 0.5 Mb/s. A value of 0 indicates that the high rate TIM frame is not transmitted.

The Low Rate TIM Rate field provides an indication of the rate that is used to transmit the low data rate TIM frame, in units of 0.5 Mb/s. A value of 0 indicates that the low rate TIM frame is not transmitted.

The TIM Broadcast Response element is included in TIM Broadcast Response frames, as described in 8.5.14.21; Association Response frames, as described in 8.3.3.6; and Reassociation Response frames, as described in 8.3.3.8. The use of the TIM Broadcast Response element and frames is described in 10.2.1.17.

8.4.2.87 Collocated Interference Report element

The Collocated Interference Report element contains some characteristics of the reported collocated interference. The Collocated Interference Report element is defined in Figure 8-341.

	Element ID	Length	Report Period	Interference Level	Interference Level Accuracy/ Interference Index
Octets:	1	1	1	1	1

	Interference Interval	Interference Burst Length	Interference Start Time/Duty Cycle	Interference Center Frequency	Interference Bandwidth
Octets	4	4	4	4	2

Figure 8-341— Collocated Interference Report element format

The Element ID field is equal to the Collocated Interference Report value in Table 8-54.

The value of the Length field is 21.

The Report Period field contains an unsigned integer value in units of 200 TUs. If the Report Period field is a value that is greater than 0, then the reporting is periodic, and the field contains the period of sending the report. If the Report Period field is 0, then the reporting is not periodic, and a report is generated when the STA knows of a change in the collocated interference. See 10.23.8 for further details.

The Interference Level field is a twos complement signed integer indicating the maximum level of the collocated interference power in units of dBm over all receive chains averaged over a 4 μs period during an interference period and across interference bandwidth. When the interference level is unknown, the field is +127 dBm. When the interference level is equal or greater than 126 dBm, the field is +126 dBm. If no collocated interference is present the field is –128 dBm. When the interference level is equal or lower than –127 dBm, the field is –127 dBm. The interference level is referenced to the antenna connector (see “antenna connector” in 3.1) used for reception, like RCPI.

The Interference Level Accuracy/Interference Index field is shown in Figure 8-342.

	B0 B3	B4 B7
	Expected Accuracy	Interference Index
Bits	4	4

Figure 8-342—Interference Level Accuracy/Interference Index field format

The Expected Accuracy field represents an unsigned integer indicating the expected accuracy of the estimate of interference in dB with 95% confidence interval. If the Interference Level field is X (dBm) and the expected accuracy field is Y (dB), the actual interference level is in the range of [X – Y, X +Y] with the probability of 95%. The range of expected accuracy is from 0 dB to 14 dB. If accuracy is unknown or greater than 14 dB, then the Expected Accuracy field is 15.

The Interference Index field indicates the interference index that is unique for each type of interference source. The field set to 0 indicates that no collocated interference is present. See 10.23.8 for further details.

The Interference Interval field indicates the interval between two successive periods of interference in microseconds. When the interval between two successive periods of interference is variable the field is 2^{32}–1. When the interval between two successive periods of interference is equal or greater than 2^{32}– 2 the field is 2^{32}– 2. If no collocated interference is present, the field is 0.

The Interference Burst Length field indicates the duration of each period of interference in microseconds. When the duration of each period of interference is variable the field is 2^{32}– 1. When the duration of each period of interference is equal or greater than 2^{32}– 2 the field is 2^{32}– 2. If no collocated interference is present, the field is 0.

The Interference Start Time/Duty Cycle field contains the least significant 4 octets (i.e., B0–B31) of the TSF timer at the start of the interference burst. When either the Interference Interval or the Interference Burst Length fields are set to 2^{32}– 1, this field indicates the average duty cycle. The average duty cycle value is defined as follows:

$$\text{Average duty cycle} = \text{Round-to-Integer}\ ((2^{32} - 2) \times (T_B / T_I))$$

where
T_B is the average interference burst length
T_I is the average interference interval

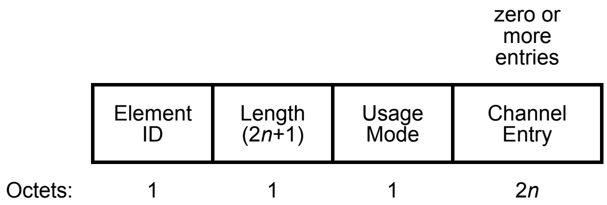
When the interference is nonperiodic or no collocated interference is present, the Interference Start Time field is 0.

The Interference Center Frequency field indicates the center frequency of interference in units of 5 kHz. When center frequency is unknown, the center frequency of the STA’s operating channel is reported. If no collocated interference is present the field is 0.

The Interference Bandwidth field indicates the bandwidth in units of 5 kHz at the –3 dB roll-off point of the interference signal. When bandwidth of the interference signal is unknown, the field is 65 535. When bandwidth of the interference signal is equal or greater than 65 534 the field is 65 534. If no collocated interference is present, the field is 0.

8.4.2.88 Channel Usage element

The Channel Usage element defines the channel usage information for noninfrastructure networks or an off channel TDLS direct link. The format of the Channel Usage element is shown in Figure 8-343.



				zero or more entries
	Element ID	Length (2*n*+1)	Usage Mode	Channel Entry
Octets:	1	1	1	2*n*

Figure 8-343—Channel Usage element format

The Element ID field is equal to the Channel Usage value in Table 8-54.

The Length field is 2*n* + 1, where *n* indicates the total number of Channel Entry subelements contained in the element.

The Usage Mode field is a number that identifies the usage of the recommended channels listed in the Operating Class/Channel Number pair fields. The Usage Mode definitions are shown in Table 8-168.

Table 8-168—Usage Mode definitions

Value	Usage Mode
0	Noninfrastructure IEEE 802.11 network
1	Off-channel TDLS direct link
2–255	Reserved

The Channel Entry field includes zero or more Operating Class and Channel pairs. The format of Channel Entry field is shown in Figure 8-313.

The Channel Usage element may be included in Probe Request frames, as described in 8.3.3.9; Probe Response frames, as described in 8.3.3.10; Channel Usage Request frames, as described in 8.5.14.23; and Channel Usage Response frames, as described in 8.5.14.24. The use of the Channel Usage element and frames is described in 10.23.14.

8.4.2.89 Time Zone element

The Time Zone element contains the local time zone of the AP. The format of the Time Zone element is shown in Figure 8-344.

	Element ID	Length	Time Zone
Octets:	1	1	variable

Figure 8-344—Time Zone element format

The Element ID field is the Time Zone value in Table 8-54.

The value of the Length field is variable.

The format of the Time Zone field is as specified in 8.3 of IEEE Std 1003.1-2004:

stdoffset[dst[offset][,start[/time],end[/time]]]

The length of the field is no less than 4 octets and no more than TZNAME_MAX, as defined in IEEE Std 1003.1-2004. The Time Zone field represents the time zone at the AP's location. The encoding of the field is in ASCII characters as shown in the following Example-1.

Example-1: EST5

Example-2: EST5EDT4,M3.2.0/02:00,M11.1.0/02:00

In the Example-2, the string is interpreted as a time zone that is normally five hours behind UTC, and four hours behind UTC during DST, which runs from the second Sunday in March at 02:00 local time through the first Sunday in November at 02:00 local time. Normally, the time zone is abbreviated "EST" but during DST it is abbreviated "EDT."

8.4.2.90 DMS Request element

The DMS Request element defines information about the group addressed frames to be transmitted as individual addressed frames. The format of the DMS Request element is shown in Figure 8-345.

	Element ID	Length	DMS Descriptor List
Octets:	1	1	variable

Figure 8-345—DMS Request element format

The Element ID field is the DMS Request value in Table 8-54.

The value of the Length field is the length of the DMS Descriptor List field.

The DMS Descriptor List field contains one or more DMS Descriptors. The format of the DMS Descriptor is defined in Figure 8-346.

zero or more
TCLAS
Elements

	DMSID	DMS Length	Request Type	TCLAS Elements	TCLAS Processing Element (optional)	TSPEC Element (optional)	Optional Subelements
Octets:	1	1	1	variable	0 or 3	0 or 57	variable

Figure 8-346—DMS Descriptor

The DMSID field is set to 0 when the Request Type field is “Add” as defined in Table 8-169; otherwise, the DMSID field is set to the nonzero value assigned by the AP STA to identify the DMS traffic flow.

The DMS Length field is set to 1 + *n*, where *n* indicates the total length in octets of all the TCLAS Elements, optional TCLAS Processing Element, optional TSPEC Element, and Optional Subelements fields contained in the DMS Descriptor field.

The Request Type field identifies the type of DMS request. The Request Types are shown in Table 8-169.

Table 8-169—Request Type definitions

Description	Request Type value
Add	0
Remove	1
Change	2
Reserved	3–255

When the Request Type field contains “Add,” the TCLAS Elements field contains one or more TCLAS elements to specify group addressed frames as defined in 8.4.2.33. When the Request Type field contains any value other than “Add,” the TCLAS Elements field contains zero TCLAS elements.

When the Request Type field contains “Add” and when there are two or more TCLAS elements present, the TCLAS Processing Element field contains one TCLAS Processing element to define how these TCLAS elements are to be processed, as defined in 8.4.2.35. Otherwise, the TCLAS Processing Element field contains zero TCLAS Processing elements.

When the Request Type field contains “Add” or “Change,” the TSPEC Element field optionally contains one TSPEC element to specify the characteristics and QoS expectations of the corresponding traffic flow as defined in 8.4.2.32. Otherwise, the TSPEC Element field contains zero TSPEC elements.

The Optional Subelements field format contains zero or more subelements, each consisting of a 1-octet Subelement ID field, a 1-octet Length field and a variable-length Data field, as shown in Figure 8-402. The optional subelements are ordered by nondecreasing Subelement ID.

The Subelement ID field values for the defined optional subelements are shown in Table 8-170. A Yes in the Extensible column of a subelement listed in Table 8-170 indicates that the length of the subelement might be extended in future revisions or amendments of this standard. When the Extensible column of an element is Subelement, then the subelement might be extended in future revisions or amendments of this standard by defining additional subelements within the subelement. See 9.24.9.

Table 8-170—Optional Subelement IDs for DMS Descriptor

Subelement ID	Name	Length field (octets)	Extensible
0–220	Reserved		
221	Vendor Specific	3 to 248	
222–255	Reserved		

The DMS Request element is included in DMS Request frames, as described in 8.5.14.25, and Reassociation Request frames, as described in 8.3.3.7. The use of the DMS Request element and frames is described in 10.23.15.

8.4.2.91 DMS Response element

The DMS Response element provides the status information about the requested group addressed frames. The format of the DMS Response element is shown in Figure 8-347.

			one or more DMS Status field
	Element ID	Length	DMS Status List
Octets:	1	1	variable

Figure 8-347—DMS Response element format

The Element ID field is the DMS Response value in Table 8-54.

The value of the Length field is the total length of the DMS Status List field.

The DMS Status List field contains one or more DMS Status field. The format of the DMS Status field is defined in Figure 8-348.

					zero or more TCLAS Elements			
	DMSID	DMS Length	Response Type	Last Sequence Control	TCLAS Elements	TCLAS Processing Element (optional)	TSPEC Element (optional)	Optional Subelements
Octets:	1	1	1	2	variable	0 or 3	0 or 57	variable

Figure 8-348—DMS Status field format

The DMSID field is assigned by the AP and provides a unique identifier within the BSS for the DMS traffic flow identified by the TCLAS Elements, TCLAS Processing Element, and TSPEC Element fields. The uniqueness of the identifier is independent of the ordering of the TCLAS elements.

The value of the DMS Length field is set to 3 + *n*, where *n* indicates the total length in octets of all the TCLAS Elements, optional TCLAS Processing Element, optional TSPEC Element, and Optional Subelements fields contained in the DMS Status field. When the Response Type field is set to "Terminate," the value of the DMS Length field is set to 3.

The Response Type field indicates the response type returned by the AP responding to the non-AP STA's request, as indicated in Table 8-171.

Table 8-171—Response Type field values

Field value	Description	Notes
0	Accept	AP accepts the DMS request
1	Denied	AP rejects the DMS request
2	Terminate	AP terminates DMS previously accepted DMS request
3–255	Reserved	

When the Last Sequence Control field is not supported the Last Sequence Control field is set to 65 535. When the Last Sequence Control field is supported and the Response Type field does not contain "Terminate," the Last Sequence Control field is reserved.

When the Response Type field is "Terminate" and the Last Sequence Control field is supported, Bit 0 to Bit 3 of the Last Sequence Control field is 0, and Bit 4 to Bit 15 of the Last Sequence Control field contains the sequence number of the last group addressed frame that the AP converted to an individually addressed frame and sent successfully to the non-AP STA that is the recipient of the DMS Response frame. If this last frame received by the non-AP STA prior to DMS termination has not also been sent using a group addressed frame, the Last Sequence Control field is set to 65 534.

When the Response Type field contains "Accept" or "Denied," the TCLAS Elements field contains one or more TCLAS elements to specify group addressed frames as defined in 8.4.2.33. Otherwise, the TCLAS Elements field contains zero TCLAS elements.

When the Response Type field contains "Accept" or "Denied," the TCLAS Processing Element field optionally contains one TCLAS Processing element to define how these TCLAS elements are to be processed, as defined in 8.4.2.35. When the Response Type field contains "Terminate" or when there is only one TCLAS element, the TCLAS Processing Element field contains zero TCLAS Processing elements.

When the Response Type field contains "Accept" or "Denied," the TSPEC Element field optionally contains one TSPEC element to specify the characteristics and QoS expectations of the corresponding traffic flow as defined in 8.4.2.32. Otherwise, the TSPEC Element field contains zero TSPEC elements.

The Optional Subelements field contains zero or more subelements, each consisting of a 1-octet Subelement ID field, a 1-octet Length field and a variable-length Data field, as shown in Figure 8-402. The optional subelements are ordered by nondecreasing Subelement ID.

The Subelement ID field values for the defined optional subelements are shown in Table 8-172. A Yes in the Extensible column of a subelement listed in Table 8-172 indicates that the length of the subelement might be extended in future revisions or amendments of this standard. When the Extensible column of an element is Subelement, then the subelement might be extended in future revisions or amendments of this standard by defining additional subelements within the subelement. See 9.24.9.

Table 8-172—Optional Subelement IDs for DMS Status

Subelement ID	Name	Length field (octets)	Extensible
0–220	Reserved		
221	Vendor Specific	3 to 248	
222–255	Reserved		

The DMS Response element is included in DMS Response frames, as described in 8.5.14.26, and Reassociation Response frames, as described in 8.3.3.8. The use of the DMS Response element and frames is described in 10.23.15.

8.4.2.92 Destination URI element

The Destination URI element contains URI and ESS Detection Interval values from the requesting STA that the responding STA may use to deliver Event or Diagnostic Report frames. The format of the Destination URI element is given in Figure 8-349.

	Element ID	Length	ESS Detection Interval	URI
Octets:	1	1	1	1–253

Figure 8-349—Destination URI element format

The Element ID field is equal to the Destination URI value in Table 8-54.

The value of the Length field is 1 plus the length of the URI field.

The ESS Detection Interval field is defined in 8.4.2.73.2 and its use for Event and Diagnostic requests is described in 10.23.2 and 10.23.3.

The URI field specifies the destination URI for Event and Diagnostic reports using the format defined in IETF RFC 3986. The URI field value is limited to 253 octets.

The Destination URI element is included as the last element in an Event or Diagnostic Request frame.

The Destination URI element is included in Event Request frames, as described in 8.5.14.2, or Diagnostic Request frames, as described in 8.5.14.4.

Use of the Destination URI element in an Event Request frame is described in 10.23.2.1. Use of the Destination URI element in a Diagnostic Request frame is described in 10.23.3.1.

8.4.2.93 U-APSD Coexistence element

The U-APSD Coexistence provides the duration of requested transmission during a U-APSD service period. The format of the U-APSD Coexistence element is shown in Figure 8-350.

	Element ID	Length	TSF 0 Offset	Interval/ Duration	Optional Subelements
Octets:	1	1	8	4	variable

Figure 8-350—U-APSD Coexistence element format

The Element ID field is equal to the U-APSD Coexistence value in Table 8-54.

The value of the Length field is 12 plus the length of any additional subelements present.

A nonzero value of the TSF 0 Offset field is the number of microseconds since TSF time 0 when the non-AP STA knew the start of interference. The AP uses the TSF 0 Offset field together with the Interval/Duration field to enqueue frames for transmission to the non-AP STA using the procedures as described in 10.2.1.5.2.

A TSF 0 Offset field value of 0 indicates the non-AP STA requests the AP transmit frames to the non-AP STA using the procedure described in 10.2.1.5.2 for the case where TSF 0 Offset is equal to 0.

The Interval/Duration field is defined as follows:

- When the TSF 0 Offset is 0, the Interval/Duration field is the number of microseconds during the U-APSD service period when the AP transmits frames to the non-AP STA as described in 10.2.1.5.2.
- When the TSF 0 Offset is nonzero, the Interval/Duration field is the number of microseconds between the start of consecutive interference bursts.

The Interval/Duration field value of 0 is reserved.

The Optional Subelements field format contains zero or more subelements, each consisting of a 1-octet Subelement ID field, a 1-octet Length field and a variable-length Data field, as shown in Figure 8-402. The optional subelements are ordered by nondecreasing Subelement ID.

The Subelement ID field values for the defined optional subelements are shown in Table 8-173. A Yes in the Extensible column of a subelement listed in Table 8-173 indicates that the length of the subelement might be extended in future revisions or amendments of this standard. When the Extensible column of an element is Subelement, then the subelement might be extended in future revisions or amendments of this standard by defining additional subelements within the subelement. See 9.24.9.

Table 8-173—Optional Subelement IDs for U-APSD Coexistence

Subelement ID	Name	Length field (octets)	Extensible
0–220	Reserved		
221	Vendor Specific	1 to 239	
222–255	Reserved		

8.4.2.94 Interworking element

The Interworking element contains information about the interworking service capabilities of a STA as shown in Figure 8-351.

	Element ID	Length	Access Network Options	Venue Info (optional)	HESSID (optional)
Octets:	1	1	1	0 or 2	0 or 6

Figure 8-351—Interworking element format

The Length field is a 1-octet field whose value is 1 plus the sum of the lengths of each optional field present in the element.

The format of Access Network Options field is shown in Figure 8-352.

Bits:	B0 B3	B4	B5	B6	B7
	Access Network Type	Internet	ASRA	ESR	UESA

Figure 8-352—Access Network Options field format

A non-AP STA sets Internet, ASRA, ESR, and UESA fields to 0 when including the Interworking element in the Probe Request frame. A non-AP STA sets the Internet, ASRA and ESR bits to 0 when including the Interworking element in (Re)association Request frames. A mesh STA sets the Internet bit to 0 when including the Interworking element in Mesh Peering Open frames. In (Re)association Request frames, a non-AP STA sets the UESA bit according to the procedures in 10.3.5. The access network types are shown in Table 8-174. The Access Network Type field is set by the AP or the mesh STA to advertise its access network type to non-AP STAs. A non-AP STA uses this field to indicate the desired access network type in an active scan. See V.2 for informative text on usage of fields contained within the Interworking element.

Table 8-174—Access network type

Access network type	Meaning	Description
0	Private network	Nonauthorized users are not permitted on this network. Examples of this access network type are home networks and enterprise networks, which may employ user accounts. Private networks do not necessarily employ encryption.
1	Private network with guest access	Private network but guest accounts are available. Example of this access network type is enterprise network offering access to guest users.
2	Chargeable public network	The network is accessible to anyone, however, access to the network requires payment. Further information on types of charges may be available through other methods (e.g., IEEE 802.21, http/https redirect or DNS redirection). Examples of this access network type is a hotspot in a coffee shop offering internet access on a subscription basis or a hotel offering in-room internet access service for a fee.

Table 8-174—Access network type *(continued)*

Access network type	Meaning	Description
3	Free public network	The network is accessible to anyone and no charges apply for the network use. An example of this access network type is an airport hotspot or municipal network providing free service.
4	Personal device network	A network of personal devices. An example of this type of network is a camera attaching to a printer, thereby forming a network for the purpose of printing pictures.
5	Emergency services only network	A network dedicated and limited to accessing emergency services.
6 to 13	Reserved	Reserved
14	Test or experimental	The network is used for test or experimental purposes only.
15	Wildcard	Wildcard access network type

Bit 4 is the Internet field. The AP or mesh STA sets this field to 1 if the network provides connectivity to the Internet; otherwise it is set to 0 indicating that it is unspecified whether the network provides connectivity to the Internet.

Bit 5 is the Additional Step Required for Access (ASRA) field. It is set to 1 by the AP to indicate that the network requires a further step for access. It is set to 0 whenever dot11RSNAActivated is true. For more information, refer to Network Authentication Type Information in 8.4.4.6. For a mesh STA the ASRA field is used as an emergency indicator. If a mesh STA requires emergency services, the ASRA field is set to 1; otherwise it is set to 0. See 10.24.6.

Bit 6 is the ESR (emergency services reachable) field. It is set to 1 by the AP or mesh STA to indicate that emergency services are reachable through the AP or mesh STA; otherwise it is set to 0 indicating that it is unable to reach the emergency services. See 10.24.6.

Bit 7 is the UESA (unauthenticated emergency service accessible) field. When set to 0, this field indicates that no unauthenticated emergency services are reachable through this AP or mesh STA. When set to 1, this field indicates that higher layer unauthenticated emergency services are reachable through this AP or mesh STA. A STA uses the Interworking element with the UESA bit set to 1 to gain unauthenticated access to a BSS to access emergency services. A mesh STA uses the Interworking element with the UESA bit set to 1 to gain unauthenticated access to another mesh STA to access emergency services. See 10.3.5.

The Venue Info field is defined in 8.4.1.34.

The HESSID field, which is the identifier for a homogeneous ESS, specifies the value of HESSID; see 10.24.2. A STA uses this field to indicate the desired HESSID in an active scan per 10.1.4. The HESSID field for an AP is set to the value of dot11HESSID. This optional field is not used by mesh STAs.

8.4.2.95 Advertisement Protocol element

The Advertisement Protocol element contains information that identifies a particular advertisement protocol and its corresponding Advertisement Control. The Advertisement Protocol element format is shown in Figure 8-353.

	Element ID	Length	Advertisement Protocol Tuple # 1	...	Advertisement Protocol Tuple # n (optional)
Octets:	1	1	variable		variable

Figure 8-353—Advertisement Protocol element format

The Length field is a 1-octet field whose value is equal to the sum of the lengths of the Advertisement Protocol Tuple fields.

The format of Advertisement Protocol Tuple field is shown in Figure 8-354.

	Query Response Info	Advertisement Protocol ID
Octets:	1	variable

Figure 8-354—Advertisement Protocol Tuple field format

The format of Query Response Info field is shown in Figure 8-355.

Bits:	B0 B6	B7
	Query Response Length Limit	PAME-BI

Figure 8-355—Query Response Info field format

The Query Response Info field is defined as follows:

- The Query Response Length Limit indicates the maximum number of octets a STA will transmit in the Query Response field contained within one or more GAS Comeback Response frames. The Query Response Length Limit may be set to a value larger than the maximum MMPDU size in which case the Query Response spans multiple MMPDUs. The Query Response Length Limit is encoded as an integer number of 256 octet units. A value of 0 is not permitted. A value of 0x7F means the maximum limit enforced is determined by the maximum allowable number of fragments in the GAS Query Response Fragment ID (see 8.4.1.33). The requesting STA sets the Query Response Length Limit to 0 on transmission and the responding STA ignores it upon reception.
- Bit 7, the Preassociation Message Exchange BSSID Independent (PAME-BI) bit, is used by an AP to indicate whether the Advertisement Server, which is the non-AP STA's peer for this Advertisement Protocol, will return a Query Response that is independent of the BSSID used for the GAS frame exchange. This bit is set to 1 to indicate the Query Response is independent of the BSSID; it is set to 0 to indicate that the Query Response may be dependent on the BSSID. See 10.24.3.1 for further information. Bit 7 is reserved for non-AP STAs.

The Advertisement Protocol ID is a variable-length field. When this field contains a vendor-specific Advertisement Protocol ID, then this field will be structured per the Vendor Specific element defined in 8.4.2.28, where the Element ID of the Vendor Specific element of 8.4.2.28 is the first octet of the field and contains the vendor-specific value for Advertisement Protocol ID defined in Table 8-175; otherwise its length is 1 octet and its value is one of the values in Table 8-175. When one or more vendor-specific tuples are included in the Advertisement Protocol element, their total length needs to be constrained such that the total length of all the Advertisement Protocol Tuple fields (both vendor specific and otherwise) is less than or equal to 255 octets.

Table 8-175—Advertisement protocol ID definitions

Name	Value
Access Network Query Protocol (ANQP)	0
MIH Information Service	1
MIH Command and Event Services Capability Discovery	2
Emergency Alert System (EAS)	3
Reserved	4–220
Vendor Specific	221
Reserved	222–255

- ANQP supports information retrieval from an Advertisement Server. ANQP is a protocol used by a requesting STA to query another STA (i.e., the receiving STA might respond to queries with or without proxying the query to a server in an external network). See 10.24.3.2 for information on ANQP procedures.
- MIH Information Service is a service defined in IEEE Std 802.21-2008 to support information retrieval from an information repository.
- MIH Command and Event Services capability discovery is a mechanism defined in IEEE 802.21 (see IEEE Std 802.21-2008) to support discovering capabilities of command service and event service entities in a STA or an external network.
- The EAS allows a network to disseminate emergency alert notifications from an external network to non-AP STAs. EAS uses the message format as defined in OASIS EDXL.
- Advertisement Protocol ID 221 is reserved for Vendor Specific Advertisement Protocols. When the Advertisement Protocol ID is equal to 221, the format of the Advertisement Protocol ID subfield follows the format of the Vendor Specific element in 8.4.2.28.

8.4.2.96 Expedited Bandwidth Request element

The Expedited Bandwidth Request element is transmitted from a non-AP STA to an AP in an ADDTS Request frame containing a TSPEC element and provides usage information for the bandwidth request. The Expedited Bandwidth Request element format is shown in Figure 8-356.

	Element ID	Length	Precedence Level
Octets:	1	1	1

Figure 8-356—Expedited Bandwidth Request element format

The Length field is 1.

The Precedence Level field is provided in Table 8-176.

The precedence levels are derived from the 3rd Generation Partnership Project (3GPP) document 3GPP TS 22.067 [B3].

The first responders (public) in Table 8-176 are government agencies or entities acting on behalf of the government, and the first responders (private) are private entities, such as individuals or companies.

Table 8-176—Precedence Level field description

Precedence level value	Description
0–15	Reserved
16	Emergency call, defined in NENA 08-002 [B52]
17	First responder (public)
18	First responder (private)
19	MLPP Level A
20	MLPP Level B
21	MLPP Level 0
22	MLPP Level 1
23	MLPP Level 2
24	MLPP Level 3
25	MLPP Level 4
26–255	Reserved

8.4.2.97 QoS Map Set element

The QoS Map Set element is transmitted from an AP to a non-AP STA in a (Re)association Response frame or a QoS Map Configure frame and provides the mapping of higher layer quality-of-service constructs to User Priorities defined by transmission of Data frames in this standard. This element maps the higher layer priority from the DSCP field used with the Internet Protocol to User Priority as defined by this standard. The QoS Map Set element is shown in Figure 8-357.

	Element ID	Length	DSCP Exception #1 (optional)	...	DSCP Exception #n (optional)	UP 0 DSCP Range	UP 1 DSCP Range	UP 2 DSCP Range	...	UP 7 DSCP Range
Octets:	1	1	2		2	2	2	2		2

Figure 8-357—QoS Map Set element description

The Length field is set to 16+2×n, where n is the number of Exception fields in the QoS Map set.

DSCP Exception fields are optionally included in the QoS Map Set. If included, the QoS Map Set has a maximum of 21 DSCP Exception fields. The format of the exception field is shown in Figure 8-358.

	DSCP Value	User Priority
Octets:	1	1

Figure 8-358—DSCP Exception format

The DSCP value in the DSCP Exception field is in the range 0 to 63 inclusive, or 255; the User Priority value is between 0 and 7, inclusive.

— When a non-AP STA begins transmission of a Data frame containing the Internet Protocol, it matches the DSCP field in the IP header to the corresponding DSCP value contained in this element.

The non-AP STA will first attempt to match the DSCP value to a DSCP exception field and uses the UP from the corresponding UP in the same DSCP exception field if successful; if no match is found then the non-AP STA attempts to match the DSCP field to a UP n DSCP Range field, and uses the n as the UP if successful; and otherwise uses a UP of 0.

- Each DSCP Exception field has a unique DSCP Value.

	DSCP Low Value	DSCP High Value
Octets:	1	1

Figure 8-359—DSCP Range description

The QoS Map Set has a DSCP Range field corresponding to each of the 8 user priorities. The format of the range field is shown in Figure 8-359. The DSCP Range value is between 0 and 63 inclusive, or 255.

- The DSCP range for each user priority is nonoverlapping.
- The DSCP High Value is greater than or equal to the DSCP Low Value.
- If the DSCP Range high value and low value are both equal to 255, then the corresponding UP is not used.

8.4.2.98 Roaming Consortium element

The Roaming Consortium element contains information identifying the roaming consortium and/or SSP whose security credentials can be used to authenticate with the AP transmitting this element; see 10.24.8. The element's format is shown in Figure 8-360.

	Element ID	Length	Number of ANQP OIs	OI #1 and #2 Lengths	OI #1	OI #2 (optional)	OI #3 (optional)
Octets:	1	1	1	1	variable	variable	variable

Figure 8-360—Roaming Consortium element format

The Length field is a 1-octet field whose value is equal to 2 plus the sum of the number of octets in each OI field present.

The format of the Number of ANQP OIs field is a one-octet unsigned integer whose value is the number of additional roaming consortium organization identifiers (OIs) obtainable via ANQP. A value of 0 means that no additional OIs will be returned in response to an ANQP query for the Roaming Consortium list. A value of 255 means that 255 or more additional OIs are obtainable via ANQP.

The OI #1 and #2 Lengths field format is shown in Figure 8-361.

- The value of the OI #1 Length subfield is the length in octets of the OI #1 field.
- The value of the OI #2 Length subfield is the length in octets of the OI #2 field. If the OI #2 field is not present, the value of the OI #2 Length subfield is set to 0.

NOTE—When there are three OIs, the OI #3 Length is calculated by subtracting sum of 2 plus the value of the OI #1 Length subfield plus the value of the OI #2 Length subfield from the value of the Length field.

Bits:	B0 B3	B4 B7
	OI #1 Length	OI #2 Length

Figure 8-361—OI #1 and #2 Lengths field format

The OI field is defined in 8.4.1.31. Each OI identifies a roaming consortium (group of SSPs with inter-SSP roaming agreement) or a single SSP. The value of the OI(s) in this table are equal to the value of the first 3 OIs in the dot11RoamingConsortiumTable. If fewer than 3 values are defined in the dot11RoamingConsortiumTable, then only as many OIs as defined in the table are populated in this element. The values of the OIs in this element are equal to the values of the first OIs, up to 3, from the table.

8.4.2.99 Emergency Alert Identifier element

The Emergency Alert Identifier element provides a hash to identify instances of the active EAS messages that are currently available from the network. The hash allows the non-AP STA to assess whether an EAS message advertised by an AP has been previously received and therefore whether it is necessary to download from the network. The format of the Emergency Alert Identifier element is provided in Figure 8-362.

	Element ID	Length	Alert Identifier Hash
Octets:	1	1	8

Figure 8-362—Emergency Alert Identifier element format

The Length field is a 1-octet field whose value is equal to 8.

The Alert Identifier Hash (AIH) is an 8-octet field. It is a unique value used to indicate an instance of an EAS message. The value of this field is the hash produced by the HMAC-SHA1-64 hash algorithm operating on the EAS message.

AIH =HMAC-SHA1-64(“ES_ALERT”, Emergency_Alert_Message)

Where AIH is then truncated to the first 64 bits of this function.

Emergency_Alert_Message is the EAS message itself.

8.4.2.100 Mesh Configuration element

8.4.2.100.1 General

The Mesh Configuration element shown in Figure 8-363 is used to advertise mesh services. It is contained in Beacon frames and Probe Response frames transmitted by mesh STAs, and is also contained in Mesh Peering Open and Mesh Peering Confirm frames.

	Element ID	Length	Active Path Selection Protocol Identifier	Active Path Selection Metric Identifier	Congestion Control Mode Identifier	Synchronization Method Identifier	Authentication Protocol Identifier	Mesh Formation Info	Mesh Capability
Octets	1	1	1	1	1	1	1	1	1

Figure 8-363—Mesh Configuration element format

The Element ID field is set to the value given in Table 8-54 for this element.

The Length field is set to 7.

The remainder of the fields are described in the following subclauses.

8.4.2.100.2 Active Path Selection Protocol Identifier

The Active Path Selection Protocol Identifier field indicates the path selection protocol that is currently activated in the MBSS. Table 8-177 provides path selection protocol identifier values defined by this standard.

Table 8-177—Active Path Selection Protocol Identifier field values

Value	Meaning
0	Reserved
1	Hybrid wireless mesh protocol (default path selection protocol) defined in 13.10 (default path selection protocol)
2–254	Reserved
255	Vendor specific (The active path selection protocol is specified in a Vendor Specific element)

When the Active Path Selection Protocol Identifier field is 255, the active path selection protocol is specified by a Vendor Specific element that is present in the frame. The content of the Vendor Specific element is beyond the scope of this standard. (See 8.4.2.28.)

8.4.2.100.3 Active Path Selection Metric Identifier

The Active Path Selection Metric Identifier field indicates the path metric that is currently used by the active path selection protocol in the MBSS. Table 8-178 provides the path selection metric identifier values defined by this standard.

Table 8-178—Active Path Selection Metric Identifier field values

Value	Meaning
0	Reserved
1	Airtime link metric defined in 13.9 (default path selection metric)
2–254	Reserved
255	Vendor specific (The active metric is specified in a Vendor Specific element)

When the Active Path Selection Metric Identifier field is 255, the active path metric is specified by a Vendor Specific element that is present in the frame. The content of the Vendor Specific element is beyond the scope of this standard. (See 8.4.2.28.)

8.4.2.100.4 Congestion Control Mode Identifier

The Congestion Control Mode Identifier field indicates the congestion control protocol that is currently activated in the MBSS. Table 8-179 provides the congestion control mode identifier values defined by this standard.

Table 8-179—Congestion Control Mode Identifier field values

Value	Meaning
0	Congestion control is not activated (default congestion control mode)
1	Congestion control signaling protocol defined in 13.12.2
2–254	Reserved
255	Vendor specific (The active congestion control protocol is specified in a Vendor Specific element)

The congestion mode identifier value of 0 indicates the mesh STA has no active congestion control protocol, and is set as the default value for the congestion control mode identifier in the MBSS.

When the Congestion Control Mode Identifier field is 255, the active congestion control protocol is specified by a Vendor Specific element that is present in the frame. The content of the Vendor Specific element is beyond the scope of this standard. (See 8.4.2.28.)

8.4.2.100.5 Synchronization Method Identifier

The Synchronization Method Identifier field indicates the synchronization method that is currently activated in the MBSS. Table 8-180 provides the synchronization method identifier values defined by this standard.

Table 8-180—Synchronization Method Identifier field values

Value	Meaning
0	Reserved
1	Neighbor offset synchronization method defined in 13.13.2.2 (default synchronization method)
2–254	Reserved
255	Vendor specific (The active synchronization method is specified in a Vendor Specific element)

The neighbor offset synchronization method is defined as the default synchronization method among mesh STAs. The details of the neighbor offset synchronization method are described in 13.13.2.2.

When the Synchronization Method Identifier field is 255, the active synchronization method is specified by a Vendor Specific element that is present in the frame. The content of the Vendor Specific element is beyond the scope of this standard. (See 8.4.2.28.)

8.4.2.100.6 Authentication Protocol Identifier

The Authentication Protocol Identifier field indicates the type of authentication protocol that is currently used to secure the MBSS. Table 8-181 provides the authentication protocol identifier values defined by this standard.

Table 8-181—Authentication Protocol Identifier field values

Value	Meaning
0	No authentication method is required to establish mesh peerings within the MBSS
1	SAE defined in 11.3
2	IEEE 802.1X authentication
3–254	Reserved
255	Vendor specific (The active authentication protocol is specified in a Vendor Specific element)

When the Authentication Protocol Identifier field is 255, the active authentication protocol is specified by a Vendor Specific element that is present in the frame. The content of the Vendor Specific element is beyond the scope of this standard. (See 8.4.2.28.)

8.4.2.100.7 Mesh Formation Info

The format of the Mesh Formation Info field is shown in Figure 8-364.

	B0	B1 B6	B7
	Connected to Mesh Gate	Number of Peerings	Connected to AS
Bits:	1	6	1

Figure 8-364—Mesh Formation Info field

The Connected to Mesh Gate subfield is set to 1, if the mesh STA has a mesh path to a mesh gate that announces its presence using GANN elements, RANN elements, or PREQ elements, and set to 0 otherwise.

The Number of Peerings subfield contains an unsigned integer that indicates the number of mesh peerings currently maintained by the mesh STA or 63, whichever is smaller.

The Connected to AS subfield is set to 1 if the Authentication Protocol Identifier field in the Mesh Configuration element is set to 2 (indicating IEEE 802.1X authentication) and the mesh STA has an active connection to an AS.

NOTE—When an AS is collocated with an IEEE 802.1X authenticator an active connection is implicitly true.

8.4.2.100.8 Mesh Capability

The Mesh Capability field comprises a set of values indicating whether a mesh STA is a possible candidate for mesh peering establishment. The details of the Mesh Capability field are shown in Figure 8-365.

	B0	B1	B2	B3	B4	B5	B6	B7
	Accepting Additional Mesh Peerings	MCCA Supported	MCCA Enabled	Forwarding	MBCA Enabled	TBTT Adjusting	Mesh Power Save Level	Reserved
Bits:	1	1	1	1	1	1	1	1

Figure 8-365—Mesh Capability field

The Accepting Additional Mesh Peerings subfield is set to 1 if the mesh STA is willing to establish additional mesh peerings with other mesh STAs and set to 0 otherwise (i.e., the Accepting Additional Mesh Peerings subfield is set in accordance with dot11MeshAcceptingAdditionalPeerings). When the Mesh Configuration element is included in the Mesh Peering Open frame and in the Mesh Peering Confirm frame, the Accepting Additional Mesh Peerings subfield is set to 1.

The MCCA Supported subfield is set to 1 if the mesh STA implements MCCA and set to 0 otherwise (i.e., the MCCA Supported subfield is set in accordance with dot11MCCAImplemented).

The MCCA Enabled subfield is set to 1 if the mesh STA is using the MCCA and set to 0 otherwise (i.e., the MCCA Enabled subfield is set in accordance with dot11MCCAActivated).

The Forwarding subfield is set to 1 if the mesh STA forwards MSDUs and set to 0 otherwise (i.e., the Forwarding subfield is set in accordance with dot11MeshForwarding).

The MBCA Enabled subfield is set to 1 if the mesh STA is using MBCA, and is set to 0 otherwise (i.e., the MBCA Enabled subfield is set in accordance with dot11MBCAActivated). (See 13.13.4.)

The TBTT Adjusting subfield is set to 1 while the TBTT adjustment procedure is ongoing, and is set to 0 otherwise. (See 13.13.4.4.3.)

The Mesh Power Save Level subfield is set to 1 if at least one of the peer-specific mesh power modes is deep sleep mode and set to 0 otherwise. The Mesh Power Save Level subfield is reserved when the Power Management field in the Frame Control Field is set to 0. See 8.2.4.5.11.

8.4.2.101 Mesh ID element

The Mesh ID element is used to advertise the identification of an MBSS and is described in 13.2.2. The format of the Mesh ID element is shown in Figure 8-366. The Mesh ID element is transmitted in Mesh Peering Open frames, Mesh Peering Confirm frames, Mesh Peering Close frames, Beacon frames, and Probe Request and Response frames.

	Element ID	Length	Mesh ID
Octets:	1	1	0–32

Figure 8-366—Mesh ID element format

The Element ID field is set to the value given in Table 8-54 for this element.

The Length field is between 0 and 32 octets.

A Mesh ID field of length 0 indicates the wildcard Mesh ID, which is used within Probe Request frame.

Detailed usage of the Mesh ID element is described in 13.2.2.

8.4.2.102 Mesh Link Metric Report element

The Mesh Link Metric Report element is transmitted by a mesh STA to a neighbor peer mesh STA to indicate the quality of the link between the transmitting mesh STA and the neighbor peer mesh STA. The format of the Mesh Link Metric Report element is shown in Figure 8-367.

	Element ID	Length	Flags	Link Metric
Octets:	1	1	1	variable

Figure 8-367—Mesh Link Metric Report element format

The Element ID field is set to the value given in Table 8-54 for this element.

The Length field indicates the number of octets in the Information field (fields following the Element ID and Length fields).

The format of the Flags field is shown in Figure 8-368.

	B0	B1 B7
	Request	Reserved
Bits:	1	7

Figure 8-368—Flags field

The Flags field is set as follows:

- Bit 0: Request subfield (0 = not a request, 1 = link metric report request). A Request subfield equal to 1 indicates that the recipient of Mesh Link Metric Report element is requested to send a link metric report to the transmitter of the Mesh Link Metric Report element.
- Bit 1–7: Reserved.

The Link Metric field indicates the value of the link metric associated with the mesh link between the peer mesh STA transmitting the Mesh Link Metric Report and the neighbor mesh STA receiving the Mesh Link Metric Report. The length and the data type of the Link Metric field are determined by the active path selection metric identifier (see 8.4.2.100.3). The length and the data type for the airtime link metric are given in Table 13-5 in 13.9.

8.4.2.103 Congestion Notification element

The Congestion Notification element is used to indicate the congestion status of the mesh STA per mesh destination and AC, and the duration for which the STA expects the congestion to last. The format of the Congestion Notification element is shown in Figure 8-369. The Congestion Notification element is included in Congestion Control Notification frames, as described in 8.5.17.5.

	Element ID	Length	Destination Mesh STA Address	Congestion Notification Duration Timer (AC_BK)	Congestion Notification Duration Timer (AC_BE)	Congestion Notification Duration Timer (AC_VI)	Congestion Notification Duration Timer (AC_VO)
Octets:	1	1	6	2	2	2	2

Figure 8-369—Congestion Notification element format

The Element ID field is set to the value given in Table 8-54 for this element.

The Length field is set to 14.

The Destination Mesh STA Address field is represented as a 48-bit MAC address and is set to the address of the mesh destination for which the intra-mesh congestion control is applied. It is set to the broadcast address if the intra-mesh congestion control is applied to all destinations.

The element contains four Congestion Notification Duration fields for the four EDCA access categories to indicate the estimated congestion duration per AC at the mesh STA transmitting the congestion notification. The congestion notification duration values are encoded as unsigned integers in units of 100 µs.

8.4.2.104 Mesh Peering Management element

The Mesh Peering Management element is used to manage a mesh peering with a neighbor mesh STA. The format of the Mesh Peering Management element is shown in Figure 8-370.

	Element ID	Length	Mesh Peering Protocol Identifier	Local Link ID	Peer Link ID (conditional)	Reason Code (conditional)	Chosen PMK (optional)
Octets:	1	1	2	2	2	2	16

Figure 8-370—Mesh Peering Management element format

The Element ID field is set to the value given in Table 8-54 for this element.

The Length field is set to the number of octets in the Mesh Peering Management element following the Length field itself.

The Mesh Peering Protocol Identifier field indicates the type of mesh peering protocol that is currently used to establish mesh peerings. Table 8-182 provides the mesh peering protocol identifier values defined by this standard.

Table 8-182—Mesh Peering Protocol Identifier field values

Value	Meaning
0	Mesh peering management protocol
1	Authenticated mesh peering exchange protocol
2–254	Reserved
255	Vendor specific (The active mesh peering protocol is specified in a Vendor Specific element)

When the Mesh Peering Protocol Identifier field is 255, the active mesh peering protocol is specified by a Vendor Specific element that is present in the frame. The content of the Vendor Specific element is beyond the scope of this standard. (See 8.4.2.28.)

The Local Link ID field is the unsigned integer value generated by the local mesh STA to identify the mesh peering instance.

The conditional components of the Mesh Peering Management element are present depending on the Action field value of the frame in which the Mesh Peering Management element is conveyed.

The Peer Link ID field is the unsigned integer value generated by the peer mesh STA to identify the mesh peering instance. This field is not present for the Mesh Peering Open frame, is present for the Mesh Peering Confirm frame, and is optionally present for the Mesh Peering Close frame. The presence or absence of the Peer Link ID in a Mesh Peering Close is inferred by the Length field.

The Reason Code field enumerates reasons for sending a Mesh Peering Close. It is present for the Mesh Peering Close frame and is not present for Mesh Peering Open or Mesh Peering Confirm frames. The reason code is defined in 8.4.1.7.

The Chosen PMK field is present when dot11MeshSecurityActivated is true and a PMK is shared between the transmitter and receiver of the frame containing the element. It contains the PMKID that identifies the PMK used to protect the Mesh Peering Management frame.

Detailed usage of the Mesh Peering Management element is described in 13.3.6, 13.3.7, 13.3.8, and 13.5.5.

8.4.2.105 Mesh Channel Switch Parameters element

The Mesh Channel Switch Parameters element is used together with Channel Switch Announcement element and Extended Channel Switch Announcement element by a mesh STA in an MBSS to advertise to other mesh STAs when it is changing to a new operating channel and/or operating class. The format of the Mesh Channel Switch Parameters element is shown in Figure 8-371.

	Element ID	Length	Time To Live	Flags	Reason Code	Precedence Value
Octets:	1	1	1	1	2	2

Figure 8-371—Mesh Channel Switch Parameters element format

The Element ID field is set to the value given in Table 8-54 for this element.

The Length field is set to 6.

The Time To Live field is coded as an unsigned integer and indicates the remaining number of hops allowed for this element.

The Flags field indicates the attribute of this channel switch attempt. The format of the Flags field is shown in Figure 8-372.

	B0	B1	B2	B3 B7
	Transmit Restrict	Initiator	Reason	Reserved
Bits:	1	1	1	5

Figure 8-372—Flags field

The Transmit Restrict subfield is set to 1 when the mesh STA asks neighboring peer mesh STAs not to transmit further frames except frames containing Mesh Channel Switch Parameters element on the current channel until the scheduled channel switch. The Transmit Restrict subfield is set to 0 otherwise.

The Initiator subfield is set to 1 when the mesh STA initiates this channel switch attempt. The Initiator subfield is set to 0 when this channel switch attempt is initiated by another mesh STA and propagated by the current mesh STA.

The Reason subfield indicates the validity of the Reason Code field. It is set to 1 if the Reason Code field is valid, and is set to 0 otherwise. When the Reason subfield is 0, the content of the Reason Code field is reserved.

The Reason Code field specifies the reason for the mesh channel switch. The Reason Code is defined in 8.4.1.7. The content of the Reason Code field is valid only when Reason subfield of Flags field is set to 1, and is reserved otherwise.

The Precedence Value field is coded as unsigned integer and is set to a random value in the range 0 to 65535 determined by the initiator of this channel switch attempt.

The Mesh Channel Switch Parameters element is included in Channel Switch Announcement frames, as described in 8.5.2.6, and Extended Channel Switch Announcement frames, as described in 8.5.8.7. During MBSS Channel Switch, the Mesh Channel Switch Parameters element is included in Beacon frames, as described in 8.3.3.2, and Probe Response frames, as described in 8.3.3.10, until scheduled channel switch.

8.4.2.106 Mesh Awake Window element

The Mesh Awake Window element is present in DTIM Beacon frames and is optionally present in Beacon and Probe Response frames. The format of the Mesh Awake Window element is shown in Figure 8-373.

	Element ID	Length	Mesh Awake Window
Octets:	1	1	2

Figure 8-373—Mesh Awake Window element format

The Element ID field is set to the value given in Table 8-54 for this element.

The Length field is set to 2.

The Mesh Awake Window field is 2 octets long and contains an unsigned integer that indicates the duration of the mesh awake window in TUs.

8.4.2.107 Beacon Timing element

The Beacon Timing element is used to advertise the beacon timing information of neighbor STAs (mesh STAs, APs, or STAs in an IBSS). The format of the Beacon Timing element is shown in Figure 8-374.

	Element ID	Length	Report Control	Beacon Timing Information #1	...	Beacon Timing Information #N
Octets:	1	1	1	6	...	6

Figure 8-374—Beacon Timing element format

The Element ID field is set to the value given in Table 8-54 for this element.

The Length field indicates the number of octets in the Information field (fields following the Element ID and Length fields).

The Report Control field is used to signal information about the beacon timing information tuple contained in the Beacon Timing element. The structure of the Report Control field is defined in Figure 8-375.

	Status Number	Beacon Timing Element Number	More Beacon Timing Elements
Bits:	4	3	1

Figure 8-375—Report Control field

The Status Number subfield is set to the status number of the beacon timing information set. The status number is managed as described in 13.13.4.2.4.

The Beacon Timing Element Number subfield is an unsigned integer that indicates the index of the beacon timing information tuple contained in this Beacon Timing element. The Beacon Timing Element Number is set to 0 in the Beacon Timing element for the first or only tuple of the beacon timing information and is incremented by one for each successive tuple of the beacon timing information. The beacon timing information tuples are managed as described in 13.13.4.2.5.

The More Beacon Timing Element subfield is set to 1 if a successive tuple of beacon timing information exists, and set to 0 otherwise.

The Beacon Timing Information field contains the beacon timing information of a neighbor STA. When the mesh STA reports multiple beacon timing information, multiple Beacon Timing Information fields are included in the Beacon Timing element. The structure of the Beacon Timing Information field is defined in Figure 8-376.

	Neighbor STA ID	Neighbor TBTT	Neighbor Beacon Interval
Octets:	1	3	2

Figure 8-376—Beacon Timing Information field

The Neighbor STA ID subfield is an unsigned integer that indicates the identification of the neighbor STA corresponding to this beacon timing information. When a mesh peering is established with this neighbor STA, the MSB of this field is set to 0, and the rest of this field is set to the last 7 digits (7 LSBs) of the AID value assigned to this neighbor mesh STA. When a mesh peering is not established with this neighbor STA, the MSB of this field is set to 1, and the rest of this field is set to the last 7 digits (7 LSBs, taking the I/G bit as the MSB) of the 48-bit MAC address of this neighbor STA.

NOTE—Since the Neighbor STA ID subfield is provided in abbreviated form, it is possible that the same Neighbor STA ID value appears in multiple Beacon Timing Information fields.

The Neighbor TBTT subfield is an unsigned integer that indicates a TBTT of the corresponding neighbor STA, measured in the local TSF timer of the mesh STA. The value is indicated in multiples of 32 μs. When the active synchronization method is the neighbor offset synchronization method, the TBTT is calculated as described in 13.13.4.2.2. The B5 to the B28 (taking the B0 as the LSB) of the calculated TBTT are contained in this subfield.

The Neighbor Beacon Interval subfield is an unsigned integer that indicates the beacon interval being used by the corresponding neighbor STA. The unit of the Neighbor Beacon Interval subfield is TU.

Detailed usage of the Beacon Timing element is described in 13.13.4.2.

8.4.2.108 MCCAOP Setup Request element

8.4.2.108.1 General

The MCCAOP Setup Request element is used to make an MCCAOP reservation. This element is transmitted in individually addressed MCCA Setup Request frames or in group addressed MCCA Setup Request frames. The mesh STA transmitting the MCCA Setup Request element is the MCCAOP owner of the MCCAOPs that will be scheduled with this reservation setup request. The receivers of the MCCAOP Setup Request are the MCCAOP responders. The format of the element is shown in Figure 8-377.

	Element ID	Length	MCCAOP Reservation ID	MCCAOP Reservation
Octets	1	1	1	5

Figure 8-377—MCCAOP Setup Request element format

The Element ID field is set to the value given in Table 8-54 for this element.

The Length field is set to 6.

The MCCAOP Reservation ID field is an eight bit unsigned integer that represents the ID for the MCCAOP reservation. It is determined by the MCCAOP owner. When used in combination with the MAC address of the MCCAOP owner, the MCCAOP Reservation ID uniquely identifies the MCCAOP reservation. If this MCCAOP Setup Request is for an individually addressed transmission, the MCCAOP Reservation ID is between 0 and 127 and the MCCAOP Setup Request element is transmitted in an individually addressed frame to the intended responder. If this MCCAOP Setup Request is for a group addressed transmission, the MCCAOP Reservation ID is between 128 and 254 and the MCCAOP Setup Request element is transmitted in a group addressed frame. The value 255 is not used to identify a single MCCAOP reservation.

The MCCAOP Reservation field is described in 8.4.2.108.2.

8.4.2.108.2 MCCAOP Reservation field

The MCCAOP Reservation field is a 5 octet field specifying a schedule for frame transmissions called MCCAOPs. The MCCAOP Reservation field consists of three subfields and its format is shown in

Figure 8-378.

	MCCAOP Duration	MCCAOP Periodicity	MCCAOP Offset
Octets:	1	1	3

Figure 8-378—MCCAOP Reservation field

The MCCAOP Duration subfield is 1 octet in length and contains an unsigned integer. It specifies the duration of the MCCAOPs in multiples of 32 µs.

The MCCAOP Periodicity subfield is 1 octet in length and contains a positive integer. It specifies the number of MCCAOPs scheduled in each DTIM interval.

The MCCAOP Offset subfield is three octets in length and contains an unsigned integer. It specifies the beginning of the first MCCAOP in each DTIM interval. The value is specified in multiples of 32 µs. The sum of MCCAOP Offset plus MCCAOP Duration is constrained to be smaller than the duration of the DTIM interval divided by MCCAOP Periodicity.

8.4.2.109 MCCAOP Setup Reply element

The MCCAOP Setup Reply element is used to reply to an MCCAOP Setup Request. This element is transmitted in individually addressed MCCA Setup Reply frames. The mesh STA transmitting the MCCA Setup Reply element is the MCCAOP responder of the MCCAOPs scheduled in this reservation setup. The receiver of the MCCAOP Setup Reply is the MCCAOP owner. The format of the element is shown in Figure 8-379.

	Element ID	Length	MCCAOP Reservation ID	MCCA Reply Code	MCCAOP Reservation
Octets:	1	1	1	1	0 or 5

Figure 8-379—MCCAOP Setup Reply element format

The Element ID field is set to the value given in Table 8-54 for this element.

The Length field is set to 2 or 7 octets.

The MCCAOP Reservation ID field is an eight bit unsigned integer that represents the ID for the requested series of MCCAOPs. It is determined by the MCCAOP owner and copied from the MCCAOP Setup Request element. When used in combination with the MAC address of the MCCAOP owner, the MCCAOP Reservation ID uniquely identifies the MCCAOP reservation. If this MCCAOP Setup Request is for an individually addressed transmission, the MCCAOP Reservation ID is between 0 and 127. If this MCCAOP Setup Request is for a group addressed transmission, the MCCAOP Reservation ID is between 128 to 254. The value 255 is not used to identify a single MCCAOP reservation.

The MCCA Reply Code field is a 1 octet field that contains the reply code used in an MCCAOP Setup Reply element. The reply codes are defined in Table 8-183.

Table 8-183—MCCA Reply Code field values

MCCA reply code	Meaning
0	Accept
1	Reject: MCCAOP reservation conflict
2	Reject: MAF limit exceeded
3	Reject: MCCA track limit (dot11MCCAMaxTrackStates) exceeded
4–255	Reserved

The MCCAOP Reservation field includes an alternative to the MCCAOP reservation specified in the MCCAOP Setup Request message. Its format is described in 8.4.2.108.2. When the MCCA Reply Code is 1, the MCCAOP Reservation field might be present. When the MCCA Reply Code is set to other values, the MCCAOP Reservation field is not present.

8.4.2.110 MCCAOP Advertisement Overview element

The MCCAOP Advertisement Overview element is used by a mesh STA to advertise its MCCA Information and information about its MCCAOP Advertisement elements, representing its MCCAOP advertisement set, to its neighbors. This element is transmitted in MCCA Advertisement frames and optionally present in Beacon frames. The format of the MCCAOP Advertisement Overview element is shown in Figure 8-380.

	Element ID	Length	Advertisement Set Sequence Number	Flags	MCCA Access Fraction	MAF Limit	Advertisement Elements Bitmap
Octets	1	1	1	1	1	1	2

Figure 8-380—MCCAOP Advertisement Overview element format

The Element ID field is set to the value given in Table 8-54 for this element.

The Length field is set to 6.

The Advertisement Set Sequence Number field is 1 octet in length and is coded as an unsigned integer. It is set to the advertisement set sequence number of the current MCCAOP advertisement set. The Advertisement Set Sequence Number, together with the MAC address of the transmitter of the MCCAOP Advertisement Overview element, identifies an MCCAOP advertisement set and provides an identifier and a chronological order of different MCCAOP advertisement sets of the same mesh STA.

The format of the Flags field is shown in Figure 8-381.

	B0	B1 B7
	Accept Reservations	Reserved
Bits:	1	7

Figure 8-381—Flags field format

The Flags field is set as follows:

- — Bit 0: Accept Reservations subfield. The Accept Reservations subfield is set to 1 if the mesh STA accepts additional reservations. It is set to 0 otherwise.
- — Bit 1–7: Reserved

The MCCA Access Fraction field is an eight bit unsigned integer. The MCCA Access Fraction field is set to the current value of the MCCA access fraction at the mesh STA rounded down (Floor) to the nearest multiple of (1/255) of the DTIM interval length.

The MAF Limit field is an eight bit unsigned integer. The MAF Limit field is set to the maximum MCCA access fraction allowed at the mesh STA rounded down (Floor) to the nearest multiple of (1/255) of the DTIM interval length.

The Advertisement Elements Bitmap field is 2 octets in length and indicates the MCCAOP Advertisement elements that are part of this MCCAOP advertisement set. The Advertisement Elements Bitmap field is a bitmap. Bit *i* in this bitmap equals 1 if the MCCAOP Advertisement element with MCCAOP Advertisement Element Index equal to *i* is part of this MCCAOP advertisement set, and it equals 0 otherwise.

8.4.2.111 MCCAOP Advertisement element

8.4.2.111.1 General

The MCCAOP Advertisement element is used by a mesh STA to advertise MCCAOP reservations to its neighbors. This element is transmitted in MCCA Advertisement frames and optionally present in Beacon frames. The format of the element is shown in Figure 8-382.

	Element ID	Length	Advertisement Set Sequence Number	MCCAOP Advertisement Element Information	TX-RX Periods Report	Broadcast Periods Report	Interference Periods Report
Octets:	1	1	1	1	variable	variable	variable

Figure 8-382—MCCAOP Advertisement element format

The Element ID field is set to the value given in Table 8-54 for this element.

The Length field is set to 2 to 255 octets.

The Advertisement Set Sequence Number field is 1 octet in length and is coded as an unsigned integer. It is set to the advertisement set sequence number of the current MCCAOP advertisement set. The Advertisement Set Sequence Number, together with the MAC address of the transmitter of the MCCAOP Advertisement element, identifies an MCCAOP advertisement set and provides an identifier and a chronological order of different MCCAOP advertisement sets of the same mesh STA.

The MCCAOP Advertisement Element Information field is 1 octet in length. It is described in 8.4.2.111.2.

The TX-RX Periods Report field is a variable-length field that contains an MCCAOP Reservation Report field, as described in 8.4.2.111.3. This field is only present when the TX-RX Report Present subfield of the MCCAOP Advertisement Element Information field is equal to 1. The TX-RX Periods Report field is described in 9.20.3.7.2.

The Broadcast Periods Report field is a variable-length field that contains an MCCAOP Reservation Report field, as described in 8.4.2.111.3. This field is only present when the Broadcast Report Present subfield of the MCCAOP Advertisement Element Information field is equal to 1. The Broadcast Periods Report field is described in 9.20.3.7.2.

The Interference Periods Report field is a variable-length field that contains an MCCAOP Reservation Report field, as described in 8.4.2.111.3. This field is only present when the Interference Report Present subfield of the MCCAOP Advertisement Element Information field is equal to 1. The Interference Periods Report field is described in 9.20.3.7.2.

8.4.2.111.2 MCCAOP Advertisement Element Information field

The MCCA Information field is 1 octets in length and provides information on the MCCAOP reservations. The field consists of four subfields and its format is shown in Figure 8-383.

	B0 B3	B4	B5	B6	B7
	MCCAOP Advertisement Element Index	TX-RX Report Present	Broadcast Report Present	Interference Report Present	Reserved
Bits:	4	1	1	1	1

Figure 8-383—MCCAOP Advertisement Element Information field

The MCCAOP Advertisement Element Index subfield is a 4-bit unsigned integer. It identifies the MCCAOP Advertisement element.

The TX-RX Report Present subfield is 1 bit in length. It is set to 1 if the TX-RX Periods Report field is present in the MCCAOP Advertisement element and set to 0 if no TX-RX Periods Report field is present.

The Broadcast Report Present subfield is 1 bit in length. It is set to 1 if the Broadcast Periods Report field is present in the MCCAOP Advertisement element and set to 0 if no Broadcast Periods Report field is present.

The Interference Report Present subfield is 1 bit in length. It is set to 1 if the Interference Periods Report field is present in the MCCAOP Advertisement element and set to 0 if no Interference Periods Report field is present.

8.4.2.111.3 MCCAOP Reservation Report field

The MCCAOP Reservation Report field is of variable length and is used to report a number of MCCAOP reservations. The field consists of a variable number of subfields and its format is shown in Figure 8-384.

	Number of Reported MCCAOP Reservations	MCCAOP Reservation 1	...	MCCAOP Reservation n
Octets:	1	5		5

Figure 8-384—MCCAOP Reservation Report field

The Number of Reported MCCAOP reservations is a field of 1 octet with an unsigned integer that specifies the number, n, of MCCAOP Reservations reported in this MCCAOP Reservation Report field.

The MCCAOP Reservation 1 through MCCAOP Reservation n fields specify the MCCAOP reservations reported. Each MCCAOP Reservation field is 5 octets in length and its format is shown in Figure 8-378 in 8.4.2.108.2.

8.4.2.112 MCCAOP Teardown element

The MCCAOP Teardown element is used to announce the teardown of an MCCAOP reservation. The MCCAOP Teardown element is transmitted in individually addressed MCCA Teardown frames or in group addressed MCCA Teardown frames. Its format is shown in Figure 8-385.

	Element ID	Length	MCCAOP Reservation ID	MCCAOP Owner
Octets:	1	1	1	0 or 6

Figure 8-385—MCCAOP Teardown element format

The Element ID field is set to the value given in Table 8-54 for this element.

The Length field is variable and set to 1 or 7 octets.

An MCCAOP Teardown element is transmitted by either the MCCAOP owner or the MCCAOP responder of a MCCAOP reservation to tear down the MCCAOP reservation.

The MCCAOP Reservation ID field is an eight bit unsigned integer that represents the ID for the MCCAOP reservation.

The MCCAOP Owner field is an optional field. It is 6 octets long and indicates the 48-bit MAC address of the MCCAOP owner. This field is only included if the element is transmitted by the MCCAOP responder.

8.4.2.113 GANN element

The GANN (gate announcement) element is used for announcing the presence of a mesh gate in the MBSS. The GANN element is transmitted in a Gate Announcement frame (see 8.5.17.4). The format of the GANN element is shown in Figure 8-386.

	Element ID	Length	Flags	Hop Count	Element TTL	Mesh Gate Address	GANN Sequence Number	Interval
Octets:	1	1	1	1	1	6	4	2

Figure 8-386—GANN element format

The Element ID field is set to the value given in Table 8-54 for this element.

The Length field is set to 15.

The Flags field is reserved.

The Hop Count field is coded as an unsigned integer and indicates the number of hops from the originating mesh gate to the mesh STA transmitting this element.

The Element TTL field is coded as an unsigned integer and indicates the remaining number of hops allowed for this element.

The Mesh Gate Address field is represented as a 48-bit MAC address and is set to the MAC address of the mesh gate.

The GANN Sequence Number field is coded as an unsigned integer and is set to a GANN Sequence Number specific for the originating mesh gate.

The Interval field is coded as an unsigned integer and is set to the number of seconds between the periodic transmissions of Gate Announcements by the mesh gate.

Detailed usage of the GANN element is described in 13.11.2.

8.4.2.114 RANN element

The RANN (root announcement) element is used for announcing the presence of a mesh STA configured as root mesh STA with dot11MeshHWMProotMode set to rann (4). RANN elements are sent out periodically by such a root mesh STA. The RANN element is transmitted in an HWMP Mesh Path Selection frame (see 8.5.17.3). The format of the RANN element is shown in Figure 8-387.

	Element ID	Length	Flags	Hop Count	Element TTL	Root Mesh STA Address	HWMP Sequence Number	Interval	Metric
Octets:	1	1	1	1	1	6	4	4	4

Figure 8-387—RANN element format

The Element ID field is set to the value given in Table 8-54 for this element.

The Length field is set to 21.

The format of the Flags field is shown in Figure 8-388.

	B0	B1 B7
	Gate Announcement	Reserved
Bits:	1	7

Figure 8-388—Flags field format

The Flags field is set as follows:

- Bit 0: Gate Announcement subfield (0 = gate announcement protocol not activated, 1 = gate announcement protocol activated). A Gate Announcement subfield equal to 1 indicates that the Root Mesh STA Address is a mesh gate with dot11MeshGateAnnouncements equal to true.
- Bit 1–7: Reserved.

The Hop Count field is coded as an unsigned integer and indicates the number of hops from the originating root mesh STA to the mesh STA transmitting this element.

The Element TTL field is coded as an unsigned integer and indicates the remaining number of hops allowed for this element.

The Root Mesh STA Address field is represented as a 48-bit MAC address and is set to the MAC address of the root mesh STA.

The HWMP Sequence Number field is coded as an unsigned integer and is set to the HWMP sequence number (SN) specific to the root mesh STA.

The Interval field is coded as an unsigned integer and is set to the number of TUs between the periodic transmissions of Root Announcements.

The Metric field is set to the cumulative metric from the originating root mesh STA to the mesh STA transmitting the announcement.

Detailed usage of the RANN element is described in 13.10.12.

8.4.2.115 PREQ element

The PREQ (path request) element is used for discovering a path to one or more target mesh STAs, maintaining a path (optional), building a proactive (reverse) path selection tree to the root mesh STA, and confirming a path to a target mesh STA (optional). The PREQ element is transmitted in an HWMP Mesh Path Selection frame (see 8.5.17.3). The format of the PREQ element is shown in Figure 8-389.

	Element ID	Length	Flags	Hop Count	Element TTL	Path Discovery ID	Originator Mesh STA Address	Originator HWMP Sequence Number	Originator External Address	Lifetime
Octets:	1	1	1	1	1	4	6	4	0 or 6	4

Metric	Target Count	Per Target Flags #1	Target Address #1	Target HWMP Sequence Number #1	...	Per Target Flags #N	Target Address #N	Target HWMP Sequence Number #N
4	1	1	6	4	...	1	6	4

Figure 8-389—PREQ element format

The Element ID field is set to the value given in Table 8-54 for this element.

The Length field is set to 37 to 252 octets.

The format of the Flags field is shown in Figure 8-390.

	B0	B1	B2	B3 B5	B6	B7
	Gate Announcement	Addressing Mode	Proactive PREP	Reserved	AE	Reserved
Bits	1	1	1	3	1	1

Figure 8-390—Flags field format

The Flags field is set as follows:

- Bit 0: Gate Announcement subfield (0 = gate announcement protocol not activated, 1 = gate announcement protocol activated). A Gate Announcement subfield equal to 1 indicates that the Originator Mesh STA Address is a mesh gate with dot11MeshGateAnnouncements equal to true.
- Bit 1: Addressing Mode subfield (0 = group addressed, 1 = individually addressed). When the Addressing Mode subfield is 0, the PREQ element is sent in an HWMP Mesh Path Selection frame that is group addressed to all neighbor peer mesh STAs. When the Addressing Mode subfield is 1, the PREQ element is sent in an HWMP Mesh Path Selection frame that is individually addressed to a neighbor peer mesh STA. Detailed addressing information is provided in 13.10.7.
- Bit 2: Proactive PREP subfield (0 = off, 1 = on). The Proactive PREP subfield is only of relevance if the Target Address is the broadcast address (all ones). If equal to 1, every recipient of a PREQ with Target Address equal to the broadcast address replies with a PREP. If equal to 0, it will only reply under certain conditions (see 13.10.4.2).
- Bit 3–5: Reserved.
- Bit 6: AE (Address Extension) subfield (1= external address present, 0 = otherwise). An AE subfield equal to 1 indicates that the field Originator External Address is present, and that the originator mesh STA is a proxy for this external address.
- Bit 7: Reserved.

The Hop Count field is coded as an unsigned integer and is set to the number of hops from the originator to the mesh STA transmitting this element.

The Element TTL field is coded as an unsigned integer and indicates the remaining number of hops allowed for this element.

The Path Discovery ID field is coded as an unsigned integer and is set to some unique ID for this PathDiscovery.

The Originator Mesh STA Address field is represented as a 48-bit MAC address and is set to the originator MAC address.

The Originator HWMP Sequence Number field is coded as an unsigned integer and is set to the HWMP SN specific to the originator.

The Originator External Address field is the MAC address of an external STA proxied by the Originator. This field is only present if the AE subfield in the Flags field is set to 1 and is represented as a 48-bit MAC address.

The Lifetime field is coded as an unsigned integer and is set to the time for which mesh STAs receiving the PREQ consider the forwarding information to be valid. The lifetime is measured in TUs.

The Metric field is set to the cumulative metric from the originator to the mesh STA transmitting the PREQ.

The Target Count N field is coded as an unsigned integer and gives the number of targets (N) contained in this PREQ. The maximum value of N is 20. The Per Target Flags field, the Target Address field, and the Target HWMP Sequence Number field are repeated N times in the element.

The format of the Per Target Flags field is shown in Figure 8-391.

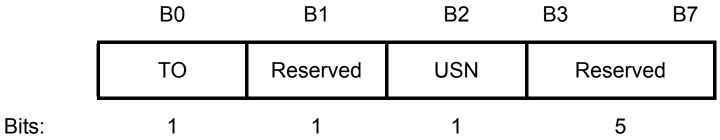


Figure 8-391—Per Target Flags field format

The Per Target Flags field is set as follows:

- Bit 0: TO (Target Only) subfield: The TO subfield defines which mesh STA responds with a PREP element to the PREQ element containing an individual target address. If TO = 1, only the target mesh STA responds with an individually addressed PREP. If TO = 0, intermediate mesh STAs with active forwarding information to the target mesh STA also respond.
- Bit 1: Reserved.
- Bit 2: USN (Unknown Target HWMP Sequence Number) subfield: The USN subfield indicates whether the Target HWMP Sequence Number field of the corresponding target is interpreted as HWMP SN (USN = 0) or not (USN = 1), the latter meaning that a target HWMP SN is unknown at the originator mesh STA.
- Bit 3–7: Reserved.

The Target Address field is represented as a 48-bit MAC address.

The Target HWMP Sequence Number field is coded as an unsigned integer and is the latest known HWMP SN received in the past by the originator mesh STA for any path towards the target. If such a target HWMP SN is not known, the USN subfield is set to 1 and Target HWMP Sequence Number field is reserved.

Detailed usage of the PREQ element is described in 13.10.9.

8.4.2.116 PREP element

The PREP (path reply) element is used to establish a forward path to a target and to confirm that a target is reachable. The PREP is issued in response to a PREQ. The PREP element is transmitted in an HWMP Mesh Path Selection frame (see 8.5.17.3). The format of the PREP element is shown in Figure 8-392.

	Element ID	Length	Flags	Hop Count	Element TTL	Target Mesh STA Address	Target HWMP Sequence Number	Target External Address	Lifetime	Metric	Originator Mesh STA Address	Originator HWMP Sequence Number
Octets:	1	1	1	1	1	6	4	0 or 6	4	4	6	4

Figure 8-392—PREP element format

The Element ID field is set to the value given in Table 8-54 for this element.

The Length field is set to 31 or 37 octets.

The format of the Flags field is shown in Figure 8-393.

	B0 B5	B6	B7
	Reserved	AE	Reserved
Bits:	6	1	1

Figure 8-393—Flags field format

The Flags field is set as follows:

- Bit 0–5: Reserved.
- Bit 6: AE (Address Extension) subfield (1 = external address present, 0 = otherwise). An AE subfield equal to 1 indicates that the field Target External Address is present, and that the target mesh STA is a proxy for this external address.
- Bit 7: Reserved.

The Hop Count field is coded as an unsigned integer and is set to the number of hops from the path target to the mesh STA transmitting this element.

The Element TTL field is coded as an unsigned integer and indicates the remaining number of hops allowed for this element.

The Target Mesh STA Address is the MAC address of the target mesh STA or target proxy mesh gate and is represented as a 48-bit MAC address.

The Target HWMP Sequence Number field is coded as an unsigned integer and is set to the HWMP SN of the target mesh STA (if the AE subfield in the Flags field is set to 0) or target proxy mesh gate (if the AE subfield in the Flags field is set to 1).

The Target External Address field is set to the external address on behalf of which the PREP is sent. This field is present only if Bit 6 (AE subfield) in Flags field equals 1 and is represented as a 48-bit MAC address.

The Lifetime field is coded as an unsigned integer and is set to the time for which mesh STAs receiving the PREP consider the forwarding information to be valid. The lifetime is measured in TUs.

The Metric field indicates the cumulative metric from the path target to the mesh STA transmitting this element.

The Originator Mesh STA Address field is represented as a 48-bit MAC address and is set to the MAC address of the originator, which is contained in the PREQ.

The Originator HWMP Sequence Number field is coded as an unsigned integer and is set to the HWMP SN of the originator mesh STA contained in the PREQ.

The detailed usage of the PREP element is described in 13.10.10.

8.4.2.117 PERR element

The PERR (path error) element is used for announcing an unreachable destination. The PERR element is transmitted in an HWMP Mesh Path Selection frame (see 8.5.17.3). The format of the PERR element is shown in Figure 8-394.

	Element ID	Length	Element TTL	Number of Destinations N	Flags #1	Destination Address #1	HWMP Sequence Number #1	Destination External Address #1	Reason Code #1	...
Octets:	1	1	1	1	1	6	4	0 or 6	2	

Figure 8-394—PERR element format

The Element ID field is set to the value given in Table 8-54 for this element.

The Length field is variable and set to (2 + 13 × Number of Destinations) or to (2 + 19 × Number of Destinations) octets.

The Element TTL field is coded as an unsigned integer and indicates the remaining number of hops allowed for this element.

The Number of Destinations N field is coded as an unsigned integer and indicates the number of announced destinations in PERR. The maximum value of N is 19. The Flags field, the Destination Address field, the HWMP Sequence Number field, the Destination External Address field, and the Reason Code field are repeated N times in the element.

The format of the Flags field is shown in Figure 8-395.

	B0 B5	B6	B7
	Reserved	AE	Reserved
Bits:	4	1	1

Figure 8-395—Flags field format

The Flags field is set as follows:

— Bit 0–5: Reserved.

— Bit 6: AE (Address Extension) subfield (1 = destination external address is present, 0 = otherwise).

— Bit 7: Reserved.

The Destination Address field is represented as a 48-bit MAC address and indicates the detected unreachable destination MAC address.

The HWMP Sequence Number field is coded as an unsigned integer and indicates the HWMP SN for the invalidated destination, if applicable. Otherwise, the HWMP Sequence Number field is reserved depending on the reason code.

The Destination External Address field is set to the external address, on behalf of which the PERR is sent. This field is present only if Bit 6 (AE subfield) in the Flags field equals 1 and is represented as a 48-bit MAC address.

The Reason Code field specifies the reason for sending a PERR element. The Reason Code is defined in 8.4.1.7.

The detailed usage of the PERR element is described in 13.10.11.

8.4.2.118 PXU element

The PXU (proxy update) element is used to inform the destination mesh STA of the proxy information at the originator mesh STA. The PXU element is transmitted in a Proxy Update frame (see 8.5.18.2). The format of the PXU element is shown in Figure 8-396.

	Element ID	Length	PXU ID	PXU Originator MAC Address	Number of Proxy Information (N)	Proxy Information #1	...	Proxy Information #N
Octets:	1	1	1	6	1	11, 15, 17, or 21		11, 15, 17, or 21

Figure 8-396—PXU element format

The Element ID field is set to the value given in Table 8-54 for this element.

The Length field is set to 8 + length of N Proxy Information fields.

The PXU ID field is coded as an unsigned integer and is set to the sequence number of the PXU. The source mesh STA sets the PXU ID field in the PXU element to a value from a single modulo-256 counter that is incremented by 1 for each new PXU element.

The PXU Originator MAC Address field is represented as a 48-bit MAC address and is the MAC address of the mesh STA that originates this PXU element.

The Number of Proxy Information fields is coded as an unsigned integer and is set to the number N of Proxy Information field that follow this field and that are reported to the destination mesh STA. The maximum value of N is 22.

The Proxy Information field contains a single proxy information (see 13.11.4.2). The length of the Proxy Information field depends on the settings of the subfields in the Flags subfield and is 11, 15, 17, or 21 octets.

The format of the Proxy Information field is defined in Figure 8-397.

	Flags	External MAC Address	Proxy Information Sequence Number	Proxy MAC Address	Proxy Information Lifetime
Octets:	1	6	4	0 or 6	0 or 4

Figure 8-397—Proxy Information field

The format of the Flags subfield is shown in Figure 8-398.

	B0	B1	B2	B3 B7
	Delete	Originator Is Proxy	Lifetime	Reserved
Bits:	1	1	1	5

Figure 8-398—Flags subfield

The Flags subfield is set as follows:

- Bit 0: The Delete subfield indicates whether this proxy information is to be deleted. It is set to 1 if the proxy information is to be deleted, and set to 0 otherwise.
- Bit 1: The Originator Is Proxy subfield indicates that the originator mesh STA of the PXU element is the proxy mesh gate of this proxy information when set to 1. In this case, there is no Proxy MAC Address subfield present in this Proxy Information field. When the Originator Is Proxy subfield is 0, the Proxy MAC Address subfield is present in this Proxy Information field.
- Bit 2: The Lifetime subfield indicates that the Proxy Information Lifetime subfield is present in this Proxy Information field when set to 1.
- Bit 3–7: Reserved.

The External MAC Address subfield is represented as a 48-bit MAC address and is the MAC address of the external STA proxied by the proxy mesh gate.

The Proxy Information Sequence Number field is coded as an unsigned integer and is set to the sequence number of the proxy information. The sequence number of the proxy information defines a chronological order of the proxy information for the external STA at this proxy mesh gate.

The Proxy MAC Address subfield is represented as a 48-bit MAC address and is set to the MAC address of proxy mesh gate. It is only present if the Originator Is Proxy subfield of the Flags subfield is 0.

The Proxy Information Lifetime subfield is coded as an unsigned integer and is set to the time for which the mesh STA receiving this PXU considers this proxy information to be valid. The proxy information lifetime is measured in TUs. It is only present if the Lifetime subfield of the Flags subfield is 1.

8.4.2.119 PXUC element

The PXUC (proxy update confirmation) element is used to confirm the previously received PXU. The PXUC element is transmitted in a Proxy Update Confirmation frame (see 8.5.18.3). The format of PXUC element is shown in Figure 8-399.

	Element ID	Length	PXU ID	PXU Recipient MAC Address
Octets	1	1	1	6

Figure 8-399—PXUC element format

The Element ID field is set to the value given in Table 8-54 for this element.

The Length field is set to 7.

The PXU ID field is coded as an unsigned integer and is the PXU ID of the received PXU that is being confirmed.

The PXU Recipient MAC Address is represented as a 48-bit MAC address and is set to the MAC address of the recipient of the PXU, i.e., the originator of the PXUC element.

8.4.2.120 Authenticated Mesh Peering Exchange element

The Authenticated Mesh Peering Exchange element includes information needed to perform the authentication sequence during an authenticated mesh peering exchange. This element is shown in Figure 8-400.

	Element ID	Length	Selected Pairwise Cipher Suite	Local Nonce	Peer Nonce	Key Replay Counter	GTKdata	IGTKdata (optional)
Octets	1	1	4	32	32	8	variable	variable

Figure 8-400—Authenticated Mesh Peering Exchange element format

The Element ID field is set to the value given in Table 8-54 for this element.

The Length field is variable and indicates the number of octets in the information field (fields following the Element ID and Length fields).

The Selected Pairwise Cipher Suite field contains a pairwise cipher suite selector, as defined in 8.4.2.27.2, indicating a cipher suite to be used to secure the link.

The Local Nonce field contains a nonce value chosen by the mesh STA that is sending the element. It is encoded following the conventions from 8.2.2.

The Peer Nonce field contains a nonce value that was chosen by the peer mesh STA or candidate peer mesh STA to which the element is being sent. It is encoded following the conventions from 8.2.2.

The Key Replay Counter field is optional. It is only used for the Mesh Group Key Inform frame (see 13.6.3) and the Mesh Group Key Acknowledge frame (see 13.6.4). It is represented as an unsigned binary number.

The GTKdata field is optional. When present, it contains the bit string of {GTK || Key RSC || GTKExpirationTime} as the GTK data material. When present, the GTKdata field is protected by the exchange in which it is contained (see 13.5). The Key RSC denotes the last frame sequence number sent using the GTK and is specified in Table 11-5 of 11.6.2. GTKExpirationTime denotes the key lifetime of the GTK in seconds and the format is specified in Figure 11-36 of 11.6.2.

The IGTKdata field is present when dot11RSNAProtectedManagementFramesActivated equals true. When present, it contains the KeyID, IPN and IGTK used with BIP for management frame protection. The format of the IGTKdata field is specified in Figure 11-38 of 11.6.2.

Detailed usage of the Authenticated Mesh Peering Exchange element is described in 13.5.5 and in 13.6.

8.4.2.121 MIC element

The MIC element provides message integrity to Mesh Peering Management frames. The format of the MIC element is shown in Figure 8-401.

	Element ID	Length	MIC
Octets:	1	1	16

Figure 8-401—MIC element format

The Element ID field is set to the value given in Table 8-54 for this element.

The Length field is set to 16.

The MIC field contains a message integrity code calculated over the Mesh Peering Management frame (as specified in 13.5) and the mesh group key handshake frame (as specified in 13.6).

8.4.3 Information Subelements

Subelements are defined to have a common general format consisting of a 1-octet element-specific Subelement ID field, a 1-octet Length field, and a variable-length subelement-specific Data field. Each subelement is assigned a subelement ID that is unique within the containing element or subelement. The Length field specifies the number of octets in the Data field. See Figure 8-402. Subelements are ordered by nondecreasing Subelement ID. See 9.24.9.

	Subelement ID	Length	Data
Octets:	1	1	Variable

Figure 8-402—Subelement format

8.4.4 Access Network Query Protocol (ANQP) elements

8.4.4.1 General

ANQP-elements are defined to have a common format consisting of a 2-octet Info ID field (information identifier), a 2-octet Length field, and a variable-length element-specific Information field. Each element is assigned a unique Info ID as defined in this standard. The ANQP-element format is shown in Figure 8-403. See V.2 for informative text on ANQP usage.

	Info ID	Length	Information
Octets:	2	2	variable

Figure 8-403—ANQP-element format

Each ANQP-element in 8.4.4 is assigned a unique 2-octet Info ID. The set of valid Info IDs are defined in Table 8-184. The 2-octet Info ID field is encoded following the conventions given in 8.2.2.

Table 8-184—ANQP-element definitions

ANQP-element name	InfoID	ANQP-element (subclause)
Reserved	0–255	n/a
Query List	256	8.4.4.2
Capability List	257	8.4.4.3
Venue Name	258	8.4.4.4
Emergency Call Number	259	8.4.4.5
Network Authentication Type	260	8.4.4.6
Roaming Consortium	261	8.4.4.7
IP Address Type Availability	262	8.4.4.9
NAI Realm	263	8.4.4.10
3GPP Cellular Network	264	8.4.4.11
AP Geospatial Location	265	8.4.4.12
AP Civic Location	266	8.4.4.13
AP Location Public Identifier URI	267	8.4.4.14
Domain Name	268	8.4.4.15
Emergency Alert Identifier URI	269	8.4.4.16
TDLS Capability	270	8.4.4.18
Emergency NAI	271	8.4.4.17
Neighbor Report	272	8.4.4.19
Reserved	273– 56796	n/a
Vendor Specific	56797	8.4.4.8
Reserved	56798–65535	n/a

The Length field specifies the number of octets in the Information field and is encoded following the conventions given in 8.2.2.

The ANQP-elements that may be configured are shown in Table 8-184. If information is not configured for a particular ANQP-element, then a query for that element will return that element with all optional fields not present.

8.4.4.2 Query List ANQP-element

The Query List ANQP-element provides a list of identifiers of ANQP-elements for which the requesting STA is querying. Each ANQP-element may be returned in response to an Query List ANQP-element using the procedures in 10.24.3.2.2).

The format of the Query List ANQP-element is provided in Figure 8-404.

	Info ID	Length	ANQP Query ID #1	…	ANQP Query ID #N (optional)
Octets:	2	2	2	…	0 or 2

Figure 8-404—Query List ANQP-element format

The Info ID field is a 2-octet field whose value is drawn from Table 8-184 corresponding to the Query List ANQP-element.

The Length field is a 2-octet field whose value is set to 2 times the number of ANQP Query ID fields.

Each ANQP Query ID field value is an Info ID drawn from Table 8-184. Including an Info ID in the Query List ANQP-element declares that the STA performing the ANQP query is requesting the ANQP-element corresponding to that Info ID be returned in the ANQP query response. The Info IDs included in the Query List ANQP-element are ordered by monotonically increasing Info ID value. The ANQP query response is defined in 10.24.3.2.1.

8.4.4.3 Capability List ANQP-element

The Capability List ANQP-element provides a list of information/capabilities that has been configured on a STA. The Capability List ANQP-element is returned in response to a Query List ANQP-element containing the Info ID of the Capabililty List ANQP-element.

The format of the Capability List ANQP-element is provided in Figure 8-405.

	Info ID	Length	ANQP Capability #1	…	ANQP Capability #N (optional)	Vendor Specific ANQP-element #1 (optional)	…	Vendor Specific ANQP-element #N (optional)
Octets:	2	2	2	…	0 or 2	variable	…	variable

Figure 8-405—Capability List ANQP-element format

The Info ID field is a 2-octet field whose value is drawn from Table 8-184 corresponding to the Capability List ANQP-element.

The Length field is a 2-octet field whose value is set to 2 times the number of ANQP Capability fields following the Length field plus the sum of the lengths of the Vendor Specific ANQP-elements.

Each ANQP Capability field value is an Info ID drawn from Table 8-184. If included in the Capability List ANQP-element, it declares that a Query List ANQP-element including that Info ID will return the requested ANQP-element. The Info ID for Capability List ANQP-element is always included in the Capability List ANQP-element returned in a GAS Query Response. The list does not include any duplicate Info IDs, except possibly the Info ID for the Vendor Specific ANQP-element. The Info IDs returned in the Capability List ANQP-element are ordered by nondecreasing Info ID value.

The Vendor Specific ANQP-element is defined in 8.4.4.8. The Vendor Specific ANQP-element is structured such that the first 2 octets of the Vendor Specific ANQP-element is the Info ID whose value corresponds to the Vendor Specific ANQP-element (see Table 8-184). When a Vendor Specific ANQP-element is present in the Capability List ANQP-element, the Vendor Specific ANQP-element element contains the capabilities of that vendor-specific query protocol.

8.4.4.4 Venue Name ANQP-element

The Venue Name ANQP-element provides zero or more venue names associated with the BSS. The format of the Venue Name ANQP-element is shown in Figure 8-406. The Venue Name ANQP-element may be used to provide additional metadata on the BSS. For example, the information may be used to assist a user in selecting the appropriate BSS with which to associate. Zero or more Venue Name fields may be included in the same or different languages.

	Info ID	Length	Venue Info	Venue Name Duple #1 (optional)	…	Venue Name Duple #N (optional)
Octets:	2	2	2	variable	…	variable

Figure 8-406—Venue Name ANQP-element format

The Info ID field is equal to the value in Table 8-184 corresponding to the Venue Name ANQP-element.

The Length field is a 2-octet field whose value is set to two plus the total length of the Venue Name Duple fields.

The Venue Info field is a 2-octet field and is defined in 8.4.1.34.

The format of the Venue Name Duple field is shown in Figure 8-407.

	Length	Language Code	Venue Name
Octets:	1	3	variable

Figure 8-407—Venue Name Duple field

— The Length field is a 1-octet field whose value is equal to 3 plus the number of octets in the Venue Name field.

— The Language Code is a 3-octet ISO-14962-1997 [B45] encoded string field that defines the language used in the Venue Name field. The Language Code field is a two or three character language code selected from ISO-639 [B44]. A two character language code has 0 (“null” in ISO-14962-1997) appended to make it 3 octets in length.

— The Venue Name is a variable-length UTF-8 formatted field containing the venue's name. The maximum length of this field is 252 octets. UTF-8 format is defined in IETF RFC 3629.

8.4.4.5 Emergency Call Number ANQP-element

The Emergency Call Number ANQP-element provides a list of emergency phone numbers to an emergency responder, such as directed by a public safety answering point (PSAP), that is used in the geographic location of the STA. The format of the Emergency Call Number ANQP-element is provided in Figure 8-408.

	Info ID	Length	Emergency Call Number Unit #1 (optional)	…	Emergency Call Number Unit #N (optional)
Octets:	2	2	variable	…	variable

Figure 8-408—Emergency Call Number ANQP-element format

The Info ID field is equal to the value in Table 8-184 corresponding to the Emergency Call Number ANQP-element.

The Length field is a 2-octet field whose value is set to the total length of the Emergency Call Number Unit fields.

Each Emergency Call Number Unit field has the structure shown in Figure 8-409.

	Length of Emergency Call Number	Emergency Call Number
Octets:	1	variable

Figure 8-409—Emergency Call Number Unit field format

The Length of Emergency Call Number field is a 1-octet field whose value is set to the length of the Emergency Call Number field.

The Emergency Call Number field is a variable-length UTF-8 formatted field containing information, used to reach emergency services, from the network (e.g., dialed digits, emergency service URN label [B41]). UTF-8 format is defined in IETF RFC 3629.

8.4.4.6 Network Authentication Type ANQP-element

The Network Authentication Type ANQP-element provides a list of authentication types when ASRA is set to 1. The format of the Network Authentication Type ANQP-element is shown in Figure 8-410.

	Info ID	Length	Network Authentication Type Unit #1 (optional)	…	Network Authentication Type Unit #N (optional)
Octets:	2	2	variable	…	variable

Figure 8-410—Network Authentication Type ANQP-element format

The Info ID field is equal to the value in Table 8-184 corresponding to the Network Authentication Type ANQP-element.

The Length field is a 2-octet field whose value is set to the total length of the Network Authentication Type Units fields.

Each Network Authentication Type Unit has the structure shown in Figure 8-411.

	Network Authentication Type Indicator	Redirect URL Length	Redirect URL (optional)
Octets:	1	2	variable

Figure 8-411—Network Authentication Type Unit field format

The Network Authentication Type Indicator field is a 1-octet field and has one of the values shown in Table 8-185.

Table 8-185—Network Authentication Type Indicator definitions

Value	Meaning
0	Acceptance of terms and conditions
1	On-line enrollment supported
2	http/https redirection
3	DNS redirection
4–255	Reserved

Each Network Authentication Type Indicator defines additional information that may be communicated.

If the Network Authentication Type Indicator is 0, the network requires the user to accept terms and conditions. The Redirect URL can be used by the non-AP STA to obtain the terms and conditions. If the Redirect URL is not present, then, the Redirect URL Length is set to 0.

If the Network Authentication Type Indicator is 1, the network supports on-line enrollment. Higher layer protocols on the non-AP STA may indicate to the user that accounts may be created. When the Network Authentication Type Indicator is 1, the Redirect URL Length is set to 0 and the Redirect URL is not present.

If the Network Authentication Type Indicator is 2, the network infrastructure performs http/https redirect. The ReDirect URL is used by the non-AP STA to perform additional steps required for network access.

If the Network Authentication Type Indicator is 3, the network supports DNS redirection. Higher layer software on the non-AP STA will exchange credentials with the network, the Redirect URL Length is set to 0 and the Redirect URL is not present.

The Redirect URL Length field is a 2-octet field whose value is the length of the Redirect URL. The value of the Redirect URL Length field is set to 0 whenever the Redirect URL is not present.

The Redirect URL field is a variable-length field that is optionally included if the Network Authentication Type Indicator is either 0 or 2. If the Network Authentication Type Indicator is other than 0 or 2, a Redirect URL is not included. The URL is formatted in accordance with IETF RFC 3986.

8.4.4.7 Roaming Consortium ANQP-element

The Roaming Consortium ANQP-element provides a list of information about the Roaming Consortium and/or SSPs whose networks are accessible via this AP. This list may be returned in response to a GAS Query using procedures in 10.24.3.2.3. The format of the Roaming Consortium ANQP-element is provided in Figure 8-412.

	Info ID	Length	OI Duple #1 (optional)	…	OI Duple #N (optional)
Octets:	2	2	variable		variable

Figure 8-412—Roaming Consortium ANQP-element format

The Info ID field is equal to the value in Table 8-184 corresponding to the Roaming Consortium ANQP-element.

The Length field is a 2-octet field whose value is set to the total length of the OI Duple fields.

OIs contained within the Roaming Consortium element (see 8.4.2.98) are also included in this ANQP-element. The value of the OI subfield(s) in this ANQP-element are equal to the values of the OI(s) in the dot11RoamingConsortiumTable.

The format of the OI Duple field is provided in Figure 8-413.

- The value of the OI Length field is equal to the number of octets in the OI field.
- The OI field is defined in 8.4.1.31. Each OI identifies a roaming consortium (group of SSPs with inter-SSP roaming agreement) or a single SSP.

	OI Length	OI
Octets:	1	variable

Figure 8-413—OI Duple field format

8.4.4.8 Vendor Specific ANQP-element

The Vendor Specific ANQP-element is used to query for information not defined in this standard within a single defined format, so that reserved Info IDs are not usurped for nonstandard purposes and interoperability is more easily achieved in the presence of nonstandard information. The ANQP-element is in the format shown in Figure 8-414.

	Info ID	Length	OI	Vendor Specific Content
Octets:	2	2	variable	variable

Figure 8-414—Vendor Specific ANQP-element format

The Info ID field is equal to the value in Table 8-184 corresponding to the Vendor Specific ANQP-element.

The Length field is a 2-octet field whose value is set to the length of the OI field plus the length of the Vendor Specific Content field.

The OI field is defined in 8.4.1.31.

The Vendor Specific Content field is a variable-length field whose content is defined by the entity identified in the OI field.

8.4.4.9 IP Address Type Availability ANQP-element

The IP Address Type Availability ANQP-element provides STA with the information about the availability of IP address version and type that could be allocated to the STA after successful association. The format of the IP Address Type Availability ANQP-element is shown in Figure 8-415.

	Info ID	Length	IP Address
Octets:	2	2	1

Figure 8-415—IP Address Type Availability ANQP-element

The Info ID field is equal to the value in Table 8-184 corresponding to the IP Address Type Availability ANQP-element.

The Length field is a 2-octet field whose value is set to 1.

The format of the IP Address field shown in Figure 8-416.

Bits:	B0B1	B2B7
	IPv6 Address	IPv4 Address

Figure 8-416—IP Address field format

The IPv6 Address field values are shown in Table 8-186.

Table 8-186—IPv6 Address field values

Address value	Meaning
0	Address type not available
1	Address type available
2	Availability of the address type not known
3	Reserved

The IPv4 Address field values are shown in Table 8-187.

Table 8-187— IPv4 Address field values

Address value	Meaning
0	Address type not available
1	Public IPv4 address available
2	Port-restricted IPv4 address available
3	Single NATed private IPv4 address available
4	Double NATed private IPv4 address available
5	Port-restricted IPv4 address and single NATed IPv4 address available
6	Port-restricted IPv4 address and double NATed IPv4 address available
7	Availability of the address type is not known
8–63	Reserved

8.4.4.10 NAI Realm ANQP-element

The NAI Realm ANQP-element provides a list of network access identifier (NAI) realms corresponding to SSPs or other entities whose networks or services are accessible via this AP; optionally included for each NAI realm is a list of one or more EAP Method subfields, which that NAI realm uses for authentication. The format of the NAI Realm ANQP-element is provided in Figure 8-417.

	Info ID	Length	NAI Realm Count (optional)	NAI Realm Data #1 (optional)	. . .	NAI Realm Data #N (optional)
Octets:	2	2	2	variable		variable

Figure 8-417—NAI Realm ANQP-element format

The Info ID field is equal to the value in Table 8-184 corresponding to the NAI Realm ANQP-element.

The Length field is a 2-octet field whose value is set to 2 plus the total length of the NAI Realm Data fields.

The NAI Realm Count field is a 2-octet field that specifies the number of NAI realms included in the NAI Realm ANQP-element.

The format of the NAI Realm Data field is shown in Figure 8-418.

	NAI Realm Data Field Length	NAI Realm Encoding	NAI Realm Length	NAI Realm	EAP Method Count	EAP Method #1 (optional)	. . .	EAP Method #N (optional)
Octets:	2	1	1	variable	1	variable		variable

Figure 8-418—NAI Realm Data field format

The NAI Realm Data Field Length is a 2-octet subfield whose value is equal to 3 plus the length of the NAI Realm subfield plus the sum of the lengths of the EAP Method subfields.

The NAI Realm Encoding is a 1-octet subfield whose format is shown in Figure 8-419.

Bits:	B0	B1B7
	NAI Realm Encoding Type	Reserved

Figure 8-419—NAI Realm Encoding subfield format

The NAI Realm Encoding Type is a 1-bit subfield. It is set to 0 to indicate that the NAI Realm in the NAI Realm subfield is formatted in accordance with IETF RFC 4282. It is set to 1 to indicate it is a UTF-8 formatted character string that is not formatted in accordance with IETF RFC 4282.

NOTE—This encoding is to facilitate roaming consortium or other entities that use nonstandard NAI Realm formats.

NAI Realm Length subfield is a 1-octet subfield whose value is the length in octets of the NAI Realm subfield.

The NAI Realm subfield is one or more NAI Realms formatted as defined in the NAI Realm Encoding Type bit of the NAI Realm Encoding subfield. If there is more than one NAI Realm in this subfield, the NAI Realms are delimited by a semi-colon character (i.e., ";", which is encoded in UTF-8 format as 0x3B). All the realms included in the NAI Realm subfield support all the EAP methods identified by the EAP Method subfields, if present. The maximum length of this subfield is 255 octets.

The EAP Method Count specifies the number of EAP methods subfields for the NAI realm. If the count is 0, there is no EAP method information provided for the NAI realm.

The format of the optional EAP Method subfield is shown in Figure 8-420. Each EAP Method subfield contains a set of Authentication Parameters associated with the EAP-Method.

	Length	EAP Method	Authentication Parameter Count	Authentication Parameter #1 (optional)	. . .	Authentication Parameter #N (optional)
Octets:	1	1	1	variable		variable

Figure 8-420—EAP Method subfield format

The Length subfield is a 1-octet subfield whose value is equal to 2 plus the length of the Authentication Parameter subfields.

The EAP method subfield is a 1-octet subfield that is set to the EAP Type value as given in IANA EAP Method Type Numbers.

The Authentication Parameter Count indicates how many additional Authentication Parameter subfields are specified for the supported EAP Method. If the Authentication Parameters Count subfield is 0, there are no Authentication Parameters subfields present, meaning no additional Authentication Parameters are specified for the EAP Method.

The format of the Authentication Parameter subfield is shown in Figure 8-421.

	ID	Length	Authentication Parameter Value
Octets:	1	1	variable

Figure 8-421—Authentication Parameter subfield format

The ID subfield is a 1-octet subfield that indicates the type of authentication information provided.

The Length subfield is a 1-octet subfield whose value is set to the length of the Authentication Parameter Value subfield.

The Authentication Parameter Value subfield is a variable-length subfield containing the value of the parameter indicated by the ID.

The ID and its associated formats are specified in Table 8-188. Each ID indicates a different type of information. Use of multiple Authentication Parameter subfields allows all the required authentication parameter requirements to be provided.

Table 8-188—Authentication Parameter types

Authentication information	ID	Description	Length (octets)
Reserved	0		
Expanded EAP Method	1	Expanded EAP Method Subfield	7
Non-EAP Inner Authentication Type	2	Enum (0 - Reserved, 1 - PAP, 2 - CHAP, 3 - MSCHAP, 4 - MSCHAPV2)	1
Inner Authentication EAP Method Type	3	Value drawn from IANA EAP Method Type Numbers	1
Expanded Inner EAP Method	4	Expanded EAP Method Subfield	7
Credential Type	5	Enum (1 - SIM, 2 - USIM, 3 - NFC Secure Element, 4 - Hardware Token, 5 - Softoken, 6 - Certificate, 7 - username/password, 8 - none*, 9 - Reserved, 10 - Vendor Specific) *none means server-side authentication only	1
Tunneled EAP Method Credential Type	6	Enum (1 - SIM, 2 - USIM, 3 - NFC Secure Element, 4 - Hardware Token, 5 - Softoken, 6 - Certificate, 7 - username/password, 8 - Reserved, 9 - Anonymous, 10 - Vendor Specific)	1
Reserved	7–220		
Vendor Specific	221	Variable	variable
Reserved	222–255		

If the EAP Method type is an Expanded EAP type (the EAP Method value is 254), the Authentication Parameter is used to specify additional information on the EAP method. Table 8-189 describes the Authentication Parameter format for the Expanded EAP method; values for the Vendor ID and Vendor Type are specified in IETF RFC 3748. The Vendor ID and Vendor Type fields are expressed in big endian byte order.

Table 8-189—Authentication Parameter format for the Expanded EAP method

Parameters	Length (octets)
ID	1
Length	1
Vendor ID	3
Vendor Type	4

The Non-EAP Inner Authentication Type is specified as single enumerated value given in Table 8-188. This Authentication Information type is used for non-EAP Inner Authentication methods. The possible values are PAP (as specified in IETF RFC 1334), CHAP (as specified in IETF RFC 1994), MSCHAP (as specified in IETF RFC 2433), and MSCHAPv2 (as specified in IETF RFC 2759).

The Inner Authentication EAP Method Type is specified as the EAP number as defined in IANA EAP Method Type Numbers. This Authentication Information type is used when the Inner Authentication method is an EAP method. If the Inner Authentication EAP Method Type is equal to 254 indicating an Expanded EAP Type, then the Expanded EAP Method Authentication Parameter is included.

A Credential Type is specified as a single enumerated value as shown in Table 8-188. If the value is equal to the "Vendor Specific" value, then a Vendor-Specific Authentication Parameter is included.

Vendor-Specific Authentication Parameters are specified as shown in Table 8-190.

Table 8-190—Vendor Specific Authentication Parameters

Parameters	Length (octets)
ID	1
Length	1
OI	variable
Authentication Parameter Value	Vendor-specific content

8.4.4.11 3GPP Cellular Network ANQP-element

The 3GPP Cellular Network ANQP-element contains cellular information such as network advertisement information e.g., network codes and country codes to assist a 3GPP non-AP STA in selecting an AP to access 3GPP networks. The format of the 3GPP Cellular Network ANQP-element is shown in Figure 8-422.

	Info ID	Length	Payload (optional)
Octets:	2	2	variable

Figure 8-422—3GPP Cellular Network ANQP-element format

The Info ID field is equal to the value in Table 8-184 corresponding to the 3GPP Cellular Network ANQP-element.

The Length field is a 2-octet field whose value is set to the length of the Payload field.

The Payload field is a variable-length field and is a generic container. An example of the format and content is defined in Annex A of 3GPP TS 24.234.

8.4.4.12 AP Geospatial Location ANQP-element

The AP Geospatial Location ANQP-element provides the AP's location in LCI format; see 8.4.2.24.10.

The format of the AP Geospatial Location ANQP-element is provided in Figure 8-423.

	Info ID	Length	Location Configuration Report
Octets:	2	2	18

Figure 8-423—AP Geospatial Location ANQP-element format

The Info ID field is equal to the value in Table 8-184 corresponding to the AP Geospatial Location ANQP-element.

The Length field is a 2-octet field whose value is set to 18.

The Location Configuration Report is an 18-octet field and the format is provided in 8.4.2.24.10. There are no Optional Subelements field present in the Location Configuration Report when it is used in the AP Geospatial Location ANQP-element. This information is taken from the dot11APLCITable MIB object.

8.4.4.13 AP Civic Location ANQP-element

The AP Civic Location ANQP-element provides the AP's location in Civic format. The format of the AP Civic Location ANQP-element is provided in Figure 8-424.

	Info ID	Length	Location Civic Report
Octets:	2	2	variable

Figure 8-424—AP Civic Location ANQP-element format

The Info ID field is equal to the value in Table 8-184 corresponding to the AP Civic Location ANQP-element.

The Length field is a 2-octet field whose value is set to the length of the Location Civic Report.

The Location Civic Report is a variable-length field and the format is provided in 8.4.2.24.13. This information is taken from the dot11ApCivicLocation MIB object.

8.4.4.14 AP Location Public Identifier URI ANQP-element

The AP Location Public Identifier URI ANQP-element provides an indirect reference to the location information for the AP. This list element may be returned in response to a GAS Query using the procedures in 10.24.3.2. The format of the AP Location Public Identifier URI ANQP-element is provided in Figure 8-425.

	Info ID	Length	Public Identifier URI
Octets:	2	2	variable

Figure 8-425—AP Location Public Identifier URI ANQP-element format

The Info ID field is equal to the value in Table 8-184 corresponding to the AP Location Public Identifier URI.

The Length field is a 2-octet field whose value is set to the length of the AP Location Public Identifier URI field.

The Public Identifier URI field is a variable-length field and is defined in 8.4.2.24.13.

8.4.4.15 Domain Name ANQP-element

The Domain Name ANQP-element provides a list of one or more domain names of the entity operating the IEEE 802.11 access network. Domain Names in this ANQP-element are taken from dot11DomainNameTable. The format of the Domain Name ANQP-element is provided in Figure 8-426.

	Info ID	Length	Domain Name field #1 (optional)	. . .	Domain Name field #N (optional)
Octets:	2	2	variable		variable

Figure 8-426—Domain Name ANQP-element format

The Info ID field is equal to the value in Table 8-184 corresponding to the Domain Name ANQP-element.

The Length field is a 2-octet field whose value is set to the total length of the Domain Name Fields.

The format of the Domain Name subfield is shown in Figure 8-427.

	Length	Domain Name
Octets:	1	variable

Figure 8-427—Domain Name subfield format

The Length subfield is a 1-octet subfield whose value is set to the length in octets of the Domain Name subfield.

The Domain Name subfield is of variable length and contains a domain name compliant with the “Preferred Name Syntax” as defined in IETF RFC 1035. The maximum length of this field is 255 octets.

8.4.4.16 Emergency Alert URI ANQP-element

The Emergency Alert URI ANQP-element provides a URI for EAS message retrieval.

The format of the Emergency Alert URI ANQP-element is provided in Figure 8-428.

	Info ID	Length	Emergency Alert URI
Octets:	2	2	variable

Figure 8-428—Emergency Alert URI ANQP-element format

The Info ID field is equal to the value in Table 8-184 corresponding to the Emergency Alert URI ANQP-element.

The Length field is a 2-octet field whose value is set to the length of the Emergency Alert Identifier URI field.

The Emergency Alert URI field is a variable-length field used to indicate the URI at which an EAS message may be retrieved as described in 10.24.7. The Emergency Alert URI field is formatted in accordance with IETF RFC 3986.

8.4.4.17 Emergency NAI ANQP-element

The Emergency NAI ANQP-element contains an emergency string, which is available for use by a STA as its identity to indicate emergency access request. The format of the Emergency NAI ANQP-element is provided in Figure 8-429.

	Info ID	Length	Emergency NAI Information
Octets:	2	2	variable

Figure 8-429—Emergency NAI ANQP-element format

The Info ID field is equal to the value in Table 8-184 corresponding to the Emergency NAI ANQP-element.

The Length field is a 2-octet field whose value is set to the length of Emergency NAI Information field.

The Emergency NAI Information field is a variable-length UTF-8 field formatted in accordance with IETF RFC 4282. UTF-8 format is defined in IETF RFC 3629.

8.4.4.18 TDLS Capability ANQP-element

The TDLS Capability ANQP-element is used by a STA to discover the TDLS capabilities of a peer STA. The format of the TDLS Capability is provided in Figure 8-430.

	Info ID	Length	Peer Information
Octets:	2	2	variable

Figure 8-430—TDLS Capability ANQP-element format

The Info ID field is equal to the value in Table 8-184 corresponding to the TDLS Capability ANQP-element.

The Length field is a 2-octet field whose value is set to the length of the Peer Information field

The Peer Information field is a variable-length field containing information that a STA can use to establish a TDLS link and is defined as an XML schema (see V.6). An example of the peer information is described in 10.24.3.2.10.

8.4.4.19 Neighbor Report ANQP-element

The Neighbor Report ANQP-element provides zero or more neighbor reports about neighboring APs. This is of benefit to a STA in a preassociated state.

	Info ID	Length	Neighbor Report element (optional)
Octets:	2	2	variable

Figure 8-431—Neighbor Report ANQP-element format

The Info ID field is equal to the value in Table 8-184 corresponding to the Neighbor Report ANQP-element.

The Length field is a 2-octet field whose value is set to the number of octets in the Neighbor Report field.

The format of the Neighbor Report element is shown in Figure 8-215 defined in 8.4.2.39. The Element ID and the Length fields of the Neighbor Report element, as shown in Figure 8-215, are not included.

8.5 Action frame format details

8.5.1 Introduction

Subclause 8.5 describes the Action field formats allowed in each of the action categories defined in Table 8-38 in 8.4.1.11.

8.5.2 Spectrum management Action frames

8.5.2.1 General

Five Action frame formats are defined for spectrum management. A Spectrum Management Action field, in the octet field immediately after the Category field, differentiates the five formats. The Spectrum Management Action field values associated with each frame format are defined in Table 8-191.

Table 8-191—Spectrum Management Action field values

Spectrum Mangement Action field value	Description
0	Measurement Request
1	Measurement Report
2	TPC Request
3	TPC Report
4	Channel Switch Announcement
5–255	Reserved

8.5.2.2 Measurement Request frame format

The Measurement Request frame uses the Action frame body format and is transmitted by a STA requesting another STA to measure one or more channels. The format of the Measurement Request Action field is shown in Figure 8-432.

	Category	Spectrum Management Action	Dialog Token	Measurement Request Elements
Octets:	1	1	1	variable

Figure 8-432—Measurement Request frame Action field format

The Category field is set to 0 (representing spectrum management).

The Spectrum Management Action field is set to 0 (representing a Measurement Request frame).

The Dialog Token field is set to a nonzero value chosen by the STA sending the measurement request to identify the request/report transaction.

The Measurement Request Elements field contains one or more of the Measurement Request elements described in 8.4.2.23. The number and length of the Measurement Request elements in a Measurement Request frame is limited by the maximum allowed MMPDU size.

8.5.2.3 Measurement Report frame format

The Measurement Report frame uses the Action frame body format and is transmitted by a STA in response to a Measurement Request frame or by a STA autonomously providing measurement information. The format of the Measurement Report Action field is shown in Figure 8-433.

	Category	Spectrum Management Action	Dialog Token	Measurement Report Elements
Octets:	1	1	1	variable

Figure 8-433—Measurement Report frame Action field format

The Category field is set to 0 (representing spectrum management).

The Spectrum Management Action field is set to 1 (representing a Measurement Report frame).

The Dialog Token field is set to the value in any corresponding Measurement Request frame. If the Measurement Report frame is not being transmitted in response to a Measurement Request frame, then the Dialog token is set to 0.

The Measurement Report Elements field contains one or more of the Measurement Report elements described in 8.4.2.24. The number and length of the Measurement Report elements in a Measurement Report frame is limited by the maximum allowed MMPDU size.

8.5.2.4 TPC Request frame format

The TPC Request frame uses the Action frame body format and is transmitted by a STA requesting another STA for transmit power and link margin information. The format of the TPC Request Action field is shown in Figure 8-434.

	Category	Spectrum Management Action	Dialog Token	TPC Request element
Octets:	1	1	1	2

Figure 8-434—TPC Request frame Action field format

The Category field is set to 0 (representing spectrum management).

The Spectrum Management Action field is set to 2 (representing a TPC Request frame).

The Dialog Token field is set to a nonzero value chosen by the STA sending the request to identify the transaction.

The TPC Request element is set as described in 8.4.2.18.

8.5.2.5 TPC Report frame format

The TPC Report frame uses the Action frame body format and is transmitted by a STA in response to a TPC Request frame. The format of the TPC Report Action field is shown in Figure 8-435.

	Category	Spectrum Management Action	Dialog Token	TPC Report element
Octets:	1	1	1	4

Figure 8-435—TPC Report frame Action field format

The Category field is set to 0 (representing spectrum management).

The Spectrum Management Action field is set to 3 (representing a TPC Report frame).

The Dialog Token field is set to the Dialog Token value in the corresponding TPC Request frame.

The TPC Report element is set as described 8.4.2.19.

8.5.2.6 Channel Switch Announcement frame format

The Channel Switch Announcement frame uses the Action frame body format and is transmitted by an AP in a BSS, a STA in an IBSS, or a mesh STA in an MBSS to advertise a channel switch. The format of the Channel Switch Announcement Action field is shown in Figure 8-436.

	Category	Spectrum Management Action	Channel Switch Announcement element	Secondary Channel Offset element	Mesh Channel Switch Parameters element
Octets:	1	1	5	3	6

Figure 8-436—Channel Switch Announcement frame Action field format

The Category field is set to 0 (representing spectrum management).

The Spectrum Management Action field is set to 4 (representing a Channel Switch Announcement frame).

The Channel Switch Announcement element is set as described 8.4.2.21.

The Secondary Channel Offset element is defined in 8.4.2.22. This element is present when switching to a 40 MHz channel. It may be present when switching to a 20 MHz channel (in which case the secondary channel offset is set to SCN).

The Mesh Channel Switch Parameters element is defined in 8.4.2.105. This element is present when a mesh STA performs MBSS channel switch. The Mesh Channel Switch Parameters element is not included for channel switch other than MBSS.

8.5.3 QoS Action frame details

8.5.3.1 General

Several Action frame formats are defined for QoS purposes. These frames are identified by the single octet QoS Action field, which follows immediately after the Category field. The values of the QoS Action field are defined in Table 8-192.

Table 8-192—QoS Action field values

QoS Action field value	Meaning
0	ADDTS Request
1	ADDTS Response
2	DELTS
3	Schedule
4	QoS Map Configure
5–255	Reserved

8.5.3.2 ADDTS Request frame format

The ADDTS frames are used to carry TSPEC and optionally TCLAS elements to set up and maintain TSs using the procedures defined in 10.4.

The Action field of the ADDTS Request frame contains the information shown in Table 8-193.

Table 8-193—ADDTS Request frame Action field format

Order	Information
1	Category
2	QoS Action
3	Dialog Token
4	TSPEC
5–n	TCLAS (optional)
n + 1	TCLAS Processing (optional)
n + 2	U-APSD Coexistence (optional)
n + 3	Expedited Bandwidth Request element (optional)

The Category field is set to 1 (representing QoS).

The QoS Action field is set to 0 (representing ADDTS request).

The Dialog Token, TCLAS, and TCLAS Processing fields of this frame are contained in an MLME-ADDTS.request primitive that causes the frame to be sent. Some of the TSPEC parameters are contained in the MLME-ADDTS.request primitive while the other parameters (i.e., Surplus Bandwidth Allowance, Minimum Service Interval, Maximum Service Interval, and Minimum PHY Rate) are generated within the MAC.

The TSPEC element, defined in 8.4.2.32, and the optional TCLAS element, defined in 8.4.2.33, contain the QoS parameters that define the TS. The TS is identified by the TSID and Direction fields within the TSPEC element. The TCLAS element is optional at the discretion of the STA that sends the ADDTS Request frame, regardless of the setting of the access policy (EDCA or HCCA). There may be one or more TCLAS elements in the ADDTS frame. The TCLAS Processing element is present when there are more than one TCLAS element and is defined in 8.4.2.35. There may be one Expedited Bandwidth Request element, which is defined in 8.4.2.96.

The U-APSD Coexistence element, defined in 8.4.2.93, contains the coexistence parameters requested by the non-AP STA when using the U-APSD Coexistence capability as described in 10.2.1.5.2. The U-APSD Coexistence element is optionally present.

8.5.3.3 ADDTS Response frame format

The ADDTS Response frame is transmitted in response to an ADDTS Request frame. The Action field of the ADDTS Response frame contains the information shown in Table 8-194.

Table 8-194—ADDTS Response frame Action field format

Order	Information
1	Category
2	QoS Action
3	Dialog Token

Table 8-194—ADDTS Response frame Action field format *(continued)*

Order	Information
4	Status Code
5	TS Delay
6	TSPEC
7–n	TCLAS (optional)
n + 1	TCLAS Processing (optional)
n + 2	Schedule
n + 3	Expedited Bandwidth Request (optional)

The Category field is set to 1(representing QoS).

The QoS Action field is set to 1 (representing ADDTS response).

The Status Code field is defined in 8.4.1.9.

The Dialog Token, TS Delay, TSPEC, TCLAS, TCLAS Processing, and Expedited Bandwidth Request fields in this frame are contained in an MLME-ADDTS.response primitive that causes the frame to be sent. The TS Delay element is present in an ADDTS Response frame only if the status code is equal to 47.

The Schedule element, defined in 8.4.2.36, is present in an ADDTS Response frame only if the status code is equal to 0 (i.e., when the TS is admitted).

8.5.3.4 DELTS frame format

The DELTS frame is used to delete a TS using the procedures defined in 10.4.9.

The Action field of a DELTS frame contains the information shown in Table 8-195.

Table 8-195—DELTS frame Action field format

Order	Information
1	Category
2	QoS Action
3	TS Info
4	Reason Code

The Category field is set to 1 (representing QoS).

The QoS Action field is set to 2 (representing DELTS).

The TS Info field is defined in 8.4.2.32.

The Reason Code field is defined in 8.4.1.7.

A DELTS frame is used to delete a TS characterized by the TS Info field in the frame. A DELTS frame may be sent from the HC to the source STA of that TS, or vice versa, to indicate an imperative request, to which no response is required from the recipient STA.

8.5.3.5 Schedule frame format

The Schedule frame is transmitted by the HC to announce the schedule of delivery of data and polls. The Action field of the Schedule frame contains the information shown in Table 8-196.

Table 8-196—Schedule frame Action field format

Order	Information
1	Category
2	QoS Action
3	Schedule

The Category field is set to 1 (representing QoS).

The QoS Action field is set to 3 (representing Schedule).

The Schedule element is defined in 8.4.2.36.

8.5.3.6 QoS Map Configure frame format

The QoS Map Configure frame is used by an AP to provide the QoS Map Set to a non-AP STA using the procedures defined in 10.24.9.

The frame body of the QoS Map Configure frame contains the information shown in Table 8-197.

Table 8-197—QoS Map configure frame body

Order	Information
0	Category
1	Action
2	QoS Map Set

The Category field is set to the value in Table 8-38.

The Action field is set to the value in Table 8-192.

The QoS Map Set element is defined in 8.4.2.97.

8.5.4 DLS Action frame details

8.5.4.1 General

Several Action frame formats are defined for DLS management purposes. A DLS Action field, in the octet field immediately after the Category field, differentiates the formats. The DLS Action field values associated with each frame format are defined in Table 8-198.

Table 8-198—DLS Action field values

DLS Action field value	Meaning
0	DLS Request
1	DLS Response
2	DLS Teardown
3–255	Reserved

8.5.4.2 DLS Request frame format

The DLS Request frame is used to set up a direct link with a peer MAC. The Action field of the DLS Request frame contains the information shown in Table 8-199, with some fields being optionally present as indicated in the “Notes” column of the table.

Table 8-199—DLS Request frame Action field format

Order	Information	Notes
1	Category	
2	DLS Action	
3	Destination MACAddress	
4	Source MACAddress	
5	Capability Information	
6	DLS Timeout Value	
7	Supported rates	
8	Extended Supported Rates	
9	HT Capabilities	The HT Capabilities element is present when the dot11HighThroughputOptionImplemented attribute is true.

The Category field is set to 2 (representing DLS).

The DLS Action field is set to 0 (representing DLS request).

The Destination MAC Address field value is the MAC address of the target destination.

The Source MAC Address field value is the MAC address of the initiating STA.

The Capability Information field value is the capability information of the originator of the request.

The DLS Timeout Value field is defined in 8.4.1.13.

The Supported Rates and Extended Supported Rates fields contain the supported rates information of the originator.

8.5.4.3 DLS Response frame format

The DLS Response frame is sent in response to a DLS Request frame. The Action field of a DLS Response frame contains the information shown in Table 8-200, with some fields being optionally present as indicated in the "Notes" column of the table.

Table 8-200—DLS Response frame Action field format

Order	Information	Notes
1	Category	
2	DLS Action	
3	Status Code	
4	Destination MACAddress	
5	Source MACAddress	
6	Capability Information	
7	Supported rates	
8	Extended Supported rates	
9	HT Capabilities	The HT Capabilities element is present when the dot11HighThroughputOptionImplemented attribute is true.

The Category field is set to 2 (representing DLS).

The DLS Action field is set to 1 (representing DLS response).

The Status Code field is defined in 8.4.1.9.

The Destination MAC Address field value and the Source MAC Address field value are copied from the corresponding fields in the DLS Request frame.

The Capability Information field is the capability information of the target destination. This information is included only if the DLS result code corresponds to SUCCESS (DLS status code 0).

The Supported Rates and Extended Supported Rates fields contain the supported rates information of the target destination. This information is included only if the DLS result code corresponds to SUCCESS (DLS status code 0).

8.5.4.4 DLS Teardown frame format

The DLS Teardown frame is sent to terminate a direct link with a peer MAC. The Action field of the DLS Teardown frame contains the information shown in Table 8-201.

Table 8-201—DLS Teardown frame Action field format

Order	Information
1	Category
2	DLS Action
3	Destination MAC Address
4	Source MAC Address
5	Reason Code

The Category field is set to 2 (representing DLS).

The DLS Action field is set to 2 (representing DLS teardown).

The Destination MAC Address field value is the MAC address of the target destination.

The Source MAC Address field value is the MAC address of the initiating STA.

The Reason Code field is defined in 8.4.1.7.

8.5.5 Block Ack Action frame details

8.5.5.1 General

The ADDBA frames are used to set up or, if PBAC is used, to modify Block Ack for a specific TC or TS. A Block Ack Action field, in the octet immediately after the Category field, differentiates the Block Ack Action frame formats. The Block Ack Action field values associated with each frame format within the Block Ack category are defined in Table 8-202.

Table 8-202—Block Ack Action field values

Block Ack Action field values	Meaning
0	ADDBA Request
1	ADDBA Response
2	DELBA
3–255	Reserved

8.5.5.2 ADDBA Request frame format

An ADDBA Request frame is sent by an originator of Block Ack to another STA. The Action field of an ADDBA Request frame contains the information shown in Table 8-203.

Table 8-203—ADDBA Request frame Action field format

Order	Information
1	Category
2	Block Ack Action
3	Dialog Token
4	Block Ack Parameter Set
5	Block Ack Timeout Value
6	Block Ack Starting Sequence Control

The Category field is set to 3 (representing Block Ack).

The Block Ack Action field is set to 0 (representing ADDBA request).

The Dialog Token field is set to a nonzero value chosen by the STA.

The Block Ack Parameter Set field is defined in 8.4.1.14.

The Block Ack Timeout Value field is defined in 8.4.1.15.

The Block Ack Starting Sequence Control field is defined in 8.3.1.8.

8.5.5.3 ADDBA Response frame format

The ADDBA Response frame is sent in response to an ADDBA Request frame. The Action field of an ADDBA Response frame contains the information shown in Table 8-204.

Table 8-204—ADDBA Response frame Action field format

Order	Information
1	Category
2	Block Ack Action
3	Dialog Token
4	Status Code
5	Block Ack Parameter Set
6	Block Ack Timeout Value

The Category field is set to 3 (representing Block Ack).

The Block Ack Action field is set to 1 (representing ADDBA response).

The Dialog Token field value is copied from the corresponding received ADDBA Request frame.

The Status Code field is defined in 8.4.1.9.

The Block Ack Parameter Set field is defined in 8.4.1.14.

The Block Ack Timeout Value field is defined in 8.4.1.15.

8.5.5.4 DELBA frame format

The DELBA frame is sent by either the originator of the traffic or the recipient to terminate the Block Ack participation. The Action field of a DELBA frame format contains the information shown in Table 8-205.

Table 8-205—DELBA frame Action field format

Order	Information
1	Category
2	Block Ack Action
3	DELBA Parameter Set
4	Reason Code

The Category field is set to 3 (representing DELBA).

The Block Ack Action field is set to 2 (representing DELBA).

The DELBA Parameters field is defined in 8.4.1.16.

The Reason Code field is defined in 8.4.1.7.

8.5.6 Vendor-specific action details

The Vendor Specific Action frame is defined for vendor-specific signaling. The format of the Action field of the Vendor Specific Action frame is shown in Figure 8-437. An Organization Identifier, in the octet field immediately after the Category field, differentiates the vendors (see 8.4.1.31).

NOTE—If management frame protection is negotiated, then Vendor Specific Protected Action frames (see Table 8-38) are protected; otherwise they are unprotected.

	Category	Organization Identifier	Vendor Specific Content
Octets:	1	*j*	Variable

Figure 8-437—Vendor Specific Action frame Action field format

The Category field is set to the value indicating the vendor-specific category, as specified in Table 8-38.

The Organization Identifier contains a public organizationally unique identifier assigned by the IEEE and is specified in 8.4.1.31. The order of the Organization Identifier field is described in 8.2.2.

The Vendor Specific Content contains vendor-specific field(s). The length of the Vendor Specific Content in a Vendor Specific Action frame is limited by the maximum allowed MMPDU size.

8.5.7 Radio Measurement action details

8.5.7.1 General

Several Action frame formats are defined for Radio Measurement purposes. A Radio Measurement Action field, in the octet field immediately after the Category field, differentiates the formats. The Radio Measurement Action field values associated with each frame format are defined in Table 8-206.

Table 8-206—Radio Measurement Action field values

Radio Measurement Action field value	Description
0	Radio Measurement Request
1	Radio Measurement Report
2	Link Measurement Request
3	Link Measurement Report
4	Neighbor Report Request
5	Neighbor Report Response
6–255	Reserved

8.5.7.2 Radio Measurement Request frame format

The Radio Measurement Request frame uses the Action frame body format. It is transmitted by a STA requesting another STA to make one or more measurements on one or more channels. The format of the Action field in the Radio Measurement Request frame is shown in Figure 8-438.

	Category	Radio Measurement Action	Dialog Token	Number of Repetitions	Measurement Request Elements
Octets:	1	1	1	2	variable

Figure 8-438—Radio Measurement Request frame Action field format

The Category field is set to the value indicating the Radio Measurement category, as specified in Table 8-38 in 8.4.1.11.

The Radio Measurement Action field is set to indicate a Measurement Request according to Table 8-206 in 8.5.7.

The Dialog Token field is set to a nonzero value chosen by the STA sending the radio measurement request to identify the request/report transaction.

The Number of Repetitions field contains the requested number of repetitions for all the Measurement Request elements in this frame. A value of 0 in the Number of Repetitions field indicates Measurement Request elements are executed once without repetition. A value of 65 535 in the Number of Repetitions field indicates Measurement Request elements are repeated until the measurement is cancelled or superseded.

The Measurement Request Elements field contains zero or more of the Measurement Request elements described in 8.4.2.23. The number and length of the Measurement Request elements in a Measurement Request frame is limited by the maximum allowed MMPDU size.

8.5.7.3 Radio Measurement Report frame format

The Radio Measurement Report frame uses the Action frame body format. It is transmitted by a STA in response to a Radio Measurement Request frame or by a STA providing a triggered autonomous measurement report. The format of the Action field in the Radio Measurement Report frame is shown in Figure 8-439.

	Category	Radio Measurement Action	Dialog Token	Measurement Report Elements
Octets:	1	1	1	variable

Figure 8-439—Radio Measurement Report frame Action field format

The Category field is set to indicate the Radio Measurement category according to Table 8-38 in 8.4.1.11.

The Radio Measurement Action field is set to indicate a radio measurement report according to Table 8-206 in 8.5.7.

The Dialog Token field is set to the value in the corresponding Radio Measurement Request frame. If the Radio Measurement Report frame is not being transmitted in response to a Radio Measurement Request frame then the Dialog token is set to 0.

The Measurement Report Elements field contains one or more Measurement Report elements described in 8.4.2.24. The number and length of the Measurement Report elements in a single Radio Measurement Report frame is limited by the maximum allowed MMPDU size. Subclause 10.11.6 describes the required behavior for multiframe Radio Measurement Report frame responses.

8.5.7.4 Link Measurement Request frame format

The Link Measurement Request frame uses the Action frame body format and is transmitted by a STA to request another STA to respond with a Link Measurement Report frame to enable measurement of link path loss and estimation of link margin. The format of the Action field in the Link Measurement Request frame is shown in Figure 8-440.

	Category	Radio Measurement Action	Dialog Token	Transmit Power Used	Max Transmit Power	Optional Subelements
Octets:	1	1	1	1	1	variable

Figure 8-440—Link Measurement Request frame Action field format

The Category field is set to the value indicating the Radio Measurement category, as specified in Table 8-38 in 8.4.1.11.

The Radio Measurement Action field is set to indicate a Link Measurement Request according to Table 8-206 in 8.5.7.

The Dialog Token field is set to a nonzero value chosen by the STA sending the request to identify the transaction.

The Transmit Power Used field is set to the transmit power used to transmit the frame containing the Link Measurement Request, as described in 8.4.1.20.

The Max Transmit Power field provides the upper limit on the transmit power as measured at the output of the antenna connector to be used by the transmitting STA on its operating channel. This field is described in 8.4.1.19. The Max Transmit Power field is a twos complement signed integer and is 1 octet in length, providing an upper limit, in a dBm scale, on the transmit power as measured at the output of the antenna connector to be used by the transmitting STA on its operating channel. The maximum tolerance for the value reported in Max Transmit Power field is ±5 dB. The value of the Max Transmit Power field is equal to the minimum of the maximum powers at which the STA is permitted to transmit in the operating channel by device capability, policy, and regulatory authority.

The Optional Subelements field format contains zero or more subelements, each consisting of a 1-octet Subelement ID field, a 1-octet Length field, and a variable-length Data field, as shown in Figure 8-402. Any optional subelements are ordered by nondecreasing subelement ID.

The Subelement ID field values for the defined optional subelements are shown in Table 8-207. A Yes in the Extensible column of a subelement listed in Table 8-207 indicates that the Length of the subelement might be extended in future revisions or amendments of this standard. When the Extensible column of an element is equal to Subelements, then the subelement might be extended in future revisions or amendments of this standard by defining additional subelements within the subelement. See 9.24.9.

Table 8-207—Optional subelement IDs for Link Measurement Request frame

Subelement ID	Name	Length field (octets)	Extensible
0–220	Reserved		
221	Vendor Specific	1 to 255	
222–255	Reserved		

The Vendor Specific subelements have the same format as their corresponding elements (see 8.4.2.28). Multiple Vendor Specific subelements are optionally present in the list of optional subelements.

8.5.7.5 Link Measurement Report frame format

The Link Measurement Report frame uses the Action frame body format and is transmitted by a STA in response to a Link Measurement Request frame. The format of the Action field in the Link Measurement Report frame is shown in Figure 8-441.

	Category	Radio Measurement Action	Dialog Token	TPC Report element	Receive Antenna ID	Transmit Antenna ID	RCPI	RSNI	Optional Subelements
Octets:	1	1	1	4	1	1	1	1	variable

Figure 8-441—Link Measurement Report frame Action field format

The Category field is set to indicate the Radio Measurement category according to Table 8-38 in 8.4.1.11.

The Radio Measurement Action field is set to indicate a Link Measurement Report according to Table 8-206 in 8.5.7.

The Dialog Token field is set to the Dialog Token value in the corresponding Link Measurement Request frame.

The TPC Report element is set as described in 8.4.2.19.

The Receive Antenna ID field contains the identifying number for the antenna(s) used to receive the corresponding Link Measurement Request frame. Antenna ID is defined in 8.4.2.42.

The Transmit Antenna ID field contains the identifying number for the antenna(s) used to transmit this Link Measurement Report frame. Antenna ID is defined in 8.4.2.42.

RCPI indicates the received channel power of the corresponding Link Measurement Request frame, which is a logarithmic function of the received signal power, as defined in the RCPI measurement subclause for the indicated PHY Type, as described in 8.4.2.40.

RSNI indicates the received signal to noise indication for the corresponding Link Measurement Request frame, as described in 8.4.2.43.

The Optional Subelements field format contains zero or more subelements, each consisting of a 1-octet Subelement ID field, a 1-octet Length field, and a variable-length Data field, as shown in Figure 8-402. Any optional subelements are ordered by nondecreasing subelement ID.

The Subelement ID field values for the defined optional subelements are shown in Table 8-208. A Yes in the Extensible column of a subelement listed in Table 8-208 indicates that the Length of the subelement might be extended in future revisions or amendments of this standard. When the Extensible column of an element is equal to Subelements, then the subelement might be extended in future revisions or amendments of this standard by defining additional subelements within the subelement. See 9.24.9.

Table 8-208—Optional subelement IDs for Link Measurement Report frame

Subelement ID	Name	Length field (octets)	Extensible
0–220	Reserved		
221	Vendor Specific	1 to 255	
222–255	Reserved		

The Vendor Specific subelements have the same format as their corresponding elements (see 8.4.2.28). Multiple Vendor Specific subelements are optionally present in the list of optional subelements.

8.5.7.6 Neighbor Report Request frame format

The Neighbor Report Request frame uses the Action frame body format and is transmitted by a STA requesting information in the neighbor report about neighboring APs. The format of the Action field in the Neighbor Report Request frame is shown in Figure 8-442.

	Category	Radio Measurement Action	Dialog Token	Optional Subelements
Octets:	1	1	1	variable

Figure 8-442—Neighbor Report Request frame Action field format

The Category field is set to the value indicating the Radio Measurement category, as specified in Table 8-38 in 8.4.1.11.

The Radio Measurement Action field is set to the value indicating Neighbor Report Request, as specified in Table 8-206 in 8.5.7.

The Dialog Token field is set to a nonzero value chosen by the STA sending the measurement request to identify the request/report transaction.

The Optional Subelements field format contains zero or more subelements each consisting of a 1-octet Subelement ID field, a 1-octet Length field, and a variable-length Data field, as shown in Figure 8-402. Any optional subelements are ordered by nondecreasing subelement ID.

The Subelement ID field values for the defined optional subelements are shown in Table 8-209. A Yes in the Extensible column of a subelement listed in Table 8-209 indicates that the Length of the subelement might be extended in future revisions or amendments of this standard. When the Extensible column of an element is equal to Subelements, then the subelement might be extended in future revisions or amendments of this standard by defining additional subelements within the subelement. See 9.24.9.

Table 8-209—Optional subelement IDs for Neighbor Report Request frame

Subelement ID	Name	Length field (octets)	Extensible
0	SSID	0 to 32	
1–220	Reserved		
221	Vendor Specific	1 to 255	
222–255	Reserved		

The SSID and Vendor Specific subelements have the same format as their corresponding elements (see 8.4.2.2 and 8.4.2.28, respectively). Multiple Vendor Specific subelements are optionally present in the list of optional subelements.

The presence of an optional SSID subelement in a Neighbor Report Request frame indicates a request for a neighbor list for the specified SSID in the SSID Element. The absence of an SSID element indicates neighbor report for the current ESS.

8.5.7.7 Neighbor Report Response frame format

The Neighbor Report Response frame uses the Action frame body format and is transmitted by a STA in response to a Neighbor Report Request frame. The format of the Action field in the Neighbor Report Response frame is shown in Figure 8-443.

	Category	Radio Measurement Action	Dialog Token	Neighbor Report Elements
Octets:	1	1	1	variable

Figure 8-443—Neighbor Report Response frame Action field format

The Category field is set to the value indicating the Radio Measurement category, as specified in Table 8-38 in 8.4.1.11.

The Radio Measurement Action field is set to the value indicating Neighbor Report Response, as specified in Table 8-206 in 8.5.7.

The Dialog Token field set to the value in the corresponding Neighbor Report Request frame. If the Neighbor Report Response frame is not being transmitted in response to a Neighbor Report Request frame, then the Dialog token is set to 0.

The Neighbor Report Elements field contains the Neighbor Report elements for validated APs described in 8.4.2.39. If the STA has no information in response to the Neighbor Report Request, the Neighbor Report elements are omitted. The number and length of the Neighbor Report Elements in a Neighbor Report frame is limited by the maximum allowed MMPDU size.

8.5.8 Public Action details

8.5.8.1 Public Action frames

The Public Action frame is defined to allow the following:

- Inter-BSS and AP to unassociated-STA communications
- Intra-BSS communication
- GAS

A Public Action field, in the octet immediately after the Category field, differentiates the Public Action frame formats. The defined Public Action frames are listed in Table 8-210.

Table 8-210—Public Action field values

Public Action field value	Description
0	20/40 BSS Coexistence Management (see 8.5.8.2)
1	DSE enablement
2	DSE deenablement

Table 8-210—Public Action field values *(continued)*

Public Action field value	Description
3	DSE Registered Location Announcement
4	Extended Channel Switch Announcement
5	DSE measurement request
6	DSE measurement report
7	Measurement Pilot
8	DSE power constraint
9	Vendor Specific
10	GAS Initial Request (see 8.5.8.12)
11	GAS Initial Response (see 8.5.8.13)
12	GAS Comeback Request (see 8.5.8.14)
13	GAS Comeback Response (see 8.5.8.15)
14	TDLS Discovery Response
15	Location Track Notification
16–255	Reserved

8.5.8.2 20/40 BSS Coexistence Management frame format

The 20/40 BSS Coexistence Management frame is a Public Action frame. The format of its Action field is defined in Table 8-211.

Table 8-211—20/40 BSS Coexistence Management frame Action field format

Order	Information	Notes
1	Category	
2	Public Action	
3	20/40 BSS Coexistence (see 8.4.2.62)	
4	20/40 BSS Intolerant Channel Report (see 8.4.2.60)	Appears zero or more times

The Category field is set to the value for Public, specified in Table 8-38.

The Public Action field is set to the value for 20/40 BSS Coexistence Management, specified in Table 8-210.

8.5.8.3 Measurement Pilot frame format

The Measurement Pilot frame uses the Action frame format. The format of the Action field is shown in Figure 8-444.

	Category	Public Action	Condensed Capability Information	Condensed Country String	Operating Class	Channel	Measurement Pilot Interval	Optional Subelements
Octets:	1	1	1	2	1	1	1	variable

Figure 8-444—Measurement Pilot frame Action field format

The Category field is set to the value indicating the Public category, as specified in Table 8-38 in 8.4.1.11.

The Public Action field is set to the value indicating Measurement Pilot, as specified in Table 8-210 in 8.5.8.1.

The Condensed Capability Information field contains two subfields as shown in Figure 8-445.

	B0	B1	B2 B7
	Spectrum Management	Short Slot Time	Reserved
Bits:	1	1	6

Figure 8-445—Condensed Capability Information field

The Spectrum Management subfield is set to 1 if dot11SpectrumManagementRequired is true; otherwise, it is set to 0.

The Short Slot Time subfield is set to 1 if dot11ShortSlotTimeOptionImplemented and dot11ShortSlotTimeOptionActivated are true. Otherwise, the Short Slot Time subfield is set to 0.

The Condensed Country String field is set to the first two octets of the value contained in dot11CountryString.

Operating Class indicates the operating class value for the operating channel. Country, Operating Class, and Channel Number together specify the channel frequency and spacing for the operating channel. Valid values of Operating Class are shown in Annex E.

Channel Number indicates the operating channel. Channel Number is defined within an Operating Class as shown in Annex E.

The Measurement Pilot Interval field is set to the value contained in dot11RMMeasurementPilotPeriod.

The Optional Subelements field format contains zero or more subelements each consisting of a 1-octet Subelement ID field, a 1-octet Length field, and a variable-length Data field, as shown in Figure 8-402. Any optional subelements are ordered by nondecreasing subelement ID.

The Subelement ID field values for the defined optional subelements are shown in Table 8-212. A Yes in the Extensible column of a subelement listed in Table 8-212 indicates that the Length of the subelement might be extended in future revisions or amendments of this standard. When the Extensible column of an element is equal to Subelements, then the subelement might be extended in future revisions or amendments of this standard by defining additional subelements within the subelement. See 9.24.9.

Table 8-212—Optional subelement IDs for Measurement Pilot frame

Subelement ID	Name	Length field (octets)	Extensible
0–70	Reserved		
71	Multiple BSSID	1 to 255	Subelements
72–220	Reserved		
221	Vendor Specific	1 to 255	
222–255	Reserved		

The Multiple BSSID and Vendor Specific subelements have the same format as the Multiple BSSID and Vendor Specific elements (see 8.4.2.48 and 8.4.2.38, respectively). Multiple Vendor Specific subelements may be included in the list of optional subelements.

8.5.8.4 DSE Enablement frame format

The DSE Enablement frame is an Action frame. It is transmitted by a STA as part of enablement. The format of the DSE Enablement frame Action field is shown in Figure 8-446.

	Category	Public Action	Requester STA Address	Responder STA Address	Reason Result Code	Enablement Identifier
Octets:	1	1	6	6	1	2

Figure 8-446—DSE Enablement frame Action field format

The Category field is set to the value for public action defined in Table 8-38.

The Public Action field is set to indicate a DSE Enablement frame, as defined in Table 8-210.

The RequesterSTAAddress field is the MAC address of the requesting STA that initiates the enablement process. The length of the RequesterSTAAddress field is 6 octets.

The ResponderSTAAddress field is the MAC address of the responding STA that grants enablement. The length of the ResponderSTAAddress field is 6 octets.

The Reason Result Code field is used to indicate the reason that a DSE Enablement frame was generated. The length of the Reason Result Code field is 1 octet. The reason result codes that have been allocated are shown in Table 8-213.

Table 8-213—Reason Result Code field values

Reason Result Code field value	Name	Description
0		Reserved
1		Reserved
2		Enablement requested

Table 8-213—Reason Result Code field values *(continued)*

Reason Result Code field value	Name	Description
3	SUCCESS	Success
4	REFUSED	Request declined
5	INVALID_PARAMETERS	Request not successful as one or more parameters have invalid values
6	TOO_MANY_ SIMULTANEOUS_REQUESTS	Enablement denied because the enabling STA is unable to handle additional dependent STAs
7–255		Reserved

The Enablement Identifier field is a 16-bit number assigned by an enabling STA to a dependent STA; otherwise, it is 0, set using the procedures defined in 10.12.

8.5.8.5 DSE Deenablement frame format

The DSE Deenablement frame is an Action frame. It is transmitted by a STA as part of deenablement. The format of the DSE Deenablement frame Action field is shown in Figure 8-447.

	Category	Public Action	Requester STA Address	Responder STA Address	Reason Result Code
Octets:	1	1	6	6	1

Figure 8-447—DSE Deenablement frame Action field format

The Category field is set to the value for public action defined in Table 8-38.

The Public Action field is set to indicate a DSE Deenablement frame, as defined in Table 8-210.

The RequesterSTAAddress field is the MAC address of the requesting STA that initiates the deenablement process. The length of the RequesterSTAAddress field is 6 octets.

The ResponderSTAAddress field is the MAC address of the responding STA that becomes deenabled. The length of the ResponderSTAAddress field is 6 octets.

The Reason Result Code field is used to indicate the reason that a DSE Deenablement frame was generated. The length of the Reason Result Code field is 1 octet. The reason result codes that have been allocated are shown in Table 8-214.

Table 8-214—Reason Result Code field values

Reason Result Code field value	Description
0	Reserved
1	Reserved
2	Deenablement requested
3–255	Reserved

8.5.8.6 DSE Registered Location Announcement frame format

The DSE Registered Location Announcement frame is transmitted by a dependent STA to advertise the registered location of its enabling STA. The format of the DSE Registered Location Announcement frame Action field is shown in Figure 8-448.

	Category	Public Action	DSE Registered Location element body fields
Octets:	1	1	20

Figure 8-448—DSE Registered Location Announcement frame Action field format

The Category field is set to the value for public action defined in Table 8-38.

The Public Action field is set to indicate a DSE Registered Location Announcement frame, as defined in Table 8-210.

The remaining fields are as defined in the DSE Registered Location element body (see 8.4.2.54).

8.5.8.7 Extended Channel Switch Announcement frame format

The Extended Channel Switch Announcement frame is transmitted by an AP in an infrastructure BSS, a STA in an IBSS, or a mesh STA in an MBSS to advertise a channel switch. The format of the Extended Channel Switch Announcement frame Action field is shown in Figure 8-449.

	Category	Public Action	Channel Switch Mode	New Operating Class	New Channel Number	Channel Switch Count	Mesh Channel Switch Parameters element
Octets:	1	1	1	1	1	1	6

Figure 8-449—Extended Channel Switch Announcement frame Action field format

The Category field is set to the value for public action defined in Table 8-38.

The Public Action field is set to indicate an Extended Channel Switch Announcement frame, as defined in Table 8-210.

The Channel Switch Mode, New Operating Class, New Channel Number, and Channel Switch Count fields are as described in the Extended Channel Switch Announcement element (see 8.4.2.55).

Mesh Channel Switch Parameters element is defined in 8.4.2.105. This element is present when a mesh STA performs MBSS channel switch. The Mesh Channel Switch Parameters element is not included for channel switch other than the MBSS channel switch.

8.5.8.8 DSE Measurement Request frame format

The DSE Measurement Request frame is a Public Action frame requesting a DSE measurement report. It is transmitted by a STA using the procedures defined in 10.12. The format of the DSE Measurement Request frame Action field is shown in Figure 8-450.

	Category	Public Action	Requester STA Address	Responder STA Address	Operating Class	Channel Number	Measurement Start Time	Measurement Duration
Octets:	1	1	6	6	1	1	8	2

Figure 8-450—DSE Measurement Request frame Action field format

The Category field is set to the value for public action defined in Table 8-38.

The Public Action field is set to indicate a DSE Measurement Request frame, as defined in Table 8-210.

The RequesterSTAAddress field is the MAC address of the requesting STA that grants enablement. The length of the RequesterSTAAddress field is 6 octets.

The ResponderSTAAddress field is the MAC address of the responding STA that operates based on the enablement. The length of the ResponderSTAAddress field is 6 octets.

The Operating Class field indicates the channel set for which the measurement request applies. The Operating Class and Channel Number fields together specify the channel frequency and channel bandwidth for which the measurement request applies. Valid values for the Operating Class field are shown in Annex E.

The Measurement Start Time field is set to the timing synchronization function (TSF) at the time (± 1 TU) at which the requested DSE request measurement starts. A value of 0 indicates it starts immediately.

The Measurement Duration field is set to the duration of the requested measurement, expressed in number of time units (TUs).

8.5.8.9 DSE Measurement Report frame format

The DSE Measurement Report frame is a Public Action frame. It is transmitted by a STA using the procedures defined in 10.12. The format of the DSE Measurement Report frame Action field is shown in Figure 8-451.

	Category	Public Action	Requester STA Address	Responder STA Address	Length	Operating Class	Channel Number	Measure-ment Report Mode	Actual Measure-ment Start Time	Measure-ment Duration	Reported DSE LCI fields
Octets:	1	1	6	6	2	1	1	1	8	2	n x 26

Figure 8-451—DSE Measurement Report frame Action field format

The Category field is set to the value for public action defined in Table 8-38.

The Public Action field is set to indicate a DSE Measurement Report frame, as defined in Table 8-210.

The RequesterSTAAddress field is the MAC address of the requesting STA that grants enablement. The length of the RequesterSTAAddress field is 6 octets.

The ResponderSTAAddress field is the MAC address of the responding STA that operates based on the enablement. The length of the ResponderSTAAddress field is 6 octets.

The Length field indicates the length of the remaining frame fields in octets, and the value is variable. The minimum value of the Length field is 13.

The Operating Class field indicates the channel set for which the measurement report applies. The Operating Class and Channel Number fields together specify the channel frequency and spacing for which the measurement request applies. Valid values for the Operating Class field are shown in Annex E.

The Measurement Report Mode field is as defined in 8.4.2.24 (see Figure 8-141).

The Actual Measurement Start Time field is set to the measuring STA’s TSF timer at the time (± 1 TU) at which the DSE measurement started.

The Measurement Duration field is set to the duration over which the requested measurement was measured, expressed in number of TUs.

The reported DSE LCI fields contain the DSE LCI received at the measuring STA. If the reported DSE LCI fields would cause the frame to exceed the maximum MAC management protocol data unit (MMPDU) size, then the reported DSE LCI fields are truncated so that the last reported DSE LCI field is complete. The DSE LCI field format is shown in Figure 8-452.

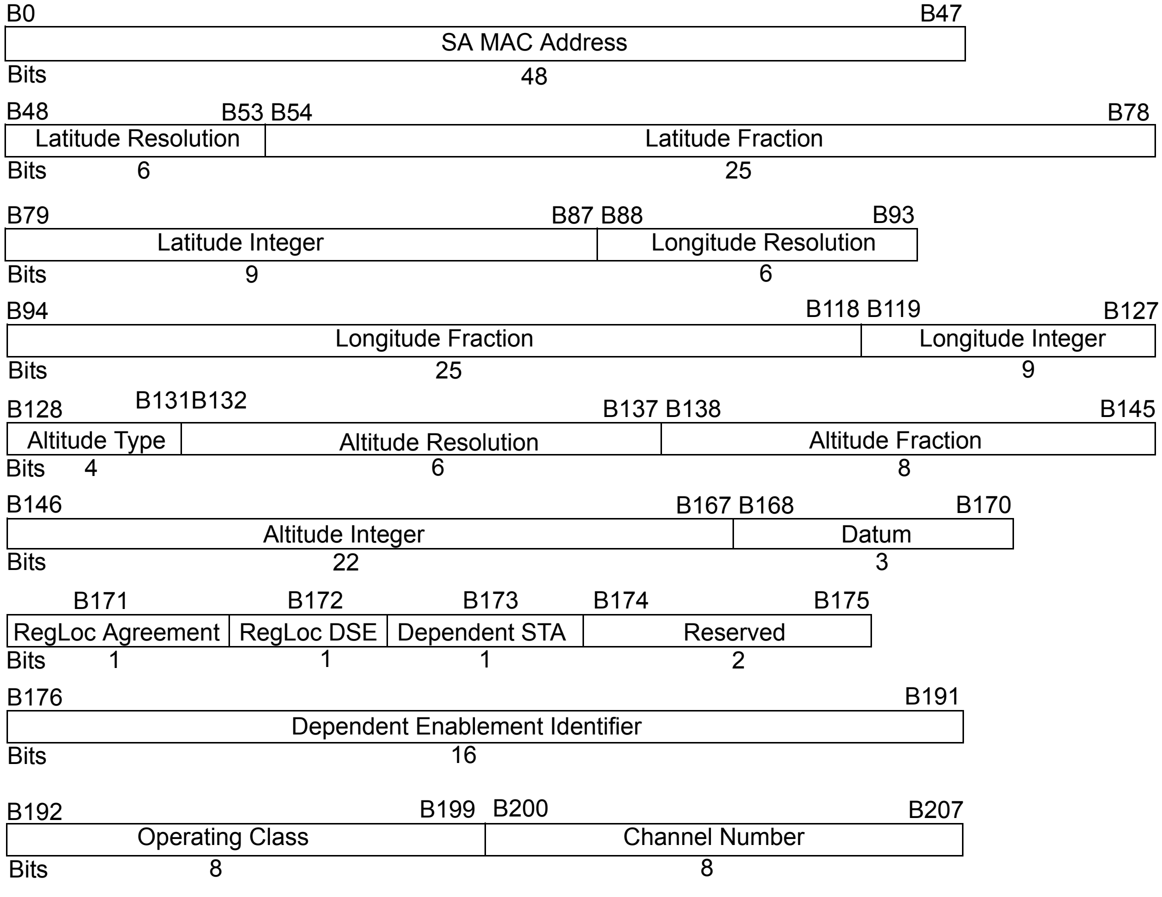


Figure 8-452—DSE LCI field format

The SA MAC Address field contains the SA MAC address from the Beacon or Public Action frame containing a DSE Registered Location element being reported.

The remaining fields are as defined in the DSE Registered Location element body (see 8.4.2.54).

8.5.8.10 DSE Power Constraint frame format

The DSE Power Constraint frame is a Public Action frame requesting that a dependent STA constrain transmit power below the regulatory limit. It is transmitted by an enabling STA as part of enablement. The format of the DSE Power Constraint frame Action field is shown in Figure 8-453.

	Category	Public Action	Requester STA Address	Responder STA Address	Reason Result Code	Local Power Constraint
Octets:	1	1	6	6	1	1

Figure 8-453—DSE Power Constraint frame Action field format

The Category field is set to the value for public action defined in Table 8-38.

The Public Action field is set to indicate a DSE Power Constraint frame, as defined in Table 8-210.

The RequesterSTAAddress field is the MAC address of the requesting STA that grants the enablement. The length of the RequesterSTAAddress field is 6 octets.

The ResponderSTAAddress field is the MAC address of the responding STA that initiates the enablement. The length of the ResponderSTAAddress field is 6 octets.

The Reason Result Code field is used to indicate the reason that a DSE Power Constraint frame was generated. The length of the Reason Result Code field is 1 octet. The reason result codes that have been allocated are shown in Table 8-215.

Table 8-215—Reason Result Code field values

Reason Result Code field value	Name	Description
0		Reserved
1		Reserved
2		Power constraint requested
3	SUCCESS	Success
4	REFUSED	Refused - no reason specified
5	INVALID_PARAMETERS	Request not successful as one or more parameters have invalid values
6-255		Reserved

The Local Power Constraint field is coded as an unsigned integer in units of decibels relative to 1 mW. The local maximum transmit power for a channel is thus defined as the maximum transmit power level specified for the channel in the Country element minus the local power constraint specified for the channel in the DSE Power Constraint frame.

8.5.8.11 Vendor Specific Public Action frame format

The Vendor Specific Public Action frame is defined for vendor-specific signaling between unassociated STAs. The format of the Action field of the Vendor Specific Public Action frame is shown in Figure 8-454.

	Category	Public Action	Organization Itentifier	Vendor Specific Content
Octets:	1	1	variable	variable

Figure 8-454—Vendor Specific Public Action frame Action field format

The Category field is set to the value indicating the Public category, as specified in Table 8-38 in 8.4.1.11.

The Public Action field is set to the value indicating Vendor Specific Public, as specified in Table 8-210 in 8.5.8.1.

The Organization Identifier field contains a public organizationally unique identifier assigned by the IEEE and is specified in 8.4.1.31. The order of the Organization Identifier field is described in 8.2.2. The length of the Organization Identifier field is variable, as specified in 8.4.1.31.

The Vendor Specific Content contains the vendor-specific fields and may include elements defined in the standard. The length of the Vendor Specific Content in a Vendor Specific Public Action frame is limited by the maximum allowed MMPDU size.

8.5.8.12 GAS Initial Request frame format

The GAS Initial Request frame is a Public Action frame. It is transmitted by a requesting STA to request information from another STA. The format of the GAS Initial Request frame body is shown in Table 8-216.

Table 8-216—GAS Initial Request frame body format

Order	Information
0	Category
1	Action
2	Dialog Token
3	Advertisement Protocol element
4	Query Request length
5	Query Request

The Category field is set to the value indicating a Public Action frame, as specified in Table 8-38.

The Action field is set to the value specified in Table 8-210 for a GAS Initial Request frame.

The Dialog Token field is defined in 8.4.1.12 and set by the requesting STA.

The Advertisement Protocol element is defined in 8.4.2.95. The Advertisement Protocol element includes exactly one Advertisement Protocol ID.

The Query Request length field is defined in Figure 8-455. The value of the Query Request length field is set to the total number of octets in the Query Request field.

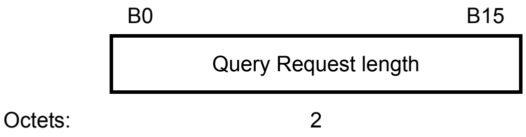


Figure 8-455—Query Request length field

The Query Request field is defined in Figure 8-456. The Query Request field is a generic container whose value is a GAS Query that is formatted in accordance with the protocol specified in the Advertisement Protocol element.

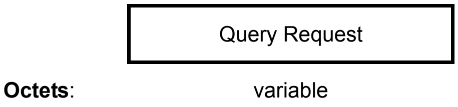


Figure 8-456—Query Request field

8.5.8.13 GAS Initial Response frame format

The GAS Initial Response frame is a Public Action frame. It is transmitted by a STA responding to a GAS Initial Request frame. The format of the GAS Initial Response frame body is shown in Table 8-217.

Table 8-217—GAS Initial Response frame body format

Order	Information
0	Category
1	Action
2	Dialog Token
3	Status Code
4	GAS Comeback Delay
5	Advertisement Protocol element
6	Query Response Length
7	Query Response (optional)

The Category field is set to the value indicating a Public Action frame, as specified in Table 8-38.

The Action field is set to the value specified in Table 8-210 for a GAS Initial Response frame.

The Dialog Token field is copied from the corresponding GAS Initial Request frame.

The Status Code values are defined in Table 8-37.

The GAS Comeback Delay field specifies the delay time value in TUs. The GAS Comeback Delay field format is provided in Figure 8-457. The behavior is described in 10.24.3.1. The value 0 will be returned by the STA when a Query Response is provided in this frame.

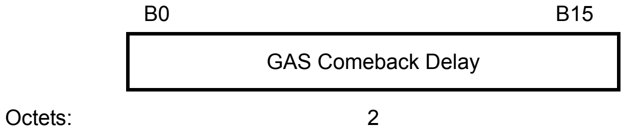


Figure 8-457—GAS Comeback Delay field

The Advertisement Protocol element is defined in 8.4.2.95. The Advertisement Protocol element includes exactly one Advertisement Protocol ID.

The Query Response Length field is defined in Figure 8-458. The value of the Query Response Length field is set to the total number of octets in the Query Response field. If the Query Response Length field is set to 0, then there is no Query Response included in this Action frame.

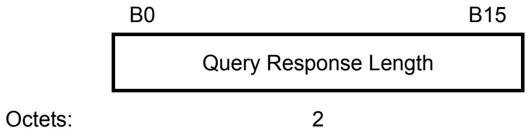


Figure 8-458—Query Response length field

The Query Response field is defined in Figure 8-459. The Query Response field is a generic container whose value is the response to a GAS Query and is formatted in accordance with the protocol specified in the Advertisement Protocol element.

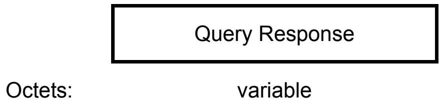


Figure 8-459—Query Response field

8.5.8.14 GAS Comeback Request frame format

The GAS Comeback Request frame is a Public Action frame. It is transmitted by a requesting STA to a responding STA. The format of the GAS Comeback Request frame body is shown in Table 8-218.

Table 8-218—GAS Comeback Request frame body format

Order	Information
0	Category
1	Action
2	Dialog Token

The Category field is set to the value indicating a Public Action frame, as specified in Table 8-38.

The Action field is set to the value specified in Table 8-210 for a GAS Comeback Request frame.

The Dialog Token field is copied from the corresponding GAS Initial Request frame.

8.5.8.15 GAS Comeback Response frame format

The GAS Comeback Response frame is a Public Action frame. It is transmitted by a responding STA to a requesting STA. The format of the GAS Comeback Response frame body is shown in Table 8-219.

Table 8-219—GAS Comeback Response frame body format

Order	Information
0	Category
1	Action
2	Dialog Token
3	Status Code
4	GAS Query Response Fragment ID
5	GAS Comeback Delay
6	Advertisement Protocol element
7	Query Response Length
8	Query Response (optional)

The Category field is set to the value indicating a Public Action frame, as specified in Table 8-38.

The Action field is set to the value specified in Table 8-210 for a GAS Comeback Response frame.

The Dialog Token field is copied from the Dialog Token field of the corresponding GAS Comeback Request frame. The same dialog token value will be present in all fragments of a multi-fragment query response.

The Status Code values are defined in Table 8-37. The same status code value will be present in all fragments of a multi-fragment query response.

The GAS Query Response Fragment ID is defined in 8.4.1.33. If the responding STA has not received a response to the query that it posted on behalf of a requesting STA, then the responding STA sets the GAS Query Response Fragment ID to 0. When there is more than one query response fragment, the responding STA sets the GAS Query Response Fragment ID to 0 for the initial fragment and increments it by 1 for each subsequent fragment in a multi-fragment Query Response. The More GAS Fragments field is set to 0 whenever the final fragment of a query response is being transmitted. A GAS Query Response Fragment ID field having a nonzero Fragment ID and the More GAS Fragments field set to 1 indicates to the requesting STA that another GAS Comeback frame exchange should be performed to continue the retrieval of the query response.

The GAS Comeback Delay field format is provided in Figure 8-457. A nonzero GAS Comeback Delay value is returned by the responding STA in this frame to indicate that the GAS Query being carried out on behalf of the requesting STA is still in progress.

- A nonzero value indicates to the requesting STA that another GAS Comeback frame exchange should be performed after expiry of the GAS Comeback Delay timer in order to retrieve the query response.
- This field is set to 0 for all GAS Comeback Response frames containing a query response or a fragment of a multi-fragment query response.

The Advertisement Protocol element is defined in 8.4.2.95. The Advertisement Protocol element includes exactly one Advertisement Protocol ID.

The Query Response Length field is defined in Figure 8-458. The value of the Query Response Length field is the total number of octets in the Query Response field. If the Query Response Length field is set to 0, then there is no Query Response included in this Action frame.

The Query Response field is defined in Figure 8-459. The value of the Query Response field is a generic container dependent on the advertisement protocol specified in the Advertisement Protocol element and the query itself. In a multi-fragment query response, the response to the query posted on behalf of a requesting STA is fragmented such that each fragment to be transmitted fits within the MMPDU size limitation.

8.5.8.16 TDLS Discovery Response frame format

The TDLS Discovery Response frame contains the information shown in Table 8-220.

Table 8-220—Information for TDLS Discovery Response frame

Order	Information	Notes
1	Category	The Category field is set to the value for Public, as defined in Table 8-38.
2	Action	The Action field is set to 14, representing TDLS Discovery Response.
3	Dialog Token	The Dialog Token is copied from the corresponding TDLS Discovery Request frame. The Dialog Token is specified in 8.4.1.12.
4	Capability	The Capability field indicates the capabilities of the STA. The Capability field is defined in 8.4.1.4.
5	Supported rates	The Supported Rates element indicates the rates which are supported by the STA. The Supported Rates element is defined in 8.4.2.3.
6	Extended supported rates	The Extended Supported Rates element is present whenever there are more than eight supported rates, and it is optional otherwise. The Extended Supported Rates element is defined in 8.4.2.15.
7	Supported Channels	The Supported Channels element is present if the TDLS channel switching capability field is equal to 1. The Supported Channels element is defined in 8.4.2.20.
8	RSNE	The RSNE is optionally present if security is required on the direct link. The RSNE is defined in 8.4.2.27.
9	Extended Capabilities	The Extended Capabilities element is optionally present if any of the fields in this element are nonzero. The Extended Capabilities element is defined in 8.4.2.29.
10	FTE	The FTE is optionally present if security is required on the direct link. The FTE is defined in 8.4.2.50.

Table 8-220—Information for TDLS Discovery Response frame *(continued)*

Order	Information	Notes
11	Timeout Interval (TPK Key Lifetime)	The Timeout Interval element contains the TPK Key Lifetime. It is present if security is required on the direct link. The Timeout Interval element is defined in 8.4.2.51.
12	Supported Operating Classes	The Supported Operating Classes element is present if the TDLS channel switching capability field is equal to 1. The Supported Operating Classes element is defined in 8.4.2.56.
13	HT Capabilities	The HT Capabilities element is present when dot11HighThroughputOptionImplemented is true. The HT Capabilities element is defined in 8.4.2.58.
14	20/40 BSS Coexistence	The 20/40 BSS Coexistence element is present when dot112040BSSCoexistenceManagementSupport is true. The 20/40 BSS Coexistence element is defined in 8.4.2.60.
15	Link Identifier	The Link Identifier element is defined in 8.4.2.64.

The TDLS Discovery Response frame is transmitted directly (i.e., not via the AP) to the TDLS peer STA that sent the corresponding TDLS Discovery Request frame. See 10.22.

8.5.8.17 Location Track Notification frame format

The Location Track Notification frame uses the Action frame body format and is transmitted by a STA to allow remote location determination to occur by another STA. The format of the Location Track Notification frame body is shown in Figure 8-460.

	Category	Action	Location Parameters Element	Measurement Report Element (optional)
Octets:	1	1	variable	variable

Figure 8-460—Location Track Notification frame format

The Category field is the value for Public Action defined Table 8-38.

The Action field is the value indicating Location Track Notification, as specified in Table 8-210.

The Location Parameters Element field contains the Location Parameters subelements, described in Table 8-153. Table 8-221 defines the allowed Location Parameters subelements for a Location Parameters element that is included in the frame.

Table 8-221—Location Parameters Element field for Location Track Notification frame

Allowed subelements	Subelement ID	Notes
Location Indication Channels	2	The Location Indication Channels subelement is included in the Location Track Notification frame.
Radio Information	4	The Radio Information subelement is included in the Location Track Notification frame.

Table 8-221—Location Parameters Element field for Location Track Notification frame *(continued)*

Allowed subelements	Subelement ID	Notes
Motion	5	The Motion subelement is included in the Location Track Notification frame if dot11MgmtOptionMotionDetectionActivated is true.
Time of Departure	7	The Time of Departure subelement is included in the Location Track Notification frame if dot11MgmtOptionTODActivated is true.
Location Indication Options	8	The Location Indication Options subelement is included in the Location Track Notification frame if the successful Location Configuration Request frame that configured the STA included a Location Indication Options subelement.
Vendor Specific	221	The Vendor Specific subelement may be included in the Location Track Notification frame.

The Measurement Report Element field contains a Measurement Report element of type Beacon Report as defined in 8.4.2.24.7. The Measurement Report Element field is included in the Location Track Notification frame if the Location Parameters Element field contains a Location Indication Options subelement. The Measurement Report element Measurement Token field is set to the same value of the Dialog Token in the Location Configuration Request frame that configured the STA.

8.5.9 FT Action frame details

8.5.9.1 General

Four Action frame formats are defined to support fast BSS transitions over the DS, which are initiated through the currently associated AP. The FT Action frames are sent over the air between the STA and the current AP. The Action frame is used as a transport mechanism for data that are destined for the target AP. An FT Action field, in the octet immediately after the Category field, differentiates the FT Action frame formats. The FT Action field values associated with each FT Action frame format are defined in Table 8-222.

Table 8-222—FT Action field values

FT Action field value	Description
0	Reserved
1	FT Request frames
2	FT Response frames
3	FT Confirm frames
4	FT Ack frames
5–255	Reserved

8.5.9.2 FT Request frame

The FT Request frame is sent by the STA to its associated AP to initiate an over-the-DS fast BSS transition.

Figure 8-461 shows the format of the FT Request frame Action field.

	Category	FT Action	STA Address	Target AP Address	FT Request frame body
Octets:	1	1	6	6	variable

Figure 8-461—FT Request frame Action field format

The Category field is set to the value given in 8.4.1.11 for FT Action frames.

The FT Action field is set to the value given in Table 8-222 for FT Request frames.

The STA Address field is set to the FTO’s MAC address.

The Target AP Address field is set to the BSSID value of the target AP.

The FT Request frame body contains the information shown in Table 8-223.

Table 8-223—FT Request frame body

Order	Information	Notes
1	RSN	A RSNE is present if dot11RSNAActivated is true.
2	Mobility Domain	The MDE is present.
3	Fast BSS Transition	An FTE is present if dot11RSNAActivated is true.

The usage of these elements is defined in 12.8.2.

8.5.9.3 FT Response frame

The FT Response frame is transmitted by the currently associated AP as a response to the STA’s FT Request frame. Figure 8-462 shows the format of the FT Response frame Action field.

	Category	FT Action	STA Address	Target AP Address	Status Code	FT Response frame body
Octets:	1	1	6	6	2	variable

Figure 8-462—FT Response frame Action field format

The Category field is set to the value given in 8.4.1.11 for FT Action frames.

The FT Action field is set to the value given in Table 8-222 for FT Response frames.

The STA Address field is set to the FTO’s MAC address.

The Target AP Address field is set to the BSSID value of the target AP.

The Status Code field is a value from the options listed in 8.4.1.9.

If the Status Code field is 0, then the FT Response frame body contains the information shown in Table 8-224.

Table 8-224—FT Response frame body

Order	Information	Notes
1	RSN	The RSNE is present if dot11RSNAActivated is true.
2	Mobility Domain	The MDE is present.
3	Fast BSS Transition	An FTE is present if dot11RSNAActivated is true.

The usage of these elements is defined in 12.8.3.

8.5.9.4 FT Confirm frame

The FT Confirm frame in an RSN is confirmation to the target AP of receipt of the ANonce and indicates the liveness of the PTKSA. The FT Confirm frame is optionally used by the FTO to request resources. Figure 8-463 shows the FT Confirm frame Action field format.

	Category	FT Action	STA Address	Target AP Address	FT Confirm frame body
Octets:	1	1	6	6	variable

Figure 8-463—FT Confirm frame Action field format

The Category field is set to the value given in 8.4.1.11 for FT Action frames.

The FT Action field is set to the value given in Table 8-222 for FT Confirm frames.

The STA Address field is set to the FTO's MAC address.

The Target AP Address field is set to the BSSID value of the target AP.

The FT Confirm frame body contains the information shown in Table 8-225.

Table 8-225—FT Confirm frame body

Order	Information	Notes
1	RSN	The RSNE is present if dot11RSNAActivated is true.
2	Mobility Domain	The MDE is present.
3	Fast BSS Transition	An FTE is present if dot11RSNAActivated is true.
4	RIC	The RIC Request field is present if resources are being requested.

The usage of these elements is defined in 12.8.4.

8.5.9.5 FT Ack frame

The FT Ack frame is transmitted by the currently associated AP as a response to the STA’s FT Confirm frame. Figure 8-464 shows the FT Ack frame Action field format.

	Category	FT Action	STA Address	Target AP Address	Status Code	FT Ack frame body
Octets	1	1	6	6	2	variable

Figure 8-464—FT Ack frame Action field format

The Category field is set to the value given in 8.4.1.11 for FT Action frames.

The FT Action field is set to the value given in Table 8-222 for FT Ack frames.

The STA Address field is set to the FTO’s MAC address.

The Target AP Address field is set to the BSSID value of the target AP.

The Status Code field is a value from the options listed in 8.4.1.9.

If the Status Code field is 0, then the FT Ack frame body contains the information shown in Table 8-226.

Table 8-226—FT Ack frame body

Order	Information	Notes
1	RSN	The RSNE is present if dot11RSNAActivated is true.
2	Mobility Domain	The MDE is present.
3	Fast BSS Transition	An FTE is present if dot11RSNAActivated is true.
4	Timeout Interval (reassociation deadline)	A TIE containing the reassociation deadline interval is present if resources were requested in the FT Confirm frame and dot11RSNAActivated is false.
5	RIC	The RIC Response field is present if resources were requested in the FT Confirm frame.

The usage of these elements is defined in 12.8.5.

8.5.10 SA Query Action frame details

8.5.10.1 General

Two Action frame formats are defined for the SA Query procedure. A SA Query Action field, in the octet field immediately after the Category field, differentiates the formats. The Action field values associated with each frame format are defined in Table 8-227.

NOTE—The SA query functionality defined in this standard is used to prevent the Association Lockout problem (defined in 10.3).

Table 8-227—SA Query Action field values

SA Query Action field value	Description
0	SA Query Request
1	SA Query Response

8.5.10.2 SA Query Request frame

The SA Query Request frame is used to request a SA Query Response from the receiving STA. The format of the Action field is shown in Figure 8-465.

	Category	SA Query Action	Transaction Identifier
Octets:	1	1	2

Figure 8-465—SA Query Request frame Action field format

The Category field is set to the value indicating the SA Query category, as specified in Table 8-38 in 8.4.1.11.

The SA Query Action field is set to the value indicating SA Query Request frame, as specified in Table 8-227 in 8.5.10.

The Transaction Identifier field is a 16-bit non-negative counter value set by the STA sending the SA Query Request frame to identify any outstanding request/response transaction.

8.5.10.3 SA Query Response frame

The SA Query Response frame is used to respond to an SA Query Request frame from another STA. The format of the Action field is shown in Figure 8-466.

	Category	SA Query Action	Transaction Identifier
Octets:	1	1	2

Figure 8-466—SA Query Response frame Action field format

The Category field is set to the value indicating the SA Query category, as specified in Table 8-38 in 8.4.1.11.

The SA Query Action field is set to the value indicating SA Query Response frame, as specified in Table 8-227 in 8.5.10.

The Transaction Identifier field is set to the same value as the Transaction Identifier field in the corresponding SA Query Request frame.

8.5.11 Protected Dual of Public Action frames

The Protected Dual of Public Action frame is defined to allow robust STA-STA communications of the same information that is conveyed in Action frames that are not robust (see 8.4.1.11). A Public Action field, in the octet immediately after the Category field, differentiates the Protected Dual of Public Action frame formats. The defined Protected Dual of Public Action frames are listed in Table 8-228.

The Protected Dual of Public Action frames have the same format as the corresponding nonprotected Public Action frame.

Table 8-228—Public Action field values defined for Protected Dual of Public Action frames

Public Action field value	Description	Defined in
0	Reserved	
1	Protected DSE Enablement	8.5.8.4
2	Protected DSE Deenablement	8.5.8.5
3	Reserved	
4	Protected Extended Channel Switch Announcement	8.5.8.7
5	Protected Measurement Request	8.5.8.8
6	Protected Measurement Report	8.5.8.9
7	Reserved	
8	Protected DSE Power Constraint	8.5.8.10
9	Protected Vendor Specific	8.5.8.11
10	Protected GAS Initial Request	8.5.8.12
11	Protected GAS Initial Response	8.5.8.13
12	Protected GAS Comeback Request	8.5.8.14
13	Protected GAS Comeback Response	8.5.8.15
14–255	Reserved	

8.5.12 HT Action frame details

8.5.12.1 HT Action field

Several Action frame formats are defined to support HT features. An HT Action field, in the octet immediately after the Category field, differentiates the HT Action frame formats. The HT Action field values associated with each frame format within the HT category are defined in Table 8-229. The frame formats are defined in 8.5.12.2 through 8.5.12.9.

Table 8-229—HT Action field values

HT Action field value	Meaning	Time priority
0	Notify Channel Width	No
1	SM Power Save	No

Table 8-229—HT Action field values *(continued)*

HT Action field value	Meaning	Time priority
2	PSMP	Yes
3	Set PCO Phase	Yes
4	CSI	Yes
5	Noncompressed Beamforming	Yes
6	Compressed Beamforming	Yes
7	ASEL Indices Feedback	Yes
8–255	Reserved	—

8.5.12.2 Notify Channel Width frame format

A STA sends the Notify Channel Width frame to another STA if it wants to change the channel width of frames that the other STA sends to it. See definition in 10.15.2.

The format of the Notify Channel Width frame Action field is defined in Table 8-230.

Table 8-230—Notify Channel Width frame Action field format

Order	Information
1	Category
2	HT Action
3	Channel Width (see 8.4.1.21)

This frame can be sent by both non-AP STA and AP. If an AP wishes to receive 20 MHz packets, it broadcasts this Action frame to all STAs in the BSS. In addition, the AP indicates its current STA channel width in the HT Operation element in the beacon.

The Category field is set to the value for HT, specified in 8-38.

The HT Action field is set to the value for Notify Channel Width, specified in Table 8-229.

8.5.12.3 SM Power Save frame format

The SM Power Save frame is of category HT. The SM Power Save frame is used to manage SM power-saving state transitions as defined in 10.2.4.

The Action field of the SM Power Save frame is defined in Table 8-231.

Table 8-231—SM Power Save frame Action field format

Order	Information
1	Category
2	HT Action
3	SM Power Control (see 8.4.1.22)

The Category field is set to the value for HT, specified in Table 8-38.

The HT Action field is set to the value for SM Power Save, specified in Table 8-229.

8.5.12.4 PSMP frame format

PSMP is an Action frame of category HT.

The DA field of this frame is a group address. (See 9.26.1.8.)

The PSMP Parameter Set field and PSMP STA Info fields define zero or more PSMP-DTT and PSMP-UTT time allocations that follow immediately after the PSMP frame.

The Action field of this frame is defined in Table 8-232. The PSMP Parameter Set field is followed by zero or more PSMP STA Info fields.

Table 8-232—PSMP frame Action field format

Order	Information
1	Category
2	HT Action
3	PSMP Parameter Set (see 8.4.1.24)
4 to (N_STA+3)	PSMP STA Info (see 8.4.1.25) Repeated N_STA times (N_STA is a subfield of the PSMP Parameter Set field)

The Category field is set to the value for HT, specified in Table 8-38.

The HT Action field is set to the value for PSMP, specified in Table 8-229.

The PSMP STA Info fields within a PSMP frame are ordered by STA_INFO Type as follows: group addressed (STA_INFO Type=1) and then individually addressed (STA_INFO Type=2).

8.5.12.5 Set PCO Phase frame format

Set PCO Phase is an Action frame of category HT that announces the phase change between 20 MHz and 40 MHz. The format of its Action field is defined in Table 8-233. The operation of the PCO feature is defined in 10.16.

Table 8-233—Set PCO Phase frame Action field format

Order	Information
1	Category
2	HT Action
3	PCO Phase Control (see 8.4.1.23)

The Category field is set to the value for HT, specified in Table 8-38.

The HT Action field is set to the value for Set PCO Phase, specified in Table 8-229.

This frame is sent by a PCO active AP.

8.5.12.6 CSI frame format

The CSI frame is an Action or an Action No Ack frame of category HT. The format of its Action field is defined in Table 8-234.

Table 8-234—CSI frame Action field format

Order	Information
1	Category
2	HT Action
3	MIMO Control (see 8.4.1.26)
4	CSI Report (see 8.4.1.27)

The Category field is set to the value for HT, specified in Table 8-38.

The HT Action field is set to the value for CSI, specified in Table 8-229.

In a CSI frame, the fields of the MIMO Control field (see 8.4.1.26) are used as described in Table 8-42. The Codebook Information subfield is reserved in this frame.

8.5.12.7 Noncompressed Beamforming frame format

The Noncompressed Beamforming frame is an Action or an Action No Ack frame of category HT. The format of its Action field is defined in Table 8-235.

Table 8-235—Noncompressed Beamforming frame Action field format

Order	Information
1	Category
2	HT Action
3	MIMO Control (see 8.4.1.26)
4	Noncompressed Beamforming Report (see 8.4.1.28)

The Category field is set to the value for HT, specified in Table 8-38.

The HT Action field is set to the value for Noncompressed Beamforming, specified in Table 8-229.

In a Noncompressed Beamforming frame, the fields of the MIMO Control field (see 8.4.1.26) are used as described in Table 8-42. The Codebook Information subfield is reserved in this frame.

8.5.12.8 Compressed Beamforming frame format

The Compressed Beamforming frame is an Action or an Action No Ack frame of category HT. The format of its Action field is defined in Table 8-236.

Table 8-236—Compressed Beamforming frame Action field format

Order	Information
1	Category
2	HT Action
3	MIMO Control (see 8.4.1.26)
4	Compressed Beamforming Report (see 8.4.1.29)

The Category field is set to the value for HT, specified in Table 8-38.

The HT Action field is set to the value for Compressed Beamforming, specified in Table 8-229.

In a Compressed Beamforming frame, the fields of the MIMO Control field (see 8.4.1.26) are used as described in Table 8-42. The Coefficient Size subfield is reserved in this frame.

8.5.12.9 Antenna Selection Indices Feedback frame format

The Antenna Selection Indices Feedback frame is an Action or Action No Ack frame of category HT. The format of its Action field is defined in Table 8-237.

Table 8-237—Antenna Selection Indices Feedback frame Action field format

Order	Information
1	Category
2	HT Action
3	Antenna Selection Indices (see 8.4.1.30)

The Category field is set to the value for HT, specified in Table 8-38.

The HT Action field is set to the value for ASEL Indices Feedback, specified in Table 8-229.

8.5.13 TDLS Action field formats

8.5.13.1 General

Several Action field formats are defined to support TDLS. A TDLS Action field, in the octet immediately after the Category field, differentiates the TDLS Action field formats. The TDLS Action field values associated with each Action field format within the TDLS category are defined in Table 8-238.

Table 8-238—TDLS Action field values

Action field value	Meaning
0	TDLS Setup Request
1	TDLS Setup Response
2	TDLS Setup Confirm
3	TDLS Teardown
4	TDLS Peer Traffic Indication
5	TDLS Channel Switch Request
6	TDLS Channel Switch Response
7	TDLS Peer PSM Request
8	TDLS Peer PSM Response
9	TDLS Peer Traffic Response
10	TDLS Discovery Request
11–255	Reserved

References to one of the TDLS Action field values as a frame, e.g., “TDLS Setup Request frame,” denote a Data frame carrying a TDLS Action Field and any vendor-specific elements tunneled as described in 10.22.1.

8.5.13.2 TDLS Setup Request Action field format

The Action field of a TDLS Setup Request Action field contains the information shown in Table 8-239.

Table 8-239—Information for TDLS Setup Request Action field

Order	Information	Notes
1	Category	The Category field is set to the value for TDLS, as defined in Table 8-38.
2	Action	The Action field is set to 0, representing TDLS Setup Request.
3	Dialog Token	The Dialog Token field contains a unique nonzero value for the conversation between the STAs involved in this request. The Dialog Token is specified in 8.4.1.12.
4	Capability	The Capability field indicates the capabilities of the STA. The Capability field is defined in 8.4.1.4.

Table 8-239—Information for TDLS Setup Request Action field *(continued)*

Order	Information	Notes
5	Supported rates	The Supported Rates element indicates the rates that are supported by the STA. The Supported Rates element is defined in 8.4.2.3.
6	Country	The Country element is present when dot11MultiDomainCapabilityActivated is true or dot11SpectrumManagementRequired is true. The Country element is defined in 8.4.2.10.
7	Extended supported rates	The Extended Supported Rates element is present whenever there are more than eight supported rates, and it is optionally present otherwise. The Extended Supported Rates element is defined in 8.4.2.15.
8	Supported Channels	The Supported Channels element is present if the TDLS channel switching capability field is equal to 1. The Supported Channels element is defined in 8.4.2.20.
9	RSNE	The RSNE is present if security is required on the direct link. The RSNE is defined in 8.4.2.27.
10	Extended Capabilities	The Extended Capabilities element is optionally present if any of the fields in this element are nonzero. The Extended Capabilities element is defined in 8.4.2.29.
11	QoS Capability	The QoS Capability element is present when dot11QosOptionImplemented is true and not present otherwise. The QoS Capability element is defined in 8.4.2.37.
12	FTE	The FTE is present if security is required on the TDLS direct link. The FTE is defined in 8.4.2.50.
13	Timeout Interval (TPK Key Lifetime)	The Timeout Interval element contains the TPK Key Lifetime and is present if security is required on the direct link. The Timeout Interval element is defined in 8.4.2.51.
14	Supported Operating Classes	The Supported Operating Classes element is present if the TDLS channel switching capability field is equal to 1. The Supported Operating Classes element is defined in 8.4.2.56 (optional).
15	HT Capabilities	The HT Capabilities element is defined in 8.4.2.58. The HT Capabilities element is present when dot11HighThroughputOptionImplemented is true.
16	20/40 BSS Coexistence	The 20/40 BSS Coexistence element is defined in 8.4.2.62. The 20/40 BSS Coexistence element is optionally present.
17	Link Identifier	The Link Identifier element is specified in 8.4.2.64.

The TDLS Setup Request Action field is encapsulated in a Data frame and transmitted to the recipient STA through the AP to request the setup of a TDLS direct link. See 10.22.

8.5.13.3 TDLS Setup Response Action field format

The Action field of a TDLS Setup Response Action field contains the information shown in Table 8-240.

Table 8-240—Information for TDLS Setup Response Action field

Order	Information	Notes
1	Category	The Category field is set to the value for TDLS, as defined in Table 8-38.
2	Action	The Action field is set to 1, representing TDLS Setup Response.
3	Status Code	The Status Code is defined in 8.4.1.9.
4	Dialog Token	The Dialog Token is copied from the corresponding TDLS Setup Request. The Dialog Token is specified in 8.4.1.12.
5	Capability	The Capability field indicates the capabilities of the STA and is present when the Status Code is 0 (successful), and is not present otherwise. The Capability field is defined in 8.4.1.4.
6	Supported rates	The Supported Rates element indicates the rates that are supported by the STA and is present when the Status Code is 0 (successful), and is not present otherwise.The Supported Rates element is defined in 8.4.2.3.
7	Country	The Country element is present when Status Code is 0 and either dot11MultiDomainCapabilityActivated is true or dot11SpectrumManagementRequired is true, and is not present otherwise. The Country element is defined in 8.4.2.10
8	Extended supported rates	The Extended Supported Rates element is present when there are more than 8 supported rates and Status Code is 0. It is optionally present when there are less than 8 supported rates and Status Code is 0. Otherwise it is not present. The Extended Supported Rates element is defined in 8.4.2.15.
9	Supported Channels	The Supported Channels element is defined in 8.4.2.20. It is present if the TDLS channel switching capability bit is equal to 1 and the Status Code is 0 (Successful), and not present otherwise.
10	RSNE	The RSNE is present if security is required on the TDLS direct link and the Status Code is 0 (successful), and not present otherwise. The RSNE is defined in 8.4.2.27.
11	Extended Capabilities	The Extended Capabilities element is present if any of the fields in this element are nonzero and the Status Code is 0 (Successful), and not present otherwise. The Extended Capabilities element is defined in 8.4.2.29.
12	QoS Capability	The QoS Capability element is present when dot11QosOptionImplemented is true and the Status Code is 0 (Successful) and not present otherwise. The QoS Capability element is defined in 8.4.2.37.
13	FTE	The FTE is present if security is required on the TDLS direct link and the Status Code is 0 (Successful), and not present otherwise. The FTE is defined in 8.4.2.50.
14	Timeout Interval (TPK Key Lifetime)	The Timeout Interval element, containing the TPK Key Lifetime, is present if security is required on the direct link and the Status Code is 0 (successful), and not present otherwise. The Timeout Interval element is defined in 8.4.2.51.
15	Supported Operating Classes	The Supported Operating Classes element is present if the TDLS channel switching capability bit is equal to 1 and the Status Code is 0 (successful) and not present otherwise. The Supported Operating Classes element is defined in 8.4.2.56.

Table 8-240—Information for TDLS Setup Response Action field *(continued)*

Order	Information	Notes
16	HT Capabilities	The HT Capabilities element is present if dot11HighThroughputOptionImplemented is true and the Status Code is 0 (successful) and is not present otherwise. The HT Capabilities element is defined in 8.4.2.58.
17	20/40 BSS Coexistence	The 20/40 BSS Coexistence element is optionally present if the Status Code is 0 and not present otherwise. The 20/40 BSS Coexistence element is defined in 8.4.2.62.
18	Link Identifier	The Link Identifier element is present if the Status Code is 0. The Link Identifier is specified in 8.4.2.64.

The TDLS Setup Response Action field is encapsulated in a Data frame and transmitted to the TDLS initiator STA through the AP in response to a received TDLS Setup Request Action field. See 10.22.

8.5.13.4 TDLS Setup Confirm Action field format

The Action field of a TDLS Setup Confirm Action field contains the information shown in Table 8-241.

Table 8-241—Information for TDLS Setup Confirm Action field

Order	Information	Notes
1	Category	The Category field is set to the value for TDLS, as defined in Table 8-38.
2	Action	The Action field is set to 2, representing TDLS Setup Confirm.
3	Status Code	The Status Code is defined in 8.4.1.9.
4	Dialog Token	The Dialog Token is copied from the corresponding TDLS Setup Response. The Dialog Token is specified in 8.4.1.12.
5	RSNE	The RSNE is present if security is required on the TDLS direct link and the Status Code is 0 (successful), and not present otherwise. The RSNE is defined in 8.4.2.27.
6	EDCA Parameter Set	The EDCA parameter set is present if QoS is supported on the direct link. The EDCA Parameter Set element is specified in 8.4.2.31. It is present for Status Code 0 (Successful).
7	FTE	The FTE is defined in 8.4.2.50. The FTE is present if security is required on the TDLS direct link and the Status Code is 0 (Successful), and not present otherwise.
8	Timeout Interval (TPK Key Lifetime)	The Timeout Interval element is defined in 8.4.2.51. The Timeout Interval, containing the TPK Key Lifetime, is present if security is required on the direct link and the Status Code is 0 (successful), and not present otherwise.
9	HT Operation	The HT Operation element is present if dot11HighThroughputOptionImplemented is true, the TDLS Setup Response Action field contained an HT Capabilities element, the Status Code is 0 (success), and the BSS does not support HT. The HT Operation element is defined in 8.4.2.59.
10	Link Identifier	The Link Identifier is specified in 8.4.2.64. It is present if the Status Code is 0.

The TDLS Setup Confirm Action field is encapsulated in a Data frame and transmitted to the TDLS responder STA through the AP in response to a received TDLS Setup Response Action field. See 10.22.

8.5.13.5 TDLS Teardown Action field format

The Action field of a TDLS Teardown Action field contains the information shown in Table 8-242.

Table 8-242—Information for TDLS Teardown Action field

Order	Information	Notes
1	Category	The Category field is set to the value for TDLS, as defined in Table 8-38.
2	Action	The Action field is set to 3, representing TDLS Teardown.
3	Reason Code	The Reason Code is defined in 8.4.1.7.
4	FTE	The FTE is present if a TPK handshake was successful for this session (optional). The FTE is defined in 8.4.2.50.
5	Link Identifier	The Link Identifier is specified in 8.4.2.64.

The TDLS Teardown Action field is encapsulated in a Data frame and transmitted to the TDLS peer STA directly or through the AP to tear down a TDLS direct link. See 10.22.

8.5.13.6 TDLS Peer Traffic Indication Action field format

The Action field of a TDLS Peer Traffic Indication Action field contains the information shown in Table 8-243.

Table 8-243—Information for TDLS Peer Traffic Indication Action field

Order	Information	Notes
1	Category	The Category field is set to the value for , as defined in Table 8-38
2	Action	The Action field is set to 4, representing Peer Traffic Indication.
3	Dialog Token	The Dialog Token is specified in 8.4.1.12.
4	Link Identifier	The Link Identifier is specified in 8.4.2.64.
5	PTI Control	The PTI Control element is optionally present. It is defined in 8.4.2.67.
6	TPU Buffer Status	The TPU Buffer Status element is defined in 8.4.2.68.

The TDLS Peer Traffic Indication Action field indicates the state of the power save buffer at the STA supporting TDLS Peer U-APSD that is buffering data for a TDLS peer STA in power save mode.

The TPU Buffer Status element indicates the status of the AC buffers at the TPU buffer STA.

The PTI Control element is optionally included in the TDLS Peer Traffic Indication Action field (see 10.2.1.15) to identify the latest MPDU transmitted to the TPU sleep STA that is the destination of the TDLS Peer Traffic Indication Action field.

The TDLS Peer Traffic Indication Action field is encapsulated in a Data frame and transmitted to the TDLS peer STA through the AP. See 10.2.1.15.

8.5.13.7 TDLS Channel Switch Request Action field format

The Action field of the TDLS Channel Switch Request Action field contains the information shown in Table 8-244.

Table 8-244—Information for TDLS Channel Switch Request Action field

Order	Information	Notes
1	Category	The Category field is set to the value for TDLS, as defined in Table 8-38.
2	Action	The Action field is set to 5, representing TDLS Channel Switch Request.
3	Target Channel	1 octet field that specifies the channel number of the target channel. See 8.4.1.35.
4	Operating Class	1 octet field that specifies the operating class for the target channel. The Operating Channel field is defined in 8.4.1.36.
5	Secondary Channel Offset	The secondary channel offset is present if a switch to a 40 MHz direct link is indicated and is absent otherwise. See 8.4.2.22.
6	Link Identifier	The Link Identifier element is specified in 8.4.2.64.
7	Channel Switch Timing	The Channel Switch Timing element is specified in 8.4.2.66.

The TDLS Channel Switch Request Action field is encapsulated in a Data frame and transmitted directly to the TDLS peer STA to request for the TDLS direct link to be switched to another channel. See 10.22.

8.5.13.8 TDLS Channel Switch Response Action field format

The Action field of the TDLS Channel Switch Response Action field contains the information shown in Table 8-245.

Table 8-245—Information for TDLS Channel Switch Response Action field

Order	Information	Notes
1	Category	The Category field is set to the value for TDLS, as defined in Table 8-38.
2	Action	The Action field is set to 6, representing TDLS Channel Switch Response.
3	Status Code	The Status Code is defined in 8.4.1.9.
5	Link Identifier	The Link Identifier element is specified in 8.4.2.64. It is present if the Status Code is 0.
6	Channel Switch Timing	The Channel Switch Timing element is specified in 8.4.2.66.

The TDLS Channel Switch Response Action field is encapsulated in a Data frame and transmitted directly to the TDLS peer STA in response to a received TDLS Channel Switch Request Action field. See 10.22.

8.5.13.9 TDLS Peer PSM Request Action field format

The TDLS Peer PSM Request Action field contains the information shown in Table 8-246.

Table 8-246—Information for TDLS Peer PSM Request Action field

Order	Information	Notes
1	Category	The Category field is set to the value for TDLS, as defined in Table 8-38.
2	Action	The Action field is set to 7, representing TDLS Peer PSM Request.
3	Dialog Token	The Dialog Token contains a value that is unique among TDLS Peer PSM Request Action fields for which a corresponding TDLS Peer PSM Response Action field has not been received. The Dialog Token is specified in 8.4.1.12.
4	Link Identifier	The Link Identifier element is specified in 8.4.2.64.
5	Wakeup Schedule	The Wakeup Schedule element is specified in 8.4.2.65.

The TDLS Peer PSM Request Action field is encapsulated in a Data frame and transmitted to the TDLS peer STA, directly or through the AP, to setup or change a periodic wakeup schedule on the TDLS direct link. See 10.2.1.14.

8.5.13.10 TDLS Peer PSM Response Action field format

The TDLS Peer PSM Response Action field contains the information shown in Table 8-247.

Table 8-247—Information for TDLS Peer PSM Response Action field

Order	Information	Notes
1	Category	The Category field is set to the value for TDLS, as defined in Table 8-38.
2	Action	The Action field is set to 8, representing TDLS Peer PSM Response.
3	Dialog Token	The Dialog Token is set to the value contained in the corresponding TDLS Peer PSM Request Action field. The Dialog Token is specified in 8.4.1.12.
4	Status Code	The Status Code is specified in 8.4.1.9.
5	Link Identifier	The Link Identifier element is specified in 8.4.2.64. It is present if the Status Code is 0.
6	Wakeup Schedule	The Wakeup Schedule element is present is present if the status code is set to 2 (“TDLS Wakeup Schedule rejected but alternative schedule provided”) and is not present otherwise. The Wakeup Schedule is specified in 8.4.2.65.

The TDLS Peer PSM Response Action field is encapsulated in a Data frame and transmitted to the TDLS peer STA directly in response to a TDLS Peer PSM Request Action field. See 10.2.1.14.

8.5.13.11 TDLS Peer Traffic Response Action field format

The Action field of a TDLS Peer Traffic Response Action field contains the information shown in Table 8-248.

Table 8-248—Information for TDLS Peer Traffic Response Action field

Order	Information	Notes
1	Category	The Category field is set to the value for TDLS, as defined in Table 8-38.
2	Action	The Action field is set to 9, representing TDLS Peer Traffic Response.
3	Dialog Token	The Dialog Token is specified in 8.4.1.12.
4	Link Identifier	The Link Identifier element is specified in 8.4.2.64.

The TDLS Peer Traffic Response Action field indicates the receipt of the corresponding TDLS Peer Traffic Indication Action field.

The Dialog Token field is set to the nonzero value of the corresponding TDLS Peer Traffic Indication Action field.

The Link Identifier field is set to identify the TDLS direct link in relation to which the TDLS Peer Traffic Response Action field is transmitted.

The Peer Traffic Response Action field is encapsulated in a Data frame and transmitted to the TDLS peer STA directly. See 10.2.1.15.

8.5.13.12 TDLS Discovery Request Action field format

The TDLS Discovery Request Action field contains the information shown in Table 8-249.

Table 8-249—Information for TDLS Discovery Request Action field

Order	Information	Notes
1	Category	The Category field is set to the value for TDLS, as defined in Table 8-38.
2	Action	The Action field is set to 10, representing TDLS Discovery Request.
3	Dialog Token	The Dialog Token can be used to match TDLS Discovery Response frames to the corresponding TDLS Discovery Request frame. The Dialog Token is specified in 8.4.1.12.
4	Link Identifier	The Link Identifier element is specified in 8.4.2.64.

The TDLS Discovery Request Action field is encapsulated in a Data frame and transmitted to a TDLS peer STA through the AP, to request TDLS capable STAs in the same BSS to respond with a TDLS Discovery Response frame. See 10.22.

8.5.14 WNM Action details

8.5.14.1 WNM Action fields

Several Action frame formats are defined for wireless network management (WNM) purposes. An Action field, in the octet field immediately after the Category field, differentiates the formats. The Action field values associated with each frame format are defined in Table 8-250.

Table 8-250—WNM Action field values

Action field value	Description
0	Event Request
1	Event Report
2	Diagnostic Request
3	Diagnostic Report
4	Location Configuration Request
5	Location Configuration Response
6	BSS Transition Management Query
7	BSS Transition Management Request
8	BSS Transition Management Response
9	FMS Request
10	FMS Response
11	Collocated Interference Request
12	Collocated Interference Report
13	TFS Request
14	TFS Response
15	TFS Notify
16	WNM-Sleep Mode Request
17	WNM-Sleep Mode Response
18	TIM Broadcast Request
19	TIM Broadcast Response
20	QoS Traffic Capability Update
21	Channel Usage Request
22	Channel Usage Response
23	DMS Request
24	DMS Response
25	Timing Measurement Request
26	WNM-Notification Request

Table 8-250—WNM Action field values *(continued)*

Action field value	Description
27	WNM-Notification Response
28–255	Reserved

8.5.14.2 Event Request frame format

The Event Request frame uses the Action frame body format and is transmitted by a STA to request another STA to report one or more events. The format of the frame is shown in Figure 8-467.

	Category	Action	Dialog Token	Event Request Elements	Destination URI Element (optional)
Octets:	1	1	1	variable	variable

Figure 8-467—Event Request frame body format

The Category field is the value indicating the WNM category, as specified in Table 8-38 in 8.4.1.11.

The Action field is the value indicating Event Request frame, as specified in Table 8-250 in 8.5.14.1.

The Dialog Token field is a nonzero value chosen by the STA sending the event request to identify the request/report transaction.

The Event Request Elements field contains one or more of the Event Request elements described in 8.4.2.69.

The Destination URI element field contains zero or one Destination URI element described in 8.4.2.92.

8.5.14.3 Event Report frame format

The Event Report frame uses the Action frame body format and is transmitted by a STA in response to an Event Request frame, or autonomously. The format of the Event Report frame body is shown in Figure 8-468.

	Category	Action	Dialog Token	Event Report Elements
Octets:	1	1	1	variable

Figure 8-468—Event Report frame body format

The Category field is the value indicating the WNM category, as specified in Table 8-38 in 8.4.1.11.

The Action field is the value indicating Event Report, as specified in Table 8-250 in 8.5.14.1.

The Dialog Token field is the nonzero value of the corresponding Event Request frame. If the Event Report frame is being transmitted other than in response to an Event Request frame, then the Dialog token is 0.

The Event Report Elements field contains one or more of the Event Report elements described in 8.4.2.70.

8.5.14.4 Diagnostic Request frame format

The Diagnostic Request frame uses the Action frame body format and is transmitted by a STA requesting the receiving STA to execute a diagnostic test. The format of the frame is shown in Figure 8-469.

	Category	Action	Dialog Token	Diagnostic Request Elements	Destination URI Element (optional)
Octets:	1	1	1	variable	variable

Figure 8-469—Diagnostic Request frame body format

The Category field is the value indicating the WNM category, as specified in Table 8-38 in 8.4.1.11.

The Action field is the value indicating Diagnostic Request frame, as specified in Table 8-250 in 8.5.14.1.

The Dialog Token field is a nonzero value chosen by the STA sending the Diagnostic request frame to identify the request/report transaction.

The Diagnostic Request Elements field contains one or more of the Diagnostic Request elements described in 8.4.2.71. The number and length of the Diagnostic Request elements in a Diagnostic Request frame is limited by the maximum MMPDU size (see 8.3.3.1).

The Destination URI element field contains zero or one Destination URI element described in 8.4.2.92.

8.5.14.5 Diagnostic Report frame format

The Diagnostic Report frame uses the Action frame body format transmitted by a STA in response to a Diagnostic Request frame. The format of the Diagnostic Report frame body is shown in Figure 8-470.

	Category	Action	Dialog Token	Diagnostic Report Elements
Octets:	1	1	1	variable

Figure 8-470—Diagnostic Report frame body format

The Category field is the value indicating the WNM category, as specified in Table 8-38 in 8.4.1.11.

The Action field is the value indicating Diagnostic Report, as specified in Table 8-250 in 8.5.14.1.

The Dialog Token field is the nonzero value in any corresponding Diagnostic Request frame.

The Diagnostic Report Elements field contains one or more of the Diagnostic Report elements described in 8.4.2.72. The number and length of the Diagnostic Report elements in a Diagnostic Report frame is limited by the maximum MMPDU size (see 8.3.3.1).

8.5.14.6 Location Configuration Request frame format

The Location Configuration Request frame uses the Action frame body format and is transmitted by a STA to configure another STA to send a Location Track Notification frame on a set of channels periodically for the purposes of determining location of the STA. The format of the Location Configuration Request frame body is shown in Figure 8-471.

	Category	Action	Dialog Token	Location Parameters Element
Octets:	1	1	1	variable

Figure 8-471—Location Configuration Request frame body format

The Category field is the value indicating the WNM category, as specified in Table 8-38 in 8.4.1.11.

The Action field is the value indicating Location Configuration Request, as specified in Table 8-250 in 8.5.14.1.

The Dialog Token field is a nonzero value that is unique among the Location Configuration Request frames sent to each destination MAC address for which a corresponding Location Configuration Response frame has not been received.

The Location Parameters Element field contains the location parameters subelements, described in 8.4.2.73. Table 8-251 defines the allowed Location Parameters subelements for a Location Parameters element that is included in the Location Configuration Request frame.

Table 8-251—Location Parameters Element field for Location Configuration Request frame

Allowed subelement fields	Subelement ID	Notes
Location Indication Parameters	1	The Location Indication Parameters subelement is included in the Location Configuration Request frame.
Location Indication Channels	2	The Location Indication Channels subelement is included in the Location Configuration Request frame.
Location Indication Broadcast Data Rate	6	The Location Indication Broadcast Data Rate subelement is included in the Location Configuration Request frame.
Location Indication Options	8	The Location Indication Options subelement may be included in the Location Configuration Request frame.
Vendor Specific Information	221	The Vendor Specific Information subelement may be included in the Location Configuration Request frame.

8.5.14.7 Location Configuration Response frame format

The Location Configuration Response frame uses the Action frame body format and is transmitted by a STA in response to the receipt of a Location Configuration Request frame. The format of the Location Configuration Response frame body is shown in Figure 8-472.

	Category	Action	Dialog Token	Location Parameters Element
Octets:	1	1	1	variable

Figure 8-472—Location Configuration Response frame body format

The Category field is the value indicating the WNM category, as specified in Table 8-38 in 8.4.1.11.

The Action field is the value indicating Location Configuration Response, as specified in Table 8-250 in 8.5.14.1.

The Dialog Token field is the nonzero value received in the Location Configuration Request frame to identify the request/response transaction.

The Location Parameters Element field contains the location parameters subelements, described in 8.4.2.73. Table 8-252 defines the allowed Location Parameters subelements for a Location Parameters element that is included in the Location Configuration Response frame.

Table 8-252—Location Parameters Element field for Location Configuration Response frame

Allowed subelement fields	Subelement ID	Notes
Location Indication Parameters	1	The Location Indication Parameters subelement may be included in the Location Configuration Response frame.
Location Indication Channels	2	The Location Indication Channels subelement may be included in the Location Configuration Response frame.
Location Indication Broadcast Data Rate	6	The Location Indication Broadcast Data Rate subelement may be included in the Location Configuration Response frame.
Location Indication Options	8	The Location Indication Options subelement may be included in the Location Configuration Response frame.
Location Status	3	The Location Status subelement is included in the Location Configuration Response frame. If all configuration of the subelements contained in a Location Configuration Request frame was successful, then a single Location Status subelement is included in the Location Configuration Response frame. For each subelement contained in the Location Configuration Request frame that is not successful a Location Status subelement is included in the Location Configuration Response frame that indicates the subelement ID and the unsuccessful status code for that subelement ID.
Vendor Specific Information	221	The Vendor Specific Information subelement may be included in the Location Configuration Response frame.

8.5.14.8 BSS Transition Management Query frame format

The BSS Transition Management Query frame uses the Action frame body format and is transmitted by a STA requesting or providing information on BSS transition candidate APs. The format of the BSS Transition Management Query frame body is shown in Figure 8-473.

	Category	Action	Dialog Token	BSS Transition Query Reason	BSS Transition Candidate List Entries (Optional)
Octets:	1	1	1	1	variable

Figure 8-473—BSS Transition Management Query frame body format

The Category field is the value indicating the WNM category, as specified in Table 8-38 in 8.4.1.11.

The Action field is the value indicating BSS Transition Management Query, as specified in Table 8-250 in 8.5.14.1.

The Dialog Token field is a nonzero value chosen by the STA sending the BSS Transition Management Query to identify the query/request/response transaction.

The BSS Transition Query Reason field contains the reason code for a BSS transition management query as defined in Table 8-138.

The BSS Transition Candidate List Entries field contains zero or more Neighbor Report elements, as described in 8.4.2.39. The Neighbor Report elements are collected by the STA as part of its scanning procedures and provided to the AP as described in 10.23.6.2. The length of the BSS Transition Candidate List Entries field in a BSS Transition Management Query frame is limited by the maximum MMPDU size (see 8.3.3.1).

8.5.14.9 BSS Transition Management Request frame format

The BSS Transition Management Request frame uses the Action frame body format and is transmitted by an AP STA in response to a BSS Transition Management Query frame, or autonomously. The format of the BSS Transition Management Request frame body is shown in Figure 8-474.

	Category	Action	Dialog Token	Request mode	Disassociation Timer
Octets:	1	1	1	1	2

	Validity Interval	BSS Termination Duration (optional)	Session Information URL (optional)	BSS Transition Candidate List Entries
Octets:	1	0 or 12	variable	variable

Figure 8-474—BSS Transition Management Request frame body format

The Category field is the value indicating the WNM category, as specified in Table 8-38 in 8.4.1.11.

The Action field is the value indicating BSS Transition Management Request frame, as specified in Table 8-250 in 8.5.14.1.

The Dialog Token field is the nonzero value received in the BSS Transition Management Query frame if the BSS Transition Management Request frame is being transmitted in response to a BSS Transition Management Query frame. If the BSS Transition Management Request frame is being transmitted other than in response to a BSS Transition Management Query frame, then the Dialog Token field is a nonzero value chosen by the AP STA sending the BSS Transition Management Request frame to identify the request/response transaction.

The Request mode field is defined in Figure 8-475.

	Preferred Candidate List Included	Abridged	Disassociation Imminent	BSS Termination Included	ESS Disassociation Imminent	Reserved
Bit:	0	1	2	3	4	5- 7

Figure 8-475—Request Mode field

- The Preferred Candidate List Included (bit 0) field indicates whether the BSS transition candidate list included in this frame is a preferred candidate list or a list of known BSS transition candidates. The Preferred Candidate List Included bit set to 0 indicates that the receiving STA may ignore the BSS Transition Candidate List Entries field. The Preferred Candidate List Included bit set to 1 indicates that the sender expects the receiving STA to process this frame.
- The Abridged (bit 1) field indicates to the recipient of the frame the intended treatment of all BSSIDs not listed in the BSS Transition Candidate List Entries field. The AP sets the Abridged bit in the Request Mode field to 1 when a preference value of 0 is assigned to all BSSIDs that do NOT appear in the BSS Transition Candidate List. The AP sets the Abridged bit in the Request Mode field to 0 when the AP has no recommendation for or against any BSSID not present in the BSS Transition Candidate List Entries field.
- The Disassociation Imminent (bit 2) field indicates whether the STA will be disassociated from the current AP. The Disassociation Imminent bit in the Request Mode field set to 1 indicates that STA is to be disassociated from the current AP. The Disassociation Imminent field set to 0 indicates that disassociation from the AP is not imminent.
- The BSS Termination Included (bit 3) field indicates that the BSS Termination Duration field is included, the BSS is shutting down and the STA will be disassociated. The AP sets the BSS Termination Included bit in the Request mode field to 1 to indicate that the BSS is shutting down. The BSS Termination Included bit is 0 if no BSS Termination Duration information is included in the BSS Transition Management Request frame.
- The ESS Disassociation Imminent (bit 4) field indicates that the Session Information URL field is included, and that the STA will be disassociated from the ESS. The ESS Disassociation Imminent bit in the Request Mode field set to 1 indicates that STA is to be disassociated from the ESS. When the ESS Disassociation Imminent bit is 1, a Session Information URL field is included in the BSS Transition Management Request frame. The ESS Disassociation Imminent field set to 0 indicates that disassociation from the ESS is not imminent.

The Disassociation Timer indicates the time after which the AP will issue a Disassociation frame to this STA. The Disassociation Timer field is the number of beacon transmission times (TBTTs) until the AP sends a Disassociation frame to this STA. A value of 0 indicates that the AP has not determined when it will send a Disassociation frame to this STA. If the Disassociation Imminent field is 0, the Disassociation Timer field is reserved. The format of the Disassociation Timer field is shown in Figure 8-476.

	B0 B15
	Disassociation Timer
Bits	16

Figure 8-476—Disassociation Timer field format

The Validity Interval field is the number of beacon transmission times (TBTTs) until the BSS transition candidate list is no longer valid. A value of 0 is reserved.

The BSS Termination Duration field contains the BSS Termination Duration subelement (see 8.4.2.39) for the current BSS and is present only when the BSS Termination Included field is 1 in the Request mode field.

The format of the optional Session Information URL field is shown in Figure 8-477. This field is present when the ESS Disassociation Imminent field is 1.

	URL Length	URL (optional)
Octets:	1	variable

Figure 8-477—Session Information URL field format

The URL Length field is the value of the length of the URL field. The URL Length field is 0 when the URL field is not present.

The URL field is a variable-length field formatted in accordance with IETF RFC 3986-2005.

The BSS Transition Candidate List Entries field contains one or more Neighbor Report elements described in 8.4.2.39. If the STA has no information in response to the BSS Transition Management Query frame, the Neighbor Report elements are omitted and the Preferred Candidate List Included bit is 0. The length of the BSS Transition Candidate List Entries in a BSS Transition Management Request frame is limited by the maximum MMPDU size (see 8.3.3.1).

8.5.14.10 BSS Transition Management Response frame format

The BSS Transition Management Response frame uses the Action frame body format and is optionally transmitted by a STA in response to a BSS Transition Management Request frame. The format of the BSS Transition Management Response frame body is shown in Figure 8-478.

	Category	Action	Dialog Token	Status code	BSS Termination Delay	Target BSSID (Optional)	BSS Transition Candidate List Entries (Optional)
Octets:	1	1	1	1	1	0 or 6	variable

Figure 8-478— BSS Transition Management Response frame body format

The Category field is the value indicating the WNM category, as specified in Table 8-38 in 8.4.1.11.

The Action field is the value indicating BSS Transition Response, as specified in Table 8-250 in 8.5.14.1.

The Dialog Token field is the value in the corresponding BSS Transition Management Request frame. The BSS Transition Management Response frame is only transmitted in response to a BSS Transition Management Request frame.

The Status code field contains the status code in response to a BSS Transition Management Request frame as defined in Table 8-253. If the STA will transition to another BSS, then the status code is 0 (i.e., Accept). If the STA intends to retain the association with the current BSS, the status code is one of the “Reject” status codes.

Table 8-253—Status code definitions

Status code	Status code description
0	Accept
1	Reject—Unspecified reject reason.
2	Reject—Insufficient Beacon or Probe Response frames received from all candidates.
3	Reject—Insufficient available capacity from all candidates.
4	Reject—BSS Termination undesired.
5	Reject—BSS Termination delay requested.
6	Reject—STA BSS Transition Candidate List provided.
7	Reject—No suitable BSS transition candidates.
8	Reject—Leaving ESS.
9–255	Reserved

The BSS Termination Delay field is the number of minutes that the responding STA requests the BSS to delay termination. This field is reserved if the Status code field value is not set to 5.

The Target BSSID field is the BSSID of the BSS that the non-AP STA transitions to. This field is not present if the STA does not transition or if no transition information is available.

The BSS Transition Candidate List Entries field contains zero or more Neighbor Report elements described in 8.4.2.39. The Neighbor Report elements are collected by the STA as part of its scanning procedures and provided to the AP as described in 10.23.6.4. The length of the BSS Transition Candidate List Entries field in a BSS Transition Management Response frame is limited by the maximum MMPDU size (see 8.3.3.1).

8.5.14.11 FMS Request frame format

The FMS Request frame is sent by a non-AP STA to the AP to request the specified FMS and to propose delivery intervals for a set of group addressed streams. The FMS Request frame is also sent by a non-AP STA to request a modification to a previous FMS Request. The format of the frame is shown in Figure 8-479.

	Category	Action	Dialog Token	FMS Request Element
Octets:	1	1	1	variable

Figure 8-479—FMS Request frame format

The Category field is the value indicating the WNM category, as specified in Table 8-38 in 8.4.1.11.

The Action field is the value indicating FMS Request frame, as specified in Table 8-250 in 8.5.14.1.

The Dialog Token field is a nonzero value that is unique among the FMS Request frame sent to each destination MAC address for which a corresponding FMS Report frame has not been received.

The FMS Request Element field indicates the group addressed traffic streams that are requested by the non-AP STA. The FMS Request Element field contains one FMS Request element described in 8.4.2.78.

8.5.14.12 FMS Response frame format

The FMS Response frame is sent by an AP in response to an FMS Request frame, or sent by the AP to the STA to instruct the non-AP STA to change the delivery interval or data rate. The format of the frame is shown in Figure 8-480.

	Category	Action	Dialog Token	FMS Response Element
Octets:	1	1	1	variable

Figure 8-480—FMS Response frame format

The Category field is the value indicating the WNM category, as specified in Table 8-38 in 8.4.1.11.

The Action field is the value indicating FMS Response frame, as specified in Table 8-250 in 8.5.14.1.

The Dialog Token field is the nonzero value received in the FMS Request frame to identify the request/response transaction. If an FMS Response frame is being transmitted other than in response to an FMS Request frame, then the Dialog Token field is 0.

The FMS Response Element indicates the group addressed traffic streams that the AP supports and the corresponding delivery intervals. The FMS Response Element field contains one FMS Response elements, described in 8.4.2.79.

8.5.14.13 Collocated Interference Request frame format

The Collocated Interference Request frame uses the Action frame body format and is transmitted by a STA to request collocated interference reports, sent using Collocated Interference Report frames. The format of the Collocated Interference Request frame body is shown in Figure 8-481.

	Category	Action	Dialog Token	Request Info
Octets:	1	1	1	1

Figure 8-481—Collocated Interference Request frame format

The Category field is the value indicating the WNM category, as specified in Table 8-38 in 8.4.1.11.

The Action field is the value indicating Collocated Interference Request, as specified in Table 8-250 in 8.5.14.1.

The Dialog Token field is any nonzero value to identify the request/response transaction.

The Request Info field is shown in Figure 8-482.

	B0 B1	B2 B7
	Automatic Report Enabled	Report Timeout
Bits:	2	6

Figure 8-482—Request Info field format

The Automatic Report Enabled subfield contains an unsigned integer value.

The Automatic Report Enabled subfield set to 0 indicates that the requesting STA cancels the previous requests so that the receiving STA stops sending collocated interference reports.

The Automatic Report Enabled subfield set to 1 indicates that the requesting STA requests the receiving STA to send a collocated interference report when the STA knows of a change in the collocated interference, subject to meeting the Report Timeout requirement.

The Automatic Report Enabled subfield set to 2 indicates that the requesting STA requests the receiving STA to send a collocated interference report periodically using the period included in the Report Period field in the Collocated Interference Report element; see 8.4.2.87.

The Automatic Report Enabled subfield set to 3 indicates that the requesting STA requests the receiving STA to send collocated interference reports periodically using the period included in the Report Period field in the Collocated Interference Report element; see 8.4.2.87, or send a collocated interference report when the STA knows of a change in the collocated interference, subject to meeting the Report Timeout requirement.

The Report Timeout field contains a value in units of 200 TUs and indicates the minimum duration between two consecutive Collocated Interference Report frames from the reporting STA. When the Automatic Report Enabled subfield is 0, the Report Timeout field is reserved.

8.5.14.14 Collocated Interference Report frame format

The Collocated Interference Report frame uses the Action frame body format and is transmitted in response to Collocated Interference Request frame. The format of the Collocated Interference Report frame body is shown in Figure 8-483.

	Category	Action	Dialog Token	Collocated Interference Report Elements
Octets:	1	1	1	variable

Figure 8-483—Collocated Interference Report frame format

The Category field is the value indicating the WNM category, as specified in Table 8-38 in 8.4.1.11.

The Action field is the value indicating Collocated Interference Report frame, as specified in Table 8-250 in 8.5.14.1.

The Dialog Token field is the nonzero value received in the corresponding Collocated Interference Request frame to identify the request/response transaction.

The Collocated Interference Report Elements field contains one or more Collocated Interference Report elements to indicate the characteristics of the reported collocated interferences, as defined in 8.4.2.87.

8.5.14.15 TFS Request frame format

The TFS Request frame is sent by a non-AP STA to the AP to request the specified traffic filtering. The format of TFS Request frame is defined in Figure 8-484.

				One or more TFS Request elements
	Category	Action	Dialog Token	TFS Request Elements
Octets:	1	1	1	variable

Figure 8-484—TFS Request frame format

The Category field is the value indicating the WNM category, as specified in Table 8-38 in 8.4.1.11.

The Action field is the value indicating TFS Request frame, as specified in Table 8-250 in 8.5.14.1.

The Dialog Token field is a value chosen by the STA sending the TFS Request frame to identify the request/response transaction.

The TFS Request Elements field contains one or more TFS Request elements to specify the traffic filters that are requested by the non-AP STA, as defined in 8.4.2.82.

8.5.14.16 TFS Response frame format

The TFS Response frame is sent by an AP in response to a TFS Request frame. The format of the TFS Response frame is defined in Figure 8-485.

				One or more TFS Response elements
	Category	Action	Dialog Token	TFS Response Elements
Octets:	1	1	1	variable

Figure 8-485—TFS Response frame format

The Category field is the value indicating the WNM category, as specified in Table 8-38 in 8.4.1.11.

The Action field is the value indicating TFS Response frame, as specified in Table 8-250 in 8.5.14.1.

The Dialog Token field is the value in the corresponding TFS Request frame.

The TFS Response Elements field contains one or more TFS Response elements to indicate the traffic filters that the AP is configured to support, as defined in 8.4.2.83.

8.5.14.17 TFS Notify frame format

The TFS Notify frame is sent by an AP to a STA when a frame matching a traffic filter is encountered. The format of the TFS Notify frame is defined in Figure 8-486.

				One or more TFS IDs
	Category	Action	Number of TFS IDs	TFS ID List
Octets:	1	1	1	variable

Figure 8-486—TFS Notify frame format

The Category field is the value indicating the WNM category, as specified in Table 8-38 in 8.4.1.11.

The Action field is the value indicating TFS Notify frame, as specified in Table 8-250 in 8.5.14.1.

The Number of TFS IDs field indicates the number of 1-octet TFS IDs present in the TFS ID List field.

The TFS ID field indicates the traffic filter set containing the matched TCLAS element.

8.5.14.18 WNM-Sleep Mode Request frame format

The WNM-Sleep Mode Request frame is sent by a non-AP STA to the AP to enter the WNM-Sleep Mode. The format of the WNM-Sleep Mode Request frame is defined in Figure 8-487.

					One or more TFS Request elements
	Category	Action	Dialog Token	WNM-Sleep Mode Element	TFS Request Elements
Octets:	1	1	1	variable	variable

Figure 8-487—WNM-Sleep Mode Request frame format

The Category field is the value indicating the WNM category, as specified in Table 8-38 in 8.4.1.11.

The Action field is the value indicating WNM-Sleep Mode Request frame, as specified in Table 8-250 in 8.5.14.1.

The Dialog Token field is a value chosen by the non-AP STA sending the WNM-Sleep Mode Request frame to identify the request/response transaction.

The WNM-Sleep Mode Element field contains a WNM-Sleep Mode element that is requested by a non-AP STA, as described in 8.4.2.84.

The TFS Request Elements field contains one or more TFS Request elements to specify the traffic filters that are requested by a non-AP STA, as defined in 8.4.2.82.

8.5.14.19 WNM-Sleep Mode Response frame format

The WNM-Sleep Mode Response frame is sent by an AP in response to a WNM-Sleep Mode Request frame. The format of the WNM-Sleep Mode Response frame is defined in Figure 8-488.

	Category	Action	Dialog Token	Key Data Length	Key Data
Octets:	1	1	1	2	variable

		One or more TFS Response elements
	WNM-Sleep Mode Element	TFS Response Elements
Octets:	variable	variable

Figure 8-488—WNM-Sleep Mode Response frame format

The Category field is the value indicating the WNM category, as specified in Table 8-38 in 8.4.1.11.

The Action field is the value indicating WNM-Sleep Mode Response frame, as specified in Table 8-250 in 8.5.14.1.

The Dialog Token field is the value in the corresponding WNM-Sleep Mode Request frame.

The Key Data Length field is the length of the Key Data field. If the management frame protection is not used, this field is 0.

The Key Data field contains zero or more subelements that provide the current GTK and IGTK to the STA. The format of these subelements is given in Figure 8-489 and Figure 8-490. The subelement IDs for these subelements are defined in Table 8-254. Each subelement starts with the ID and Length fields. The Length field in the subelement is the length of the contents of the subelement. When management frame protection is not used, the Key Data field is not present.

Table 8-254—WNM-Sleep Mode subelement IDs

Value	Contents of subelement
0	GTK
1	IGTK
2–255	Reserved

The GTK subelement contains the Group Key as shown in Figure 8-489.

	Subelement ID	Length	Key Info	Key Length	RSC	Key
Octets:	1	1	2	1	8	5 to 32

Figure 8-489—WNM-Sleep Mode GTK subelement format

The Subelement ID field is 0.

The value of the Length field is 16 or 43.

The Key Info field is defined in Figure 11-29.

The Key Length field is the length of the Key field in octets.

The RSC field contains the receive sequence counter (RSC) for the GTK being installed. The RSC field gives the current message number for the GTK to allow a STA to identify replayed MPDUs. If the RSC field value is less than 8 octets in length, the remaining octets are set to 0. The least significant octet of the TSC or PN is in the first octet of the RSC field.

NOTE—The RSC field value for TKIP is the Transmit Sequence Counter (TSC), and is stored in the first 6 octets; for CCMP it is the Packet Number (PN), and is stored in the first 6 octets; see Table 11-5.

The IGTK subelement contains the Integrity GTK as shown in Figure 8-490.

	Subelement ID	Length	KeyID	PN	Key
Octets:	1	1	2	6	16

Figure 8-490—WNM-Sleep Mode IGTK subelement format

The Subelement ID field is 1.

The value of the Length field is 24.

The KeyID field indicates the value of the BIP key ID.

The PN field indicates the receive sequence counter for the IGTK being installed. The PN field gives the current message number for the IGTK, to allow a STA to identify replayed MPDUs.

The Key field is the IGTK being distributed.

NOTE 1—There may be multiple GTK and multiple IGTK subelements if a group rekeying is in process when the non-AP STA wakes up from WNM-Sleep mode.

NOTE 2—Management frame protection is used to provide confidentiality, replay, and integrity protection for GTK/IGTK update in WNM-Sleep Mode Response frames.

The WNM-Sleep Mode Element field contains a WNM-Sleep Mode element, as described in 8.4.2.84.

The TFS Response Elements field contains one or more TFS Response elements to specify the traffic filters, as defined in 8.4.2.83.

8.5.14.20 TIM Broadcast Request frame format

The format of the TIM Broadcast Request frame is shown in Figure 8-491.

	Category	Action	Dialog Token	TIM Broadcast Request Element
Octets:	1	1	1	3

Figure 8-491—TIM Broadcast Request frame format

The Category field is the value indicating the WNM category, as specified in Table 8-38 in 8.4.1.11.

The Action field is the value indicating TIM Broadcast Request frame, as specified in Table 8-250 in 8.5.14.1.

The Dialog Token field is a nonzero value chosen by the STA sending the TIM Broadcast Request frame to identify the request/response transaction.

The TIM Broadcast Request Element field contains a TIM Broadcast Request Element as specified in 8.4.2.85.

8.5.14.21 TIM Broadcast Response frame format

The format of the TIM Broadcast Response frame is shown in Figure 8-492.

	Category	Action	Dialog Token	TIM Broadcast Response Element
Octets:	1	1	1	3 or 12

Figure 8-492—TIM Broadcast Response frame format

The Category field is the value indicating the WNM category, as specified in Table 8-38 in 8.4.1.11.

The Action field is the value indicating TIM Broadcast Response frame, as specified in Table 8-250 in 8.5.14.1.

The Dialog Token field is the nonzero value of the corresponding TIM Broadcast Request frame. If the TIM Broadcast Response frame is being transmitted other than in response to a TIM Broadcast Request frame, then the Dialog token is 0.

The TIM Broadcast Response Element field contains a TIM Broadcast Response Element as specified in 8.4.2.86.

8.5.14.22 QoS Traffic Capability Update frame format

The QoS Traffic Capability Update frame is sent by a non-AP STA to the AP to update QoS Traffic Capability information. The format of the frame is shown in Figure 8-493.

	Category	Action	QoS Traffic Capability Flags
Octets:	1	1	1

Figure 8-493—QoS Traffic Capability Update frame format

The Category field is the value indicating the WNM category, as specified in Table 8-38 in 8.4.1.11.

The Action field is the value indicating QoS Traffic Capability Update frame, as specified in Table 8-250 in 8.5.14.1.

The QoS Traffic Capability Flags field is defined in Table 8-255. The QoS Traffic Capability Flags field comprises 1 octet, with the bits 4–6 serving as QoS Traffic Capability Flags. Each of the bits 4–6 serves as a flag for one of the three user priorities, UP 4–6. The field is 1 to indicate the expectation of generating traffic belonging to the corresponding user priority (UP). The field is 0 to indicate that such expectation does not exist. The use of the QoS Traffic Capability Update frame is described in 10.23.9.

Table 8-255—QoS Traffic Capability Flags definition

Bit	Description
0–3	Reserved
4	UP 4 Traffic
5	UP 5 Traffic
6	UP 6 Traffic
7	Reserved

8.5.14.23 Channel Usage Request frame format

The Channel Usage Request frame is sent by a non-AP STA to the AP to request the specified Channel Usage information. The format of the Channel Usage Request frame is defined in Figure 8-494.

	Category	Action	Dialog Token	Channel Usage Elements	Supported Operating Classes Element
Octets:	1	1	1	variable	variable

Figure 8-494—Channel Usage Request frame format

The Category field is the value indicating the WNM category, as specified in 8.4.1.11.

The Action field is the value indicating Channel Usage Request frame, as specified in 8.5.14.1.

The Dialog Token field is a nonzero value chosen by the non-AP STA sending the Channel Usage Request frame to identify the request/response transaction.

The Channel Usage Element field includes one or more Channel Usage elements described in 8.4.2.88 to identify the request Usage Mode.

The Supported Operating Classes Element field includes a Supported Operating Classes element to indicate the supported operating classes for the requested network type, consistent with the Country element advertised by the AP. The Supported Operating Classes is described in 8.4.2.56.

8.5.14.24 Channel Usage Response frame format

The Channel Usage Response frame is sent by an AP STA in response to a Channel Usage Request frame, or autonomously. The format of the Channel Usage Response frame is shown in Figure 8-495.

	Category	Action	Dialog Token	Channel Usage Elements	Country String	Power Constraint Element (optional)	EDCA Parameter Set Element (optional)
Octets:	1	1	1	variable	3	0 or 3	0 or 20

Figure 8-495—Channel Usage Response frame format

The Category field is the value indicating the WNM category, as specified in 8.4.1.11.

The Action field is the value indicating Channel Usage Response frame, as specified in 8.5.14.1.

The Dialog Token field is the nonzero value received in the Channel Usage Request frame if the Channel Usage Response frame is being transmitted in response to a Channel Usage Request frame. The Dialog Token field is 0 if the Channel Usage Response frame is being transmitted other than in response to a Channel Usage Request frame.

The Channel Usage Element field includes zero or more Channel Usage elements described in 8.4.2.88.

The Country String field is the value contained in the dot11CountryString attribute.

The Power Constraint Element field includes zero or one Power Constraint elements described in 8.4.2.16. The use of the Power Constraint element included in the Power Constraint Element field is described in 10.23.14.

The EDCA Parameter Set Element field includes zero or one EDCA Parameter Set elements described in 8.4.2.31. The use of the EDCA Parameter Set element included in the EDCA Parameter Set Element field is described in 10.23.14.

8.5.14.25 DMS Request frame format

The DMS Request frame is sent by a non-AP STA to the AP to define information about a DMS request to the AP. The format of the DMS Request frame is defined in Figure 8-496.

	Category	Action	Dialog Token	DMS Request Element
Octets:	1	1	1	variable

Figure 8-496—DMS Request frame format

The Category field is the value indicating the WNM category, as specified in Table 8-38 in 8.4.1.11.

The Action field is the value indicating DMS Request frame, as specified in Table 8-250 in 8.5.14.1.

The Dialog Token field is a nonzero value chosen by the non-AP STA sending the DMS Request frame to identify the request/response transaction.

The DMS Request Element field contains a DMS Request element as specified in 8.4.2.90.

8.5.14.26 DMS Response frame format

The DMS Response frame is sent by an AP in response to a DMS Request frame, or autonomously to terminate a requested DMS stream. The format of the DMS Response frame is shown in Figure 8-497.

	Category	Action	Dialog Token	DMS Response Element
Octets:	1	1	1	variable

Figure 8-497—DMS Response frame format

The Category field is the value indicating the WNM category, as specified in Table 8-38 in 8.4.1.11.

The Action field is the value indicating DMS Response as specified in Table 8-250 in 8.5.14.1.

The Dialog Token field is the nonzero value received in the DMS Request frame if the DMS Response frame is being transmitted in response to a DMS Request frame. The Dialog Token field is 0 if the DMS Response frame is being transmitted autonomously, and not in response to a DMS Request frame.

The DMS Response Element field contains a DMS Response element as specified in 8.4.2.91.

8.5.14.27 Timing Measurement Request frame format

The format of the Timing Measurement Request frame is shown in Figure 8-498.

	Category	Action	Trigger
Octets:	1	1	1

Figure 8-498—Timing Measurement Request frame format

The Category field is the value indicating the WNM category, as specified in Table 8-38 in 8.4.1.11.

The Action field is the value indicating Timing Measurement Request frame as specified in Table 8-250 in 8.5.14.1.

The Trigger field set to the value 1 indicates that the sending STA requests a Timing Measurement procedure at the receiving STA as defined in 10.23.5. The trigger field set to the value 0 indicates that the sending STA requests that the receiving STA stops sending Timing Measurement frames. Trigger field values 2–255 are reserved.

8.5.14.28 WNM-Notification Request frame format

The format of the WNM-Notification Request frame is shown in Figure 8-499.

	Category	Action	Dialog Token	Type	Optional Subelements
Octets:	1	1	1	1	variable

Figure 8-499—WNM-Notification Request frame format

The Category field is the value indicating the WNM category, as specified in Table 8-38 in 8.4.1.11.

The Action field is the value indicating WNM-Notification Request frame as specified in Table 8-250 in 8.5.14.1.

The Dialog Token field is set to a nonzero value chosen by the STA sending the WNM-Notification request to identify the request/response transaction.

The Type field indicates the type of WNM-Notification, as defined in Table 8-256.

Table 8-256—WNM-Notification type

Value	Description
0	Firmware Update Notification
1–255	Reserved

The Optional Subelements field format contains zero or more subelements, each consisting of a 1-octet Subelement ID field, a 1-octet Length field and a variable-length Data field, as shown in Figure 8-402. The optional subelements are ordered by nondecreasing subelement ID.

The Subelement ID field values for the defined optional subelements are shown in Table 8-257. A Yes in the Extensible column of a subelement listed in Table 8-257 indicates that the length of the subelement might be extended in future revisions or amendments of this standard. When the Extensible column of an element is Subelement, then the subelement might be extended in future revisions or amendments of this standard by defining additional subelements within the subelement. See 9.24.9.

Table 8-257—Optional subelement IDs for WNM-Notification Request

Subelement ID	Name	Length field (octets)	Extensible
0	AP Descriptor	10	
1	Firmware Version—Current	3 to 251	
2	Firmware Version—New	3 to 251	
3–220	Reserved		
221	Vendor Specific	1 to 239	
222–255	Reserved		

When the Type field is 0, the AP Descriptor, Firmware Version—Current, and Firmware Version—New optional subelements are included in the WNM-Notification Request frame. The AP descriptor field format is as defined in Figure 8-290. The Firmware Version—Current subelement contains the current firmware version, in the Firmware Version subelement format defined in Figure 8-296. The Firmware Version—New subelement contains the new firmware version, in the Firmware Version subelement format defined in Figure 8-296.

The Vendor Specific subelements have the same format as their corresponding elements (see 8.4.2.28). Multiple Vendor Specific subelements are optionally present in the list of optional subelements.

8.5.14.29 WNM-Notification Response frame format

The format of the WNM-Notification Response frame is shown in Figure 8-500.

	Category	Action	Dialog Token	Response Status	Optional Subelements
Octets:	1	1	1	1	variable

Figure 8-500—WNM-Notification Response frame format

The Category field is the value indicating the WNM category, as specified in Table 8-38 in 8.4.1.11.

The Action field is the value indicating WNM-Notification Response frame as specified in Table 8-250 in 8.5.14.1.

The Dialog Token field is set to the nonzero value of the corresponding WNM-Notification Request frame.

The Response Status field indicates the Response Status value, as defined in Table 8-258.

Table 8-258—WNM-Notification Response Status

Value	Description
0	Notification Acknowledged
1–255	Reserved

The Optional Subelements field format contains zero or more subelements, each consisting of a 1-octet Subelement ID field, a 1-octet Length field and a variable-length Data field, as shown in Figure 8-402. The optional subelements are ordered by nondecreasing subelement ID.

The Subelement ID field values for the defined optional subelements are shown in Table 8-259. A Yes in the Extensible column of a subelement listed in Table 8-259 indicates that the length of the subelement might be extended in future revisions or amendments of this standard. When the Extensible column of an element is Subelement, then the subelement might be extended in future revisions or amendments of this standard by defining additional subelements within the subelement. See 9.24.9.

Table 8-259—Optional subelement IDs for WNM-Notification Response

Subelement ID	Name	Length field (octets)	Extensible
0–220	Reserved		
221	Vendor Specific	1 to 239	
222–255	Reserved		

The Vendor Specific subelements have the same format as their corresponding elements (see 8.4.2.28). Multiple Vendor Specific subelements are optionally present in the list of optional subelements.

8.5.15 Unprotected WNM Action details

8.5.15.1 Unprotected WNM Action fields

Unprotected WNM Action frames are not encapsulated using mechanisms defined for robust management frames. An Action field, in the octet field immediately after the Category field, differentiates the formats. The Action field values associated with each frame format is defined in Table 8-260.

Table 8-260—Unprotected WNM Action field values

Action field value	Description
0	TIM
1	Timing Measurement
2–255	Reserved

8.5.15.2 TIM frame format

The format of the TIM frame is shown in Figure 8-501.

	Category	Action	Check Beacon	Timestamp	TIM Element
Octets:	1	1	1	8	6–257

Figure 8-501—TIM frame format

The Category field is the value indicating the Unprotected WNM category, as specified in Table 8-38 in 8.4.1.11.

The Action field is the value indicating TIM frame, as specified in Table 8-260 in 8.5.15.1.

The Check Beacon field is defined as an unsigned integer initialized to 0, that increments when a critical update to the Beacon frame has occurred; see 10.2.1.17.

The Timestamp field is defined in 8.4.1.10. The field contains a valid TSF timestamp when the TIM Broadcast Response frame contained a reason code “Accept, valid timestamp present in TIM frames.” The field is reserved otherwise.

The TIM Element field contains a TIM element as specified in 8.4.2.7. The bit corresponding to buffered group addressed frames is 0 for all BSSIDs and ignored upon reception.

8.5.15.3 Timing Measurement frame format

The Timing Measurement frame is used to support the timing measurement procedure described in 10.23.5. The format of the Timing Measurement frame is shown in Figure 8-502.

	Category	Action	Dialog Token	Follow Up Dialog Token	TOD
Octets:	1	1	1	1	4

	TOA	Max TOD Error	Max TOA Error
Octets:	4	1	1

Figure 8-502—Timing Measurement frame format

The Category field is the value indicating the Unprotected WNM category, as specified in Table 8-38 in 8.4.1.11.

The Action field is the value indicating Timing Measurement as specified in Table 8-260 in 8.5.15.1.

The Dialog Token field is a nonzero value chosen by the sending STA to identify the Timing Measurement frame as the first of a pair, with the second or follow-up Timing Measurement frame to be sent later. The Dialog Token field is set to 0 to indicate that the Timing Measurement frame will not be followed by a subsequent follow-up Timing Measurement frame.

The Follow Up Dialog Token is the nonzero value of the Dialog Token field of the previously transmitted Timing Measurement frame to indicate that it is the follow up Timing Measurement frame and that the TOD, TOA, Max TOD Error and Max TOA Error fields contain the values of the timestamps captured with the first Timing Measurement frame of the pair. The Follow Up Dialog Token is 0 to indicate that the Timing Measurement frame is not a follow up to a previously transmitted Timing Measurement frame. The value 0 in this field also indicates that the TOD, TOA, Max TOD Error, and Max TOA Error fields are reserved. See 10.23.5.

The TOD, TOA, Max TOD Error, and Max TOA Error fields are expressed in units of 10 ns.

The TOD field contains a timestamp that represents the time at which the start of the preamble of the previously transmitted Timing Measurement frame appeared at the transmit antenna port.

The TOA field contains a timestamp that represents the time at which the start of the preamble of the ACK to the previously transmitted Timing Measurement frame arrived at the receive antenna port.

NOTE—The values specified in the TOD and TOA fields are described in 6.3.68.

The Max TOD Error field contains an upper bound for the error in the value specified in the TOD field. For instance, a value of 2 in the Max TOD Error field indicates that the value in the TOD field has a maximum error of ± 20 ns.

The Max TOA Error field contains an upper bound for the error in the value specified in the TOA field. For instance, a value of 2 in the Max TOA Error field indicates that the value in the TOA field has a maximum error of ± 20 ns.

A value of 0 for the Max TOD Error or the Max TOA Error field indicates that the upper bound on the error in the corresponding TOD or TOA value is unknown. A value of 255 indicates that the upper bound on the error is greater than or equal to 2.55 µs.

8.5.16 Self-protected Action frame details

8.5.16.1 Self-protected Action fields

The Self-protected Action frame is defined to allow robust STA-STA communications of the Action frames that are not robust (see 8.4.1.11). The protocols that use these Action frames are responsible for deciding whether to protect these frames and supporting protection mechanisms for these frames as needed.

Self-protected Action frames have a different nature than Public Action frames and Robust Action frames. Robust Action frames assume the existence of a completely established security association. Self-protected Action frames typically exist to manage the creation and destruction of security associations, regardless of whether they are completely established.

Public Action frames are defined as public for all STAs, including those that are not in the BSS and MBSS. Self-protected Action frames, however, are used for relationship creation and maintenance between two specific STAs. Their public nature is incidental.

A Self-protected Action field, in the octet field immediately after the Category field, differentiates the formats. The defined Self-protected Action frames are listed in Table 8-261.

Table 8-261—Self-protected Action field values

Self-protected Action field value	Description
0	Reserved
1	Mesh Peering Open
2	Mesh Peering Confirm
3	Mesh Peering Close
4	Mesh Group Key Inform
5	Mesh Group Key Acknowledge
6–255	Reserved

The Mesh Peering Open frame, the Mesh Peering Confirm frame, and the Mesh Peering Close frame are referred to as “Mesh Peering Management frames.”

8.5.16.2 Mesh Peering Open frame format

8.5.16.2.1 Mesh Peering Open frame self protection

Protection of this frame is provided when authenticated mesh peering exchange (AMPE) is enabled. AMPE provides integrity protection of Mesh Peering Open frames.

When the Mesh Peering Open frame is used by the mesh peering management (MPM) protocol, integrity protection on the frame is not enabled.

8.5.16.2.2 Mesh Peering Open frame details

The Mesh Peering Open frame is used to open a mesh peering using the procedures defined in 13.3.6 and in 13.5.5. The Mesh Peering Open frame is also, together with Mesh Peering Confirm and Mesh Peering Close frames, referred to as a Mesh Peering Management frame. The format of the Mesh Peering Open frame Action field is shown in Table 8-262.

Table 8-262—Mesh Peering Open frame Action field format

Order	Information	Notes
1	Category	
2	Self-protected Action	
3	Capability	
4	Supported Rates	
5	Extended Supported Rates	The Extended Supported Rates element is present if there are more than eight supported rates, and is optionally present otherwise.
6	Power Capability	The Power Capability element is present if dot11SpectrumManagementRequired is true.
7	Supported Channels	The Supported Channels element is present if dot11SpectrumManagementRequired is true and dot11ExtendedChannelSwitchActivated is false.
8	RSN	The RSNE is present only if dot11MeshSecurityActivated is true.
9	Mesh ID	The Mesh ID element is set as described in 8.4.2.101.
10	Mesh Configuration	The Mesh Configuration element is set as described in 8.4.2.100.
11	Mesh Peering Management	The Mesh Peering Management element is set as described in 8.4.2.104.
12	ERP Information	The ERP element is present if ERP mesh STA detects NonERP STAs in its vicinity, and is optionally present otherwise.
13	Supported Operating Classes	The Supported Operating Classes element is present if dot11ExtendedChannelSwitchActivated is true.
14	HT Capabilities	The HT Capabilities element is present when dot11HighThroughputOptionImplemented is true.
15	HT Operation	The HT Operation element is included when dot11HighThroughputOptionImplemented is true.
16	20/40 BSS Coexistence element	The 20/40 BSS Coexistence element is optionally present when the dot112040BSSCoexistenceManagementSupport is true.
17	Extended Capabilities element	The Extended Capabilities element is optionally present if any of the fields in this element are nonzero.
18	Interworking	The Interworking element is present if dot11InterworkingServiceActivated is true.
Last–2	Vendor Specific	One or more vendor-specific elements are optionally present. These elements follow all other elements except MIC element and Authenticated Mesh Peering Exchange element.

Table 8-262—Mesh Peering Open frame Action field format *(continued)*

Order	Information	Notes
Last–1	MIC element	MIC element is present when dot11MeshSecurityActivated is true and a PMK exists between the sender and recipient of this frame.
Last	Authenticated Mesh Peering Exchange	The Authenticated Mesh Peering Exchange element is present when dot11MeshSecurityActivated is true and a PMK exists between the sender and recipient of this frame.

The Category field is set to the value in Table 8-38 for category Self-protected.

The Self-protected Action field is set to the value in Table 8-261 representing Mesh Peering Open.

The MIC element appears prior to the Authenticated Mesh Peering Exchange element in the Mesh Peering Open frame. The information following the MIC element through to the end of the Mesh Peering Open frame body is encrypted and authenticated (see 13.5).

8.5.16.3 Mesh Peering Confirm frame format

8.5.16.3.1 Mesh Peering Confirm frame self protection

Protection of this frame is provided when authenticated mesh peering exchange (AMPE) is enabled. AMPE provides integrity protection of Mesh Peering Confirm frames.

When the Mesh Peering Confirm frame is used by the mesh peering management (MPM) protocol, integrity protection on the frame is not enabled.

8.5.16.3.2 Mesh Peering Confirm frame details

The Mesh Peering Confirm frame is used to confirm a mesh peering using the procedures defined in 13.3.7 and 13.5.5. The Mesh Peering Confirm frame is also, together with Mesh Peering Open and Mesh Peering Close frames, referred to as a Mesh Peering Management frame. The format of the Mesh Peering Confirm frame Action field is shown in Table 8-263.

Table 8-263—Mesh Peering Confirm frame Action field format

Order	Information	Notes
1	Category	
2	Self-protected Action	
3	Capability	
4	AID	
5	Supported Rates	
6	Extended Supported Rates	The Extended Supported Rates element is present if there are more than eight supported rates, and is optionally present otherwise.
7	RSN	The RSNE is present only when dot11MeshSecurityActivated is true.
8	Mesh ID	The Mesh ID element is set as described in 8.4.2.101.
9	Mesh Configuration	The Mesh Configuration element is set as described in 8.4.2.100.

Table 8-263—Mesh Peering Confirm frame Action field format *(continued)*

Order	Information	Notes
10	Mesh Peering Management	The Mesh Peering Management element is set as described in 8.4.2.104.
11	HT Capabilities	The HT Capabilities element is present when dot11HighThroughputOptionImplemented is true.
12	HT Operation	The HT Operation element is included when dot11HighThroughputOptionImplemented is true.
13	20/40 BSS Coexistence element	The 20/40 BSS Coexistence element is optionally present when the dot112040BSSCoexistenceManagementSupport is true.
14	Extended Capabilities element	The Extended Capabilities element is optionally present if any of the fields in this element are nonzero.
Last–2	Vendor Specific	One or more vendor-specific elements are optionally present. These elements follow all other elements except MIC element and Authenticated Mesh Peering Exchange element.
Last–1	MIC element	MIC element is present when dot11MeshSecurityActivated is true and a PMK exists between the sender and recipient of this frame.
Last	Authenticated Mesh Peering Exchange	The Authenticated Mesh Peering Exchange element is present when dot11MeshSecurityActivated is true and a PMK exists between the sender and recipient of this frame.

The Category field is set to the value in Table 8-38 for category Self-protected.

The Self-protected Action field is set to the value in Table 8-261 representing Mesh Peering Confirm.

The MIC element appears prior to the Authenticated Mesh Peering Exchange element in the Mesh Peering Open frame. The information following the MIC element through to the end of the Mesh Peering Confirm frame body is encrypted and authenticated (see 13.5).

8.5.16.4 Mesh Peering Close frame format

8.5.16.4.1 Mesh Peering Close frame self protection

Protection of this frame is provided when authenticated mesh peering exchange (AMPE) is enabled. AMPE provides integrity protection of Mesh Peering Close frames.

When the Mesh Peering Close frame is used by the mesh peering management (MPM) protocol, integrity protection on the frame is not enabled.

8.5.16.4.2 Mesh Peering Close frame details

The Mesh Peering Close frame is used to close a mesh peering using the procedures defined in 13.3.8 and in 13.5.5. The Mesh Peering Close frame is also, together with Mesh Peering Open and Mesh Peering Confirm frames, referred to as a Mesh Peering Management frame. The format of the Mesh Peering Close frame Action field is shown in Table 8-264.

Table 8-264—Mesh Peering Close frame Action field format

Order	Information	Notes
1	Category	
2	Self-protected Action	
3	Mesh ID	The Mesh ID element is set as described in 8.4.2.101.
4	Mesh Peering Management	The Mesh Peering Management element is set as described in 8.4.2.104.
Last–2	Vendor Specific	One or more vendor-specific elements are optionally present. These elements follow all other elements except MIC element and Authenticated Mesh Peering Exchange element.
Last–1	MIC element	MIC element is present when dot11MeshSecurityActivated is true and a PMK exists between the sender and recipient of this frame.
Last	Authenticated Mesh Peering Exchange	The Authenticated Mesh Peering Exchange element is present when dot11MeshSecurityActivated is true and a PMK exists between the sender and recipient of this frame.

The Category field is set to the value in Table 8-38 for category Self-protected.

The Self-protected Action field is set to the value in Table 8-261 representing Mesh Peering Close.

The MIC element appears prior to the Authenticated Mesh Peering Exchange element in the Mesh Peering Open frame. The information following the MIC element through to the end of the Mesh Peering Close frame body is encrypted and authenticated (see 13.5).

8.5.16.5 Mesh Group Key Inform frame format

8.5.16.5.1 Mesh Group Key Inform frame self protection

The protection of the frames is provided by the mesh group key handshake protocol (see 13.6) that uses Mesh Group Key Inform frames.

8.5.16.5.2 Mesh Group Key Inform frame details

The Mesh Group Key Inform frame is used to update a mesh GTK (MGTK) with a peer. The format of the Mesh Group Key Inform frame Action field is shown in Table 8-265.

Table 8-265—Mesh Group Key Inform frame Action field format

Order	Information	Notes
1	Category	
2	Self-protected Action	
3	MIC element	
4	Authenticated Mesh Peering Exchange	

The Category field is set to the value in Table 8-38 for category Self-protected.

The Self-protected Action field is set to the value in Table 8-261 representing Mesh Group Key Inform.

The MIC element is set as defined in 8.4.2.121.

The Authenticated Mesh Peering Exchange element is set according to 13.6. The information following the MIC element through to the end of the Mesh Group Key Inform frame body is encrypted and authenticated (see 13.6.2).

8.5.16.6 Mesh Group Key Acknowledge frame format

8.5.16.6.1 Mesh Group Key Acknowledge frame self protection

The protection of the frames is provided by the mesh group key handshake protocol (see 13.6) that uses Mesh Group Key Acknowledge frames.

8.5.16.6.2 Mesh Group Key Acknowledge frame details

The Mesh Group Key Acknowledge frame is used to acknowledge receipt and processing of a Mesh Group Key Inform frame. The format of the Mesh Group Key Acknowledge frame Action field is shown in Table 8-266.

Table 8-266—Mesh Group Key Acknowledge frame Action field format

Order	Information	Notes
1	Category	
2	Self-protected Action	
3	MIC element	
4	Authenticated Mesh Peering Exchange	

The Category field is set to the value in Table 8-38 for category Self-protected.

The Self-protected Action field is set to the value in Table 8-261 representing Mesh Group Key Acknowledge.

The MIC element is set as defined in 8.4.2.121.

The Authenticated Mesh Peering Exchange element is set according to 13.6. The information following the MIC element through to the end of the Mesh Group Key Acknowledge frame body is encrypted and authenticated (see 13.6.2).

8.5.17 Mesh Action frame details

8.5.17.1 Mesh Action fields

Several Mesh Action frame formats are defined for mesh BSS operation. A Mesh Action field, in the octet field immediately after the Category field, differentiates the formats. The Mesh Action field values associated with each frame format are defined in Table 8-267.

Table 8-267—Mesh Action field values

Mesh Action field value	Description
0	Mesh Link Metric Report
1	HWMP Mesh Path Selection
2	Gate Announcement
3	Congestion Control Notification
4	MCCA Setup Request
5	MCCA Setup Reply
6	MCCA Advertisement Request
7	MCCA Advertisement
8	MCCA Teardown
9	TBTT Adjustment Request
10	TBTT Adjustment Response
11–255	Reserved

8.5.17.2 Mesh Link Metric Report frame format

The Mesh Link Metric Report frame is transmitted by a mesh STA to a neighbor peer mesh STA to report metric information on the link between the two mesh STAs. It is also transmitted by a mesh STA to a neighbor peer mesh STA to request metric information on the link between the two mesh STAs from the recipient. This frame is transmitted using an individually addressed frame. The format of the Mesh Link Metric Report frame Action field is shown in Table 8-268.

Table 8-268—Mesh Link Metric Report frame Action field format

Order	Information	Notes
1	Category	
2	Mesh Action	
3	Mesh Link Metric Report element	

The Category field is set to the value in Table 8-38 for category Mesh Action.

The Mesh Action field is set to the value in Table 8-267 representing Mesh Link Metric Report.

The Mesh Link Metric Report element is set as described in 8.4.2.102.

8.5.17.3 HWMP Mesh Path Selection frame format

The HWMP Mesh Path Selection frame is transmitted by a mesh STA to establish, update or delete paths to other mesh STAs using the HWMP defined in 13.10. This frame is transmitted in an individually or group addressed frame depending on the contained elements and as defined in 13.10.7. The format of the HWMP Mesh Path Selection frame Action field is shown in Table 8-269.

HWMP Mesh Path Selection frame contains one or more of the elements indicated in Table 8-269.

Table 8-269—HWMP Mesh Path Selection frame Action field format

Order	Information	Notes
1	Category	
2	Mesh Action	
3	PREQ element	A PREQ element is optionally present
4	PREP element	A PREP element is optionally present
5	PERR element	A PERR element is optionally present
6	RANN element	A RANN element is optionally present

The Category field is set to the value in Table 8-38 for category Mesh Action.

The Mesh Action field is set to the value in Table 8-267 representing HWMP Mesh Path Selection.

The PREQ element is set as described in 8.4.2.115.

The PREP element is set as described in 8.4.2.116.

The PERR element is set as described in 8.4.2.117.

The RANN element is set as described in 8.4.2.114.

8.5.17.4 Gate Announcement frame format

The Gate Announcement frame is transmitted by a mesh gate to announce its presence in the MBSS. This frame is transmitted using group addresses. The format of the Gate Announcement frame Action field is shown in Table 8-270.

Table 8-270—Gate Announcement frame Action field format

Order	Information	Notes
1	Category	
2	Mesh Action	
3	GANN element	

The Category field is set to the value in Table 8-38 for category Mesh Action.

The Mesh Action field is set to the value in Table 8-267 representing Gate Announcement.

The GANN element is set as described in 8.4.2.113.

8.5.17.5 Congestion Control Notification frame format

A mesh STA uses the Congestion Control Notification frame to indicate its congestion status to its neighbor peer mesh STA(s). This frame is transmitted using individual addresses or group addresses. The format of the Congestion Control Notification frame Action field is shown in Table 8-271.

Table 8-271—Congestion Control Notification frame Action field format

Order	Information	Notes
1	Category	
2	Mesh Action	
3 to (*N*+2)	Congestion Notification element	One or more Congestion Notification elements (8.4.2.103) are present. Repeated *N* times (*N* is the number of Congestion Notification elements contained in the frame).

The Category field is set to the value in Table 8-38 for category Mesh Action.

The Mesh Action field is set to the value in Table 8-267 representing Congestion Control Notification.

The Congestion Notification element is set as described in 8.4.2.103.

8.5.17.6 MCCA Setup Request frame format

The MCCA Setup Request frame is used to set up an MCCAOP reservation. It is transmitted by a mesh STA with dot11MCCAActivated equal to true to one or more neighbor peer mesh STA with dot11MCCAActivated equal to true. This frame is transmitted using individual addresses or group addresses. The format of the MCCA Setup Request frame Action field is shown in Table 8-272.

Table 8-272—MCCA Setup Request frame Action field format

Order	Information	Notes
1	Category	
2	Mesh Action	
3	MCCAOP Setup Request element	

The Category field is set to the value in Table 8-38 for category Mesh Action.

The Mesh Action field is set to the value in Table 8-267 representing MCCA Setup Request.

The MCCAOP Setup Request element is described in 8.4.2.108.

8.5.17.7 MCCA Setup Reply frame format

The MCCA Setup Reply frame is used to reply to an MCCA Setup Request frame. It is transmitted by a mesh STA with dot11MCCAActivated equal to true to a neighbor peer mesh STA with dot11MCCAActivated equal to true. This frame is transmitted using individual addresses. The format of the MCCA Setup Reply frame Action field is shown in Table 8-273.

Table 8-273—MCCA Setup Reply frame Action field format

Order	Information	Notes
1	Category	
2	Mesh Action	
3	MCCAOP Setup Reply element	

The Category field is set to the value in Table 8-38 for category Mesh Action.

The Mesh Action field is set to the value in Table 8-267 representing MCCA Setup Reply.

The MCCAOP Setup Reply element is described in 8.4.2.109.

8.5.17.8 MCCA Advertisement Request frame format

The MCCA Advertisement Request frame is transmitted by a mesh STA with dot11MCCAActivated equal to true to a neighbor peer mesh STA with dot11MCCAActivated equal to true in order to request MCCAOP advertisements from the neighbor peer mesh STA. This frame is transmitted using individual addresses. The format of the MCCA Advertisement Request frame Action field is shown in Table 8-274.

Table 8-274—MCCA Advertisement Request frame Action field format

Order	Information	Notes
1	Category	
2	Mesh Action	
3	MCCAOP Advertisement Overview element	An MCCAOP Advertisement Overview element is optionally present.

The Category field is set to the value in Table 8-38 for category Mesh Action.

The Mesh Action field is set to the value in Table 8-267 representing MCCA Advertisement Request.

The MCCAOP Advertisement Overview element is described in 8.4.2.110.

8.5.17.9 MCCA Advertisement frame format

The MCCA Advertisement frame is transmitted by a mesh STA with dot11MCCAActivated equal to true to one or more neighbor peer mesh STAs with dot11MCCAActivated equal to true. This frame is transmitted using group addresses or individual addresses. The format of the MCCA Advertisement frame Action field is shown in Table 8-275.

Table 8-275—MCCA Advertisement frame Action field format

Order	Information	Notes
1	Category	
2	Mesh Action	
3	MCCAOP Advertisement Overview element	An MCCAOP Advertisement Overview element is optionally present.
4	MCCAOP Advertisement elements	If an MCCAOP Advertisement Overview element is present, zero or more MCCAOP Advertisement elements are present. If an MCCAOP Advertisement Overview element is not present, one or more MCCAOP Advertisement elements are present.

The Category field is set to the value in Table 8-38 for category Mesh Action.

The Mesh Action field is set to the value in Table 8-267 representing MCCA Advertisement.

The MCCAOP Advertisement Overview element is described in 8.4.2.110.

The MCCAOP Advertisement element is described in 8.4.2.111.

8.5.17.10 MCCA Teardown frame format

The MCCA Teardown frame is transmitted by a mesh STA with dot11MCCAActivated equal to true to one or more neighbor peer mesh STAs with dot11MCCAActivated equal to true. This frame is transmitted using group addresses or individual addresses. The format of the MCCA Teardown frame Action field is shown in Table 8-276.

Table 8-276—MCCA Teardown frame Action field format

Order	Information	Notes
1	Category	
2	Mesh Action	
3	MCCAOP Teardown element	

The Category field is set to the value in Table 8-38 for category Mesh Action.

The Mesh Action field is set to the value in Table 8-267 representing MCCA Teardown.

The MCCAOP Teardown element is described in 8.4.2.112.

8.5.17.11 TBTT Adjustment Request frame format

The TBTT Adjustment Request frame is used to request a particular neighbor peer mesh STA to adjust its TBTT. This frame is transmitted using individual addresses. The format of the TBTT Adjustment Request frame Action field is shown in Table 8-277.

Table 8-277—TBTT Adjustment Request frame Action field format

Order	Information	Notes
1	Category	
2	Mesh Action	
3 to (*N_Info*+2)	Beacon Timing element	Repeated *N_Info* times (*N_Info* is a number of beacon timing information tuples as described in 13.13.4.2.5).

The Category field is set to the value in Table 8-38 for category Mesh Action.

The Mesh Action field is set to the value in Table 8-267 representing TBTT Adjustment Request.

The Beacon Timing element is set as described in 8.4.2.107. When not all beacon timing information is included in a Beacon Timing element due to the maximum element size limit, multiple Beacon Timing elements are present. The elements are present in the order of Beacon Timing Element Number field value in the Report Control field of the Beacon Timing element.

8.5.17.12 TBTT Adjustment Response frame format

The TBTT Adjustment Response frame is used to respond to a TBTT adjustment request. This frame is transmitted using individual addresses. The format of the TBTT Adjustment Response frame Action field is shown in Table 8-278.

Table 8-278—TBTT Adjustment Response frame Action field format

Order	Information	Notes
1	Category	
2	Mesh Action	
3	Status Code	
4 to (*N_Info*+3)	Beacon Timing element (optional)	Repeated *N_Info* times (*N_Info* is a number of beacon timing information tuples as described in 13.13.4.2.5).

The Category field is set to the value in Table 8-38 for category Mesh Action.

The Mesh Action field is set to the value in Table 8-267 representing TBTT Adjustment Response.

The Status Code field is set as described in 13.13.4.4.2.

The Beacon Timing element is set as defined in 8.4.2.107. It is present only if the Status Code is set to 78 (i.e., when the request is not successful due to the neighbor constraint). When not all beacon timing information is included in a Beacon Timing element due to the maximum element size limit, multiple Beacon Timing elements are present. The elements are present in the order of Beacon Timing Element Number field value in the Report Control field of the Beacon Timing element.

8.5.18 Multihop Action frame details

8.5.18.1 Multihop Action fields

Several Multihop Action frame formats are defined for mesh BSS operation. A Multihop Action field, in the octet field immediately after the Category field, differentiates the formats. The Multihop Action field values associated with each frame format are defined in Table 8-279. The Mesh Control field is present immediately after the Multihop Action field in all Multihop Action frames.

Table 8-279—Multihop Action field values

Multihop Action field value	Description
0	Proxy Update
1	Proxy Update Confirmation
2–255	Reserved

8.5.18.2 Proxy Update frame format

The Proxy Update frame is used to inform the recipient about new, updated, or deleted proxy information. This frame is transmitted using individual addresses. The format of the Proxy Update frame Action field is shown in Table 8-280.

Table 8-280—Proxy Update frame Action field format

Order	Information	Notes
1	Category	
2	Multihop Action	
3	Mesh Control	
4 to (*N*+3)	PXU element	Repeated *N* times (*N* is the number of PXU elements contained in the frame).

The Category field is 1 octet and is set to the value in Table 8-38 for category Multihop Action.

The Multihop Action field is set to the value in Table 8-279 representing Proxy Update.

The Mesh Control field is set as defined in 8.2.4.7.3.

The PXU element is described in 8.4.2.118. The Proxy Update frame allows the inclusion of multiple PXU elements.

8.5.18.3 Proxy Update Confirmation frame format

The Proxy Update Confirmation frame is transmitted by a mesh STA in response to a Proxy Update frame. This frame is used to inform that the corresponding PXU element has been properly received, and is transmitted using individual addresses. The format of the Proxy Update Confirmation frame Action field is shown in Table 8-281.

Table 8-281—Proxy Update Confirmation frame Action field format

Order	Information	Notes
1	Category	
2	Multihop Action	
3	Mesh Control	
4 to (*N*+3)	PXUC element	Repeated *N* times (*N* is the number of PXUC elements contained in the frame).

The Category field is 1 octet and is set to the value in Table 8-38 for category Multihop Action.

The Multihop Action field is set to the value in Table 8-279 representing Proxy Update Confirmation.

The Mesh Control field is set as defined in 8.2.4.7.3.

The PXUC element is described in 8.4.2.119. The Proxy Update Confirmation frame allows the inclusion of multiple PXUC elements.

8.6 Aggregate MPDU (A-MPDU)

8.6.1 A-MPDU format

An A-MPDU consists of a sequence of one or more A-MPDU subframes as shown in Figure 8-503.

	A-MPDU subframe 1	A-MPDU subframe 2	…	A-MPDU subframe n
Octets:	variable	variable		variable

Figure 8-503—A-MPDU format

The structure of the A-MPDU subframe is shown in Figure 8-504. Each A-MPDU subframe consists of an MPDU delimiter followed by an MPDU. Except when an A-MPDU subframe is the last one in an A-MPDU, padding octets are appended to make each A-MPDU subframe a multiple of 4 octets in length. The A-MPDU maximum length is 65 535 octets. The length of an A-MPDU addressed to a particular STA may be further constrained as described in 9.12.2.

	MPDU delimiter	MPDU	Pad
Octets:	4	Variable	0–3

Figure 8-504—A-MPDU subframe format

The MPDU delimiter is 4 octets in length. The structure of the MPDU delimiter is defined in Figure 8-505.

	B0 B3	B4 B15	B16 B23	B24 B31
	Reserved	MPDU length	CRC	Delimiter Signature
Bits:	4	12	8	8

Figure 8-505—MPDU delimiter

The fields of the MPDU delimiter are defined in Table 8-282.

Table 8-282— MPDU delimiter fields

Field	Size (bits)	Description
Reserved	4	
MPDU length	12	Length of the MPDU in octets
CRC	8	8-bit CRC of the preceding 16-bits.
Delimiter Signature	8	Pattern that may be used to detect an MPDU delimiter when scanning for a delimiter. The unique pattern is set to the value 0x4E. NOTE—As the Delimiter Signature field was created by the IEEE 802.11 Task Group n, it chose the ASCII value for the character ‘N’ as the unique pattern.

The purpose of the MPDU delimiter is to locate the MPDUs within the A-MPDU so that the structure of the A-MPDU can usually be recovered when one or more MPDU delimiters are received with errors. See S.2 for a description of a deaggregation algorithm.

A delimiter with MPDU length of 0 is valid. This value is used as defined in 9.12.3 to meet the minimum MPDU start spacing requirement.

8.6.2 MPDU delimiter CRC field

The MPDU delimiter CRC field is an 8-bit CRC value. It is used as a frame check sequence (FCS) to protect the Reserved and MPDU Length fields. The CRC field is the ones complement of the remainder generated by the modulo 2 division of the protected bits by the polynomial $x^8 + x^2 + x^1 + 1$, where the shift-register state is preset to all ones.

NOTE—The order of transmission of bits within the CRC field is defined in 8.2.2.

Figure 8-506 illustrates the CRC calculation for the MPDU delimiter.

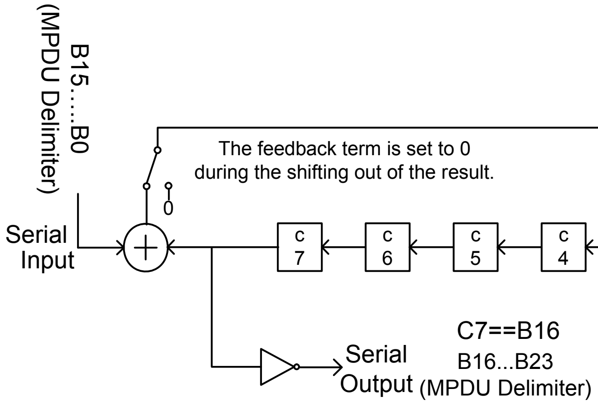


Figure 8-506—MPDU delimiter CRC calculation

8.6.3 A-MPDU contents

An A-MPDU is a sequence of MPDUs carried in a single PPDU with the TXVECTOR/RXVECTOR AGGREGATION parameter set to 1.

All the MPDUs within an A-MPDU are addressed to the same RA. All QoS data frames within an A-MPDU that have a TID for which an HT-immediate Block Ack agreement exists have the same value for the Ack Policy subfield of the QoS Control field.

All protected MPDUs within an A-MPDU have the same Key ID.

The Duration/ID fields in the MAC headers of all MPDUs in an A-MPDU carry the same value.

An A-MPDU is transmitted in one of the contexts specified in Table 8-283. Ordering of MPDUs within an A-MPDU is not constrained, except where noted in these tables. See 9.12.1.

NOTE 1—The TIDs present in a data enabled A-MPDU context are also constrained by the channel access rules (for a TXOP holder; see 9.19.2 and 9.19.3) and the RD response rules (for an RD responder, see 9.25.4). This is not shown in these tables.

NOTE 2—MPDUs carried in an A-MPDU are limited to a maximum length of 4095 octets. If a STA supports A-MSDUs of 7935 octets (indicated by the Maximum A-MSDU Length field in the HT Capabilities element), A-MSDUs transmitted by that STA within an A-MPDU are constrained so that the length of the QoS data MPDU carrying the A-MSDU is no more than 4095 octets. The use of A-MSDU within A-MPDU might be further constrained as described in 8.4.1.14 through the operation of the A-MSDU Supported field.

Table 8-283—A-MPDU Contexts

Name of Context	Definition of Context	Table defining permitted contents
Data Enabled Immediate Response	The A-MPDU is transmitted outside a PSMP sequence by a TXOP holder or an RD responder including potential immediate responses.	Table 8-284
Data Enabled No Immediate Response	The A-MPDU is transmitted outside a PSMP sequence by a TXOP holder that does not include or solicit an immediate response. See NOTE.	Table 8-285
PSMP	The A-MPDU is transmitted within a PSMP sequence.	Table 8-286
Control Response	The A-MPDU is transmitted by a STA that is neither a TXOP holder nor an RD responder that also needs to transmit one of the following immediate response frames: Ack BlockAck with a TID for which an HT-immediate Block Ack agreement exists	Table 8-287
NOTE—This context includes cases when no response is generated or when a response is generated later by the operation of the delayed Block Ack rules.		

Table 8-284—A-MPDU contents in the data enabled immediate response context

MPDU Description	Conditions	
ACK MPDU	If the preceding PPDU contains an MPDU that requires an ACK response, a single ACK MPDU at the start of the A-MPDU.	At most one of these MPDUs is present.
HT-immediate BlockAck	If the preceding PPDU contains an implicit or explicit Block Ack request for a TID for which an HT-immediate Block Ack agreement exists, at most one BlockAck for this TID, in which case it occurs at the start of the A-MPDU.	
Delayed BlockAcks	BlockAck frames with the BA Ack Policy subfield equal to No Acknowledgment with a TID for which an HT-delayed Block Ack agreement exists.	
Delayed Block Ack data	QoS Data MPDUs with a TID that corresponds to a Delayed or HT-delayed Block Ack agreement. These have the Ack Policy field equal to Block Ack.	
Action No Ack	Management frames of subtype Action No Ack.	
Delayed BlockAckReqs	BlockAckReq MPDUs with a TID that corresponds to an HT-delayed Block Ack agreement in which the BA Ack Policy subfield is equal to No Acknowledgment.	

Table 8-284—A-MPDU contents in the data enabled immediate response context *(continued)*

MPDU Description	Conditions	
Data MPDUs sent under an HT-immediate Block Ack agreement	QoS Data MPDUs with the same TID, which corresponds to an HT-immediate Block Ack agreement. These MPDUs all have the Ack Policy field equal to the same value, which is either Implicit Block Ack Request or Block Ack.	Of these, at most one of the following is present: One or more QoS Data MPDUs with the Ack Policy field equal to Implicit Block Ack Request BlockAckReq
Immediate BlockAckReq	At most one BlockAckReq frame with a TID that corresponds to an HT-immediate Block Ack agreement. This is the last MPDU in the A-MPDU. It is not present if any QoS data frames for that TID are present.	

Table 8-285—A-MPDU contents in the data enabled no immediate response context

MPDU Description	Conditions
Delayed BlockAcks	BlockAck frames for a TID for which an HT-delayed Block Ack agreement exists with the BA Ack Policy subfield equal to No Acknowledgment.
Delayed Block Ack data	QoS Data MPDUs with a TID that corresponds to a Delayed or HT-delayed Block Ack agreement. These have the Ack Policy field equal to Block Ack.
Data without a Block Ack agreement	QoS Data MPDUs with a TID that does not correspond to a Block Ack agreement. These have the Ack Policy field equal to No Ack and the A-MSDU Present subfield equal to 0.
Action No Ack	Management frames of subtype Action No Ack.
Delayed BlockAckReqs	BlockAckReq MPDUs with the BA Ack Policy subfield equal to No Acknowledgment and with a TID that corresponds to an HT-delayed Block Ack agreement.

Table 8-286—A-MPDU contents in the PSMP context

MPDU Description	Conditions	
Acknowledgment for PSMP data	At most one Multi-TID BlockAck MPDU. Acknowledgment in response to data received with the Ack Policy field equal to PSMP Ack and/or a Multi-TID BlockAckReq MPDU in the previous PSMP-UTT or PSMP-DTT.	
Delayed BlockAcks	BlockAck frames with the BA Ack Policy subfield equal to No Acknowledgment and with a TID for which an HT-delayed Block Ack agreement exists.	
HT-immediate Data	QoS Data MPDUs in which the Ack Policy field is equal to PSMP Ack or Block Ack and with a TID that corresponds to an HT-immediate Block Ack agreement.	An A-MPDU containing MPDUs with a Block Ack agreement does not also contain MPDUs without a Block Ack agreement.
Delayed Block Ack data	QoS Data MPDUs with a TID that corresponds to a Delayed or HT-delayed Block Ack agreement. These have the Ack Policy field equal to Block Ack.	
Data without a Block Ack agreement	QoS Data MPDUs with a TID that does not correspond to a Block Ack agreement. These have the Ack Policy field equal to No Ack and the A-MSDU Present subfield is equal to 0.	
Action No Ack	Management frames of subtype Action No Ack.	
Delayed BlockAckReqs	BlockAckReq MPDUs with the BA Ack Policy subfield equal to No Acknowledgment and with a TID that corresponds to an HT-delayed Block Ack.	
Multi-TID BlockAckReq	At most one Multi-TID BlockAckReq MPDU with the BA Ack Policy subfield equal to No Ack.	

Table 8-287—A-MPDU contents MPDUs in the control response context

MPDU	Conditions	
ACK	ACK transmitted in response to an MPDU that requires an ACK.	Only one of these is present at the start of the A-MPDU.
BlockAck	BlockAck with a TID that corresponds to an HT-immediate Block Ack agreement.	
Action No Ack	Management frames of subtype Action No Ack +HTC carrying a Management Action Body containing an explicit feedback response.	

9. MAC sublayer functional description

9.1 Introduction

The MAC functional description is presented in this clause. The architecture of the MAC sublayer, including the distributed coordination function (DCF), the point coordination function (PCF), the hybrid coordination function (HCF), the mesh coordination function (MCF), and their coexistence in an IEEE 802.11 LAN are introduced in 9.2. These functions are expanded on in 9.3 (DCF), 9.4 (PCF), 9.19 (HCF), and 9.20 (MCF). Fragmentation and defragmentation are defined in 9.5 and 9.6. Multirate support is addressed in 9.7. A number of additional restrictions to limit the cases in which MSDUs are reordered or discarded are described in 9.8. Operation across regulatory domains is defined in 9.18. The Block Ack mechanism is described in 9.21. The No Ack mechanism is described in 9.22. The protection mechanism is described in 9.23. Rules for processing MAC frames are described in 9.24.

9.2 MAC architecture

9.2.1 General

A representation of the MAC architecture is shown in Figure 9-1 in which the PCF and HCF services are provided using the services of the DCF. Note that in a non-QoS STA, HCF is not present. In a QoS STA implementation, both DCF and HCF are present. PCF is optional in all STAs.

Due to the distributed nature of the MBSS, only the MCF is present in a mesh STA.

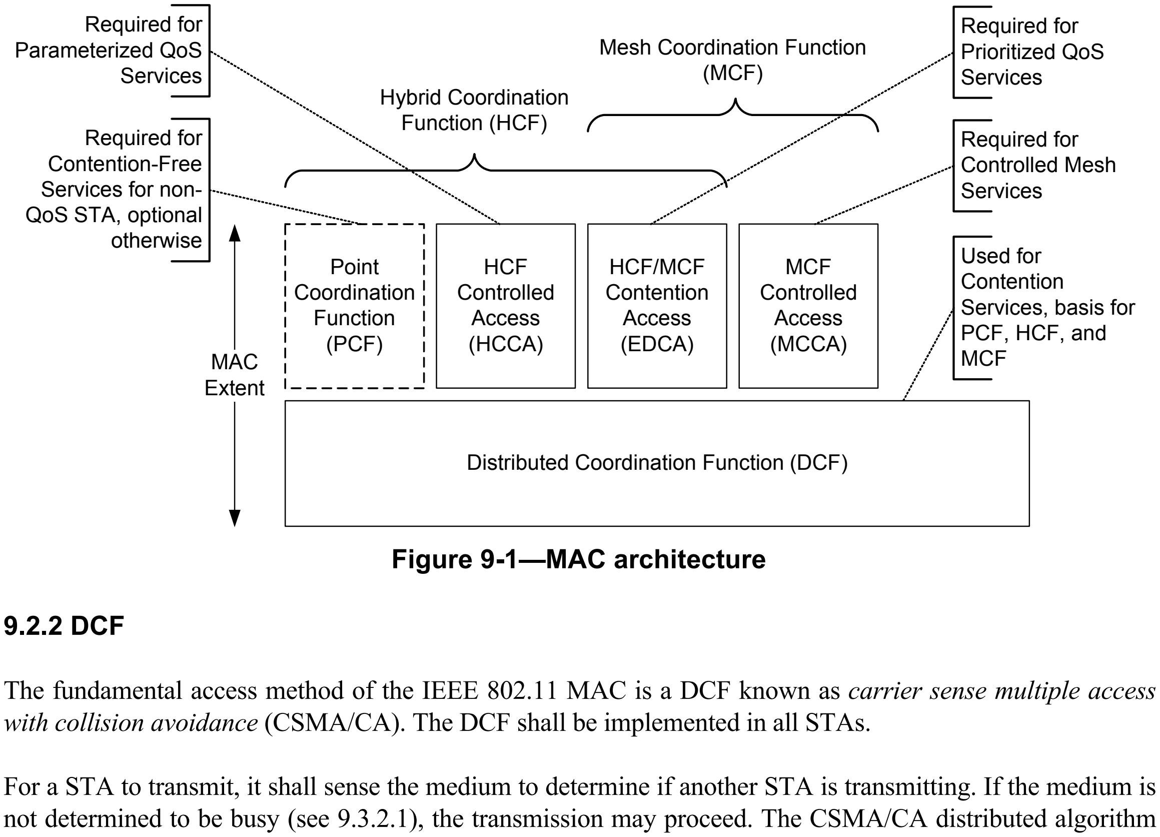

Required for Parameterized QoS Services
Required for Contention-Free Services for non-QoS STA, optional otherwise
Hybrid Coordination Function (HCF)
Mesh Coordination Function (MCF)
Required for Prioritized QoS Services
Required for Controlled Mesh Services
Used for Contention Services, basis for PCF, HCF, and MCF
MAC Extent
Point Coordination Function (PCF)
HCF Controlled Access (HCCA)
HCF/MCF Contention Access (EDCA)
MCF Controlled Access (MCCA)
Distributed Coordination Function (DCF)

Figure 9-1—MAC architecture

9.2.2 DCF

The fundamental access method of the IEEE 802.11 MAC is a DCF known as *carrier sense multiple access with collision avoidance* (CSMA/CA). The DCF shall be implemented in all STAs.

For a STA to transmit, it shall sense the medium to determine if another STA is transmitting. If the medium is not determined to be busy (see 9.3.2.1), the transmission may proceed. The CSMA/CA distributed algorithm mandates that a gap of a minimum specified duration exists between contiguous frame sequences. A transmitting STA shall verify that the medium is idle for this required duration before attempting to transmit. If the medium is determined to be busy, the STA shall defer until the end of the current transmission. After deferral, or prior to attempting to transmit again immediately after a successful transmission, the STA shall select a random backoff interval and shall decrement the backoff interval counter while the medium is idle. A transmission is successful either when an ACK frame is received from the STA addressed by the RA field of the transmitted frame or when a frame with a group address in the RA field is transmitted completely. A refinement of the method may be used under various circumstances to further minimize collisions—here the transmitting and receiving STA exchange short control frames (RTS and CTS frames) after determining that the medium is idle and after any deferrals or backoffs, prior to data transmission. The details of CSMA/CA, deferrals, and backoffs are described in 9.3. RTS/CTS exchanges are also presented in 9.3.

9.2.3 PCF

The IEEE 802.11 MAC may also incorporate an optional access method called a PCF, which is only usable on infrastructure network configurations. This access method uses a PC, which shall operate at the AP of the BSS, to determine which STA currently has the right to transmit. The operation is essentially that of polling, with the PC performing the role of the polling master. The operation of the PCF may require additional coordination, not specified in this standard, to permit efficient operation in cases where multiple point-coordinated BSSs are operating on the same channel, in overlapping physical space.

The PCF uses a virtual carrier sense (CS) mechanism aided by an access priority mechanism. The PCF shall distribute information within Beacon management frames to gain control of the medium by setting the NAV in STAs. In addition, all frame transmissions under the PCF may use an interframe space (IFS) that is smaller than the IFS for frames transmitted via the DCF. The use of a smaller IFS implies that point-coordinated traffic has priority access to the medium over STAs in overlapping BSSs operating under the DCF access method.

The access priority provided by a PCF may be utilized to create a CF access method. The PC controls the frame transmissions of the STAs so as to eliminate contention for a limited period of time.

9.2.4 Hybrid coordination function (HCF)

9.2.4.1 General

The QoS facility includes an additional coordination function called *HCF* that is only usable in QoS network configurations. The HCF shall be implemented in all QoS STAs except mesh STAs. Instead, mesh STAs implement the MCF. The HCF combines functions from the DCF and PCF with some enhanced, QoS-specific mechanisms and frame subtypes to allow a uniform set of frame exchange sequences to be used for QoS data transfers during both the CP and CFP. The HCF uses both a contention-based channel access method, called the *enhanced distributed channel access* (EDCA) mechanism for contention-based transfer and a controlled channel access, referred to as the *HCF controlled channel access* (HCCA) mechanism, for contention-free transfer.

STAs may obtain TXOPs using one or both of the channel access mechanisms specified in 9.19. If a TXOP is obtained using the contention-based channel access, it is defined as *EDCA TXOP*. If a TXOP is obtained using the controlled channel access, it is defined as *HCCA TXOP*. If an HCCA TXOP is obtained due to a QoS (+)CF-Poll frame from the HC, the TXOP is defined as a *polled TXOP*.

Time priority management frames are transmitted outside of the normal MAC queuing process as per individually described transmission rules. Frames listed in Table 8-229 with a value of "Yes" in the "Time Priority" column are time priority management frames. No other frames are time priority management frames.

9.2.4.2 HCF contention-based channel access (EDCA)

The EDCA mechanism provides differentiated, distributed access to the WM for STAs using eight different UPs. The EDCA mechanism defines four access categories (ACs) that provide support for the delivery of traffic with UPs at the STAs. The AC is derived from the UPs as shown in Table 9-1.

Table 9-1—UP-to-AC mappings

Priority	UP (Same as 802.1D user priority)	802.1D designation	AC	Designation (informative)
Lowest	1	BK	AC_BK	Background
	2	—	AC_BK	Background
	0	BE	AC_BE	Best Effort
	3	EE	AC_BE	Best Effort
	4	CL	AC_VI	Video
	5	VI	AC_VI	Video
	6	VO	AC_VO	Voice
Highest	7	NC	AC_VO	Voice

For each AC, an enhanced variant of the DCF, called an *enhanced distributed channel access function* (EDCAF), contends for TXOPs using a set of EDCA parameters. When communicating data frames outside the context of a BSS (dot11OCBActivated is true), the EDCA parameters are the corresponding default values or are as set by the SME in dot11EDCATable (except for TXOP limit values, which shall be set to 0 for each AC). When communicating within a BSS, the EDCA parameters used are from the EDCA Parameter Set element or from the default values for the parameters when no EDCA Parameter Set element is received from the AP of the BSS with which the STA is associated or when the STA is a mesh STA. The parameters used by the EDCAF to control its operation are defined by dot11QAPEDCATable at the AP and by dot11EDCATable at the non-AP STA.

The following rules apply for HCF contention-based channel access:

a) The minimum specified idle duration time is not the constant value (DIFS) as defined for DCF, but is a distinct value (contained in table dot11QAPEDCATableAIFSN for an AP and in table dot11EDCATableAIFSN for a non-AP STA; see 9.19.2) assigned either by a management entity or by an AP.

b) The contention window limits aCWmin and aCWmax, from which the random backoff is computed, are not fixed per PHY, as with DCF, but are variable (contained in tables dot11QAPEDCACWmin and dot11QAPEDCACWmax for an AP and in tables dot11EDCATableCWmin and dot11EDCATableCWmax for a non-AP STA) and assigned by a management entity or by an AP.

c) Collisions between contending EDCAFs within a STA are resolved within the STA so that the data frames from the higher priority AC receive the TXOP and the data frames from the lower priority colliding AC(s) behave as if there were an external collision on the WM.

   NOTE—This collision behavior does not include setting retry bits in the MAC headers of MPDUs at the head of the lower priority ACs, as would be done after a transmission attempt that was unsuccessful due to an actual external collision on the WM.

d) During an EDCA TXOP won by an EDCAF, a STA may initiate multiple frame exchange sequences to transmit MMPDUs and/or MSDUs within the same AC. The duration of this EDCA TXOP is bounded, for an AC, by the value dot11QAPEDCATXOPLimit for an AP and by dot11EDCATableTXOPLimit for a non-AP STA. A value of 0 for this duration means that the EDCA TXOP is limited as defined by the rule for TXOP Limit value 0 found in 9.19.2.2.

The QoS AP shall announce the EDCA parameters in selected Beacon frames and in all Probe Response and (Re)Association Response frames by the inclusion of the EDCA Parameter Set element using the information from the MIB entries in dot11ECDATable. If no such element is received, the STAs shall use the default values for the parameters. The fields following the QoS Info field in the EDCA Parameter Set element shall be included in all Beacon frames occurring within two (optionally more) delivery traffic indication map (DTIM) periods following a change in AC parameters, which provides all STAs an opportunity to receive the updated EDCA parameters. A QoS STA shall update its MIB values of the EDCA parameters within an interval of time equal to one beacon interval after receiving an updated EDCA parameter set. QoS STAs update the MIB attributes and store the EDCA Parameter Set update count value in the QoS Info field. An AP may change the EDCA access parameters by changing the EDCA Parameter Set element in the Beacon frame, Probe Response frame, and (Re)Association Response frame. However, the AP should change them only rarely. A QoS STA shall use the EDCA Parameter Set Update Count Value subfield in the QoS Capability element of all Beacon frames to determine whether the STA is using the current EDCA Parameter Values. If the EDCA Parameter Set update count value in the QoS Capability element is different from the value that has been stored, the QoS STA shall query the updated EDCA parameter values by sending a Probe Request frame to the AP.

In the Beacon frame, the EDCA Parameter Set Update Count subfield is initially set by the AP to 0 and is incremented every time any of the AC parameters changes.

The AP may use a different set of EDCA parameter values than it advertises to the STAs in its BSS.

A QoS STA should send individually addressed Management frames that are addressed to a non-QoS STA using the access category AC_BE and shall send all other management frames using the access category AC_VO. A QoS STA that does not send individually addressed Management frames that are addressed to a non-QoS STA using the access category AC_BE shall send them using the access category AC_VO. Management frames are exempted from any and all restrictions on transmissions arising from admission control procedures. A QoS STA shall also send management frames using the access category AC_VO before associating with any BSS and before establishing mesh peerings in an MBSS, even if there is no QoS facility available in that BSS. BlockAckReq and BlockAck frames shall be sent using the same access category as the corresponding QoS data frames. PS-Poll frames shall be sent using the access category AC_BE (to reduce the likelihood of collision following a Beacon frame) and are exempted from any and all restrictions on transmissions arising from admission control procedures. When the first frame in a frame exchange sequence is an RTS or CTS frame, the RTS or CTS frame shall be transmitted using the access category of the corresponding QoS Data/QoS Null frame(s) or AC_VO for management frames. Control Wrapper frames shall be sent using the access category that would apply to the carried control frame.

Note—A QoS STA can choose to use AC_VO when transmitting management frames to a non-QoS STA when no prior data frames have been transmitted to the non-QoS STA.

The operation rules of HCF contention-based channel access are defined in 9.19.2.

9.2.4.3 HCF controlled channel access (HCCA)

The HCCA mechanism uses a QoS-aware centralized coordinator, called a *hybrid coordinator* (HC), and operates under rules that are different from the PC of the PCF. The HC is collocated with the AP of the BSS and uses the HC's higher priority of access to the WM to initiate frame exchange sequences and to allocate TXOPs to itself and other STAs in order to provide limited-duration CAPs for contention-free transfer of QoS data.

The HC traffic delivery and TXOP allocation may be scheduled during the CP and any locally generated CFP (generated optionally by the HC) to meet the QoS requirements of a particular TC or TS. TXOP allocations and contention-free transfers of QoS traffic might be based on the HC's BSS-wide knowledge of the amounts of pending traffic belonging to different TS and/or TCs and are subject to BSS-specific QoS policies.

An AP may transmit a CF-Poll to a non-QoS STA, thereby providing non-QoS contention-free transfers during the CFP. This provisioning of contention-free transfers during the CFP to non-QoS STAs, however, is not recommended. Implementers are cautioned that QoS STAs are not required to interpret data subtypes that include QoS +CF-Ack in frames not addressed to themselves unless they set the Q-Ack subfield in the QoS Capability element to 1. QoS STAs are also not required to interpret data subtypes that are non-QoS (+)CF-Poll frames (i.e., data frames with bits 7, 5, and 4 in the Frame Control field equal to 0, 1, and 0, respectively); therefore, QoS STAs cannot be treated as CF-Pollable STAs. This requires an AP that provides non-QoS CF-polling to adhere to frame sequence restrictions considerably more complex than, and less efficient than, those specified for either PCF or HCF. In addition, the achievable service quality is likely to be degraded when non-QoS STAs are associated and being polled.

The HCF protects the transmissions during each CAP using the virtual CS mechanism.

A STA may initiate multiple frame exchange sequences during a polled TXOP of sufficient duration to perform more than one such sequence. The use of virtual CS by the HC provides improved protection of the CFP, in addition to the protection provided by having all STAs in the BSA setting their NAVs to dot11CFPMaxDuration at the target beacon transmission time (TBTT) of DTIM Beacon frames.

The operation rules of the HCCA are defined in 9.19.3.

9.2.5 Mesh coordination function (MCF)

The mesh facility includes an additional coordination function called MCF that is usable only in an MBSS. Mesh STAs shall implement the MCF only. MCF has both a contention-based channel access and contention free channel access mechanism. The contention based mechanism is EDCA and the contention free mechanism is called the MCF controlled channel access (MCCA). MCF uses the default values for the PTKSA, GTKSA and STKSA Replay Counters. The operation rules of the EDCA are defined in 9.19.2. The operation rules of the MCCA are defined in 9.20.3.

9.2.6 Combined use of DCF, PCF, and HCF

The DCF and a centralized coordination function (either PCF or HCF) are defined so they may operate within the same BSS. When a PC is operating in a BSS, the PCF and DCF access methods alternate, with a CFP followed by a CP. This is described in greater detail in 9.4. When an HC is operating in a BSS, it may generate an alternation of CFP and CP in the same way as a PC, using the DCF access method only during the CP. The HCF access methods (controlled and contention-based) operate sequentially when the channel is in CP. Sequential operation allows the polled and contention-based access methods to alternate, within intervals as short as the time to transmit a frame exchange sequence, under rules defined in 9.19.

9.2.7 Fragmentation/defragmentation overview

The process of partitioning an MSDU or an MMPDU into smaller MAC level frames, MPDUs, is called *fragmentation*. Fragmentation creates MPDUs smaller than the original MSDU or MMPDU length to increase reliability, by increasing the probability of successful transmission (as defined in 9.2.2) of the MSDU or MMPDU in cases where channel characteristics limit reception reliability for longer frames. STAs may use fragmentation to use the medium efficiently in consideration of the duration available in granted TXOPs, as long as the rules in 9.5 are followed. Fragmentation is accomplished at each immediate transmitter. The process of recombining MPDUs into a single MSDU or MMPDU is defined as *defragmentation*. Defragmentation is accomplished at each immediate recipient.

An MSDU transmitted under HT-immediate or HT-delayed Block Ack agreement shall not be fragmented even if its length exceeds dot11FragmentationThreshold. An MSDU or MMPDU transmitted within an A-MPDU shall not be fragmented even if its length exceeds dot11FragmentationThreshold. Group addressed MSDUs or MMPDUs shall not be fragmented even if their length exceeds dot11FragmentationThreshold.

NOTE 1—A fragmented MSDU or MMPDU transmitted by an HT STA to another HT STA can be acknowledged only using immediate acknowledgment (i.e., transmission of an ACK frame after a SIFS).

NOTE 2—As specified in 9.11, A-MSDUs are never fragmented.

Except as described below, when an individually addressed MSDU received from the LLC would result in an MPDU of length greater than dot11FragmentationThreshold, the MSDU shall be fragmented. Except as described below, when an individually addressed MMPDU received from the MLME, would result in an MPDU of length greater than dot11FragmentationThreshold, the MMPDU shall be fragmented.

The exception applies when an MSDU is transmitted using an HT-immediate or HT-delayed Block Ack agreement or when the MSDU or MMPDU is carried in an A-MPDU, in which case the MSDU or MMPDU is transmitted without fragmentation. Each fragment is a frame no longer than dot11FragmentationThreshold, if security encapsulation is not invoked for the MPDU. If security encapsulation is active for the MPDU, then the fragments shall be expanded by the encapsulation overhead and this may result in a fragment larger than dot11FragmentationThreshold. It is possible that any fragment may be a frame smaller than dot11FragmentationThreshold. An illustration of fragmentation is shown in Figure 9-2.

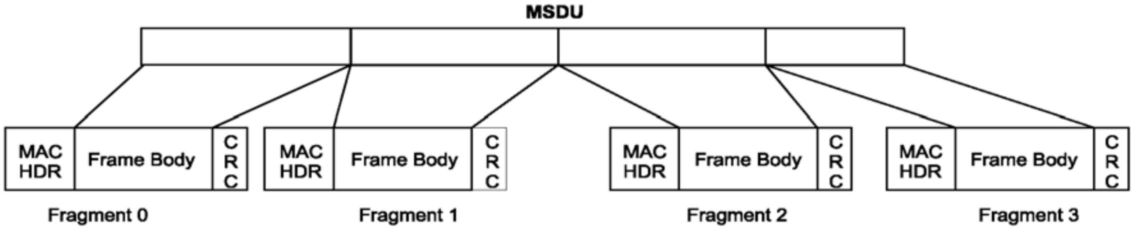


Figure 9-2—Fragmentation

The MPDUs resulting from the fragmentation of an MSDU or MMPDU are sent as independent transmissions, each of which is separately acknowledged. This permits transmission retries to occur per fragment, rather than per MSDU or MMPDU. The fragments of a single MSDU or MMPDU are either

- Sent during a CP as individual frames using the DCF, or
- Sent during a CFP as individual frames obeying the rules of the PC medium access procedure, or
- Sent as a burst in an EDCA or HCCA TXOP, subject to TXOP limits and medium occupancy limits for the attached PHY.

9.2.8 MAC data service

The MAC data service provides the transport of MSDUs between MAC peer entities as characterized in 5.1.1.

The transmission process is started by receipt of an MA-UNITDATA.request primitive containing an MSDU and the associated parameters. This might cause one or more data MPDUs containing the MSDU to be transmitted following A-MSDU aggregation, fragmentation, and security encapsulation, as appropriate.

The MA-UNITDATA.indication primitive is generated in response to one or more received data MPDUs containing an MSDU following validation, address filtering, decryption, decapsulation, defragmentation, and

A-MSDU deaggregation, as appropriate. Address filtering is performed on the Address 1 field of each MPDU contained in a PPDU and on the DA of each MSDU within an A-MSDU. When the Address 1 field or DA field contains a group address, address filtering is performed by comparing the value in the Address 1 field or DA field to all values in the dot11GroupAddressesTable. If the Address 1 field of an MPDU carrying an A-MSDU does not match any address (i.e., individual or group address) at a receiving STA, then the entire A-MSDU is discarded.

In a QoS STA, the TID parameter of the MA-UNITDATA.request primitive results in a TID being specified for the transmitted MSDU. This TID associates the MSDU with the AC or TS queue for the indicated traffic.

9.3 DCF

9.3.1 General

The basic medium access protocol is a DCF that allows for automatic medium sharing between compatible PHYs through the use of CSMA/CA and a random backoff time following a busy medium condition. In addition, all individually addressed traffic uses immediate positive acknowledgment (ACK frame) where retransmission is scheduled by the sender if no ACK is received.

The CSMA/CA protocol is designed to reduce the collision probability between multiple STAs accessing a medium, at the point where collisions would most likely occur. Just after the medium becomes idle following a busy medium (as indicated by the CS function) is when the highest probability of a collision exists. This is because multiple STAs could have been waiting for the medium to become available again. This is the situation that necessitates a random backoff procedure to resolve medium contention conflicts.

CS shall be performed both through physical and virtual mechanisms.

The virtual CS mechanism is achieved by distributing reservation information announcing the impending use of the medium. The exchange of RTS and CTS frames prior to the actual data frame is one means of distribution of this medium reservation information. The RTS and CTS frames contain a Duration field that defines the period of time that the medium is to be reserved to transmit the actual data frame and the returning ACK frame. A STA receiving either the RTS (sent by the originating STA) or the CTS (sent by the destination STA) shall process the medium reservation. Thus, a STA might be unable to receive from the originating STA and yet still know about the impending use of the medium to transmit a data frame.

Another means of distributing the medium reservation information is the Duration/ID field in individually addressed frames. This field gives the time that the medium is reserved, either to the end of the immediately following ACK, or in the case of a fragment sequence, to the end of the ACK following the next fragment.

The RTS/CTS exchange also performs both a type of fast collision inference and a transmission path check. If the return CTS is not detected by the STA originating the RTS, the originating STA may repeat the process (after observing the other medium-use rules) more quickly than if the long data frame had been transmitted and a return ACK frame had not been detected.

Another advantage of the RTS/CTS mechanism occurs where multiple BSSs utilizing the same channel overlap. The medium reservation mechanism works across the BSS boundaries. The RTS/CTS mechanism might also improve operation in a typical situation in which all STAs are able to receive from the AP, but might not be able to receive from all other STAs in the BSA.

The RTS/CTS mechanism cannot be used for MPDUs with a group addressed immediate destination because there are multiple recipients for the RTS, and thus potentially multiple concurrent senders of the CTS in response. The RTS/CTS mechanism is not used for every data frame transmission. Because the additional RTS and CTS frames add overhead inefficiency, the mechanism is not always justified, especially for short data frames.

The use of the RTS/CTS mechanism is under control of the dot11RTSThreshold attribute. This attribute may be set on a per-STA basis. This mechanism allows STAs to be configured to initiate RTS/CTS either always, never, or only on frames longer than a specified length.

NOTE—A STA configured not to initiate the RTS/CTS mechanism updates its virtual CS mechanism with the duration information contained in a received RTS or CTS frame, and responds to an RTS addressed to it with a CTS if permitted by medium access rules.

All STA that are members of a BSS are able to receive and transmit at all the data rates in the BSSBasicRateSet parameter of the MLME-START.request primitive or BSSBasicRateSet parameter of the BSSDescription representing the SelectedBSS parameter of the MLME-JOIN.request primitive; see 6.3.4.2.4 and 6.3.11.2.4. All HT STAs that are members of a BSS are able to receive and transmit using all the MCSs in the BSSBasicMCSSet parameter of the MLME-START.request primitive or BSSBasicMCSSet parameter of the BSSDescription representing the SelectedBSS parameter of the MLME-JOIN.request primitive; see 6.3.4.2.4 and 6.3.11.2.4. To support the proper operation of the RTS/CTS and the virtual CS mechanism, all STAs shall be able to interpret control frames with the Subtype field equal to RTS or CTS.

Data frames sent under the DCF shall use the frame type Data and subtype Data or Null Function. STAs receiving Data type frames shall not indicate a data frame to LLC when the subtype is Null Function, but shall indicate a data frame to LLC when the subtype is Data, even if the frame body contains zero octets.

9.3.2 Procedures common to the DCF and EDCAF

9.3.2.1 CS mechanism

Physical and virtual CS functions are used to determine the state of the medium. When either function indicates a busy medium, the medium shall be considered busy; otherwise, it shall be considered idle.

A physical CS mechanism shall be provided by the PHY. See Clause 7 for how this information is conveyed to the MAC. The details of physical CS are provided in the individual PHY specifications.

A virtual CS mechanism shall be provided by the MAC. This mechanism is referred to as the NAV. The NAV maintains a prediction of future traffic on the medium based on duration information that is announced in RTS/CTS frames prior to the actual exchange of data. The duration information is also available in the MAC headers of all frames sent during the CP other than PS-Poll frames. The mechanism for setting the NAV using RTS/CTS in the DCF is described in 9.3.2.4, use of the NAV in PCF is described in 9.4.3.3, and the use of the NAV in HCF is described in 9.19.2.2 and 9.19.3.4. Additional details regarding NAV usage and update appear in 9.3.2.5, 9.3.2.11, and 9.23.

The CS mechanism combines the NAV state and the STA's transmitter status with physical CS to determine the busy/idle state of the medium. The NAV may be thought of as a counter, which counts down to 0 at a uniform rate. When the counter is 0, the virtual CS indication is that the medium is idle; when nonzero, the indication is busy. The medium shall be determined to be busy when the STA is transmitting.

9.3.2.2 MAC-Level Acknowledgements

The reception of some frames, as described in 9.3.2.8 and 9.4.4.5, requires the receiving STA to respond with an acknowledgment if the FCS of the received frame is correct. This technique is known as positive acknowledgment.

Lack of reception of an expected frame containing an acknowledgement indicates to the STA initiating the frame exchange that an error has occurred. Note, however, that the destination STA may have received the frame correctly, and that the error may have occurred in the transfer or reception of the frame containing an acknowledgement. When a frame containing an acknowledgement is lost, the MAC that initiated the frame exchange does not receive a protocol indication of whether the initial frame was correctly received.

9.3.2.3 IFS

9.3.2.3.1 General

The time interval between frames is called the IFS. A STA shall determine that the medium is idle through the use of the CS function for the interval specified. Six different IFSs are defined to provide priority levels for access to the wireless medium. Figure 9-3 shows some of these relationships. All timings are referenced from the PHY-TXEND.confirm, PHY-TXSTART.confirm, PHY-RXSTART.indication, and PHY-RXEND.indication primitives.

The IFSs are:

a)	RIFS	reduced interframe space
b)	SIFS	short interframe space
c)	PIFS	PCF interframe space
d)	DIFS	DCF interframe space
e)	AIFS	arbitration interframe space (used by the QoS facility)
f)	EIFS	extended interframe space

The different IFSs shall be independent of the STA bit rate. The IFS timings are defined as time gaps on the medium, and the IFS timings except AIFS are fixed for each PHY (even in multirate-capable PHYs). The IFS values are determined from attributes specified by the PHY.

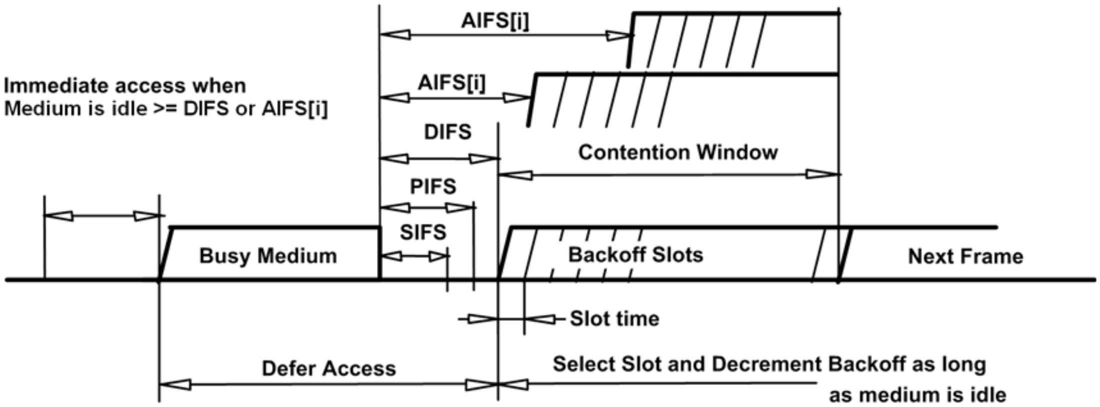


Figure 9-3—Some IFS relationships

9.3.2.3.2 RIFS

RIFS is a means of reducing overhead and thereby increasing network efficiency.

RIFS may be used in place of SIFS to separate multiple transmissions from a single transmitter, when no SIFS-separated response transmission is expected. RIFS shall not be used between frames with different RA values. The duration of RIFS is defined by the aRIFS PHY characteristic (see Table 20-25). The RIFS is the time from the end of the last symbol of the previous frame to the beginning of the first symbol of the preamble of the subsequent frame as seen at the air interface. A STA shall not allow the space between frames that are defined to be separated by a RIFS time, as measured on the medium, to vary from the nominal RIFS value (aRIFSTime) by more than ± 10% of aRIFSTime. Two frames separated by a RIFS shall both be HT PPDUs.

There are additional restrictions regarding when RIFS may be employed as defined in 9.25 and 9.26. See also 9.23.3.3.

9.3.2.3.3 SIFS

The SIFS shall be used prior to transmission of an ACK frame, a CTS frame, a PPDU containing a BlockAck frame that is an immediate response to either a BlockAckReq frame or an A-MPDU, the second or subsequent MPDU of a fragment burst, and by a STA responding to any polling by the PCF. The SIFS may also be used by a PC for any types of frames during the CFP (see 9.4). The SIFS is the time from the end of the last symbol, or signal extension if present, of the previous frame to the beginning of the first symbol of the preamble of the subsequent frame as seen at the air interface.

The SIFS timing shall be achieved when the transmission of the subsequent frame is started at the TxSIFS Slot boundary as specified in 9.3.7. An IEEE 802.11 implementation shall not allow the space between frames that are defined to be separated by a SIFS time, as measured on the medium, to vary from the nominal SIFS value by more than ±10% of aSlotTime for the PHY in use.

SIFS is the shortest of the IFSs between transmissions from different STAs. SIFS shall be used when STAs have seized the medium and need to keep it for the duration of the frame exchange sequence to be performed. Using the smallest gap between transmissions within the frame exchange sequence prevents other STAs, which are required to wait for the medium to be idle for a longer gap, from attempting to use the medium, thus giving priority to completion of the frame exchange sequence in progress.

9.3.2.3.4 PIFS

The PIFS is used to gain priority access to the medium.

The PIFS may be used as described in the following list and shall not be used otherwise:

- A STA operating under the PCF as described in 9.4
- A STA transmitting a Channel Switch Announcement frame as described in 10.9
- A STA transmitting a TIM frame as described in 10.2.1.17
- An HC starting a CFP or a TXOP as described in 9.19.3.2.3
- An HC or a non-AP QoS STA that is a polled TXOP holder recovering from the absence of an expected reception in a CAP as described in 9.19.3.2.4
- An HT STA using dual CTS protection before transmission of the CTS2 as described in 9.3.2.7
- A TXOP holder continuing to transmit after a transmission failure as described in 9.19.2.4
- An RD initiator continuing to transmit using error recovery as described in 9.25.3
- An HT AP during a PSMP sequence transmitting a PSMP recovery frame as described in 9.26.1.3
- An HT STA performing clear channel assessment (CCA) in the secondary channel before transmitting a 40 MHz mask PPDU using EDCA channel access as described in 10.15.9

With the exception of performing CCA in the secondary channel (where the timing is defined in 10.15.9), a STA using PIFS starts its transmission after its CS mechanism (see 9.3.2.1) determines that the medium is idle at the TxPIFS slot boundary as defined in 9.3.7.

9.3.2.3.5 DIFS

The DIFS shall be used by STAs operating under the DCF to transmit data frames (MPDUs) and management frames (MMPDUs). A STA using the DCF may transmit if its CS mechanism (see 9.3.2.1) determines that the medium is idle at the TxDIFS slot boundary as defined in 9.3.7 after a correctly received frame, and its backoff time has expired. A correctly received frame is one where the PHY-RXEND.indication primitive does not indicate an error and the FCS indicates the frame is error free.

9.3.2.3.6 AIFS

The AIFS shall be used by QoS STAs that access the medium using the EDCAF to transmit: all data frames (MPDUs), all management frames (MMPDUs), and the following control frames: PS-Poll, RTS, CTS (when not transmitted as a response to the RTS), BlockAckReq, and BlockAck (when not transmitted as a response to the BlockAckReq). A STA using the EDCAF shall obtain a TXOP for an AC if the STA's CS mechanism (see 9.3.2.1) determines that the medium is idle at the AIFS[AC] slot boundary (see 9.19.2.3), after a correctly received frame, and the backoff time for that AC has expired.

A QoS STA computes the time periods for each AIFS[AC] from the dot11EDCATableAIFSN attributes in the MIB. In an infrastructure BSS, QoS STAs update their dot11EDCATableAIFSN values using information in the most recent EDCA Parameter Set element of Beacon frames received from the AP of the BSS (see 8.4.2.31). A QoS AP computes the time periods for each AIFS[AC] from the dot11QAPEDCATableAIFSN attributes in its MIB.

9.3.2.3.7 EIFS

A DCF shall use EIFS before transmission, when it determines that the medium is idle following reception of a frame for which the PHY-RXEND.indication primitive contained an error or a frame for which the MAC FCS value was not correct. Similarly, a STA's EDCA mechanism under HCF shall use the EIFS-DIFS+AIFS[AC] interval. The duration of an EIFS is defined in 9.3.7. The EIFS or EIFS-DIFS+AIFS[AC] interval shall begin following indication by the PHY that the medium is idle after detection of the erroneous frame, without regard to the virtual CS mechanism. The STA shall not begin a transmission until the expiration of the later of the NAV and EIFS or EIFS-DIFS+AIFS[AC]. The EIFS and EIFS-DIFS+AIFS[AC] are defined to provide enough time for another STA to acknowledge what was, to this STA, an incorrectly received frame before this STA commences transmission. Reception of an error-free frame during the EIFS or EIFS-DIFS+AIFS[AC] resynchronizes the STA to the actual busy/idle state of the medium, so the EIFS or EIFS-DIFS+AIFS[AC] is terminated and medium access (using DIFS or AIFS as appropriate and, if necessary, backoff) continues following reception of that frame. At the expiration or termination of the EIFS or EIFS-DIFS+AIFS[AC], the STA reverts to the NAV and physical CS to control access to the medium.

EIFS shall not be invoked if the NAV is updated by the frame that would have caused an EIFS, such as when the MAC FCS fails and the L-SIG TXOP function employs L-SIG information to update the NAV. EIFS shall not be invoked for an A-MPDU if one or more of its frames are received correctly.

9.3.2.4 Setting and resetting the NAV

A STA that receives at least one valid frame within a received PSDU shall update its NAV with the information received in any valid Duration field from within that PSDU for all frames where the new NAV value is greater than the current NAV value, except for those where the RA is equal to the MAC address of the STA. Upon receipt of a PS-Poll frame, a STA shall update its NAV settings as appropriate under the data rate selection rules using a duration value equal to the time, in microseconds, required to transmit one ACK frame plus one SIFS interval, but only when the new NAV value is greater than the current NAV value. If the calculated duration includes a fractional microsecond, that value is rounded up to the next higher integer. Various additional conditions may set or reset the NAV, as described in 9.4.3.3. When the NAV is reset, a

PHY-CCARESET.request primitive shall be issued. This NAV update operation is performed when the PHY-RXEND.indication primitive is received.

Figure 9-4 indicates the NAV for STAs that may receive the RTS frame, while other STAs may only receive the CTS frame, resulting in the lower NAV bar as shown (with the exception of the STA to which the RTS was addressed).

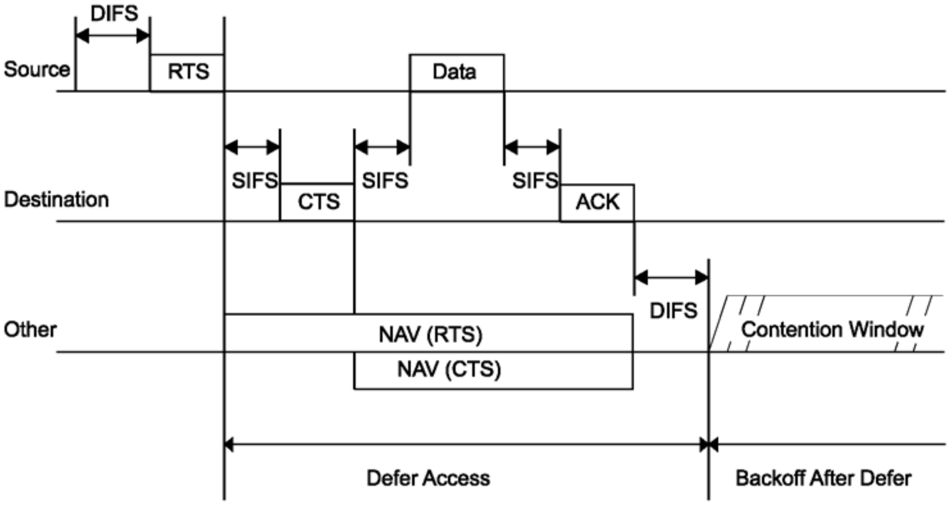


Figure 9-4—RTS/CTS/data/ACK and NAV setting

A STA that used information from an RTS frame as the most recent basis to update its NAV setting is permitted to reset its NAV if no PHY-RXSTART.indication primitive is detected from the PHY during a period with a duration of (2 × aSIFSTime) + (CTS_Time) + aPHY-RX-START-Delay + (2 × aSlotTime) starting at the PHY-RXEND.indication primitive corresponding to the detection of the RTS frame. The “CTS_Time” shall be calculated using the length of the CTS frame and the data rate at which the RTS frame used for the most recent NAV update was received.

A STA supporting L-SIG TXOP that used the information from a frame with different L-SIG duration and MAC duration endpoints (characteristics of an L-SIG TXOP initiating frame; see 9.23.5.4 for details) as the most recent basis to update its NAV setting may reset its NAV if no PHY-RXSTART.indication primitive is detected from the PHY during a period with a duration of aSIFSTime + aPHY-RX-START-Delay + (2 × aSlotTime) starting at the expiration of the L-SIG duration. For details of L-SIG duration, see 9.23.5.

9.3.2.5 RTS/CTS with fragmentation

The following is a description of using RTS/CTS for a fragmented MSDU or MMPDU. The RTS/CTS frames define the duration of the following frame and acknowledgment. The Duration/ID field in the data and ACK frames specifies the total duration of the next fragment and acknowledgment. This is illustrated in Figure 9-5.

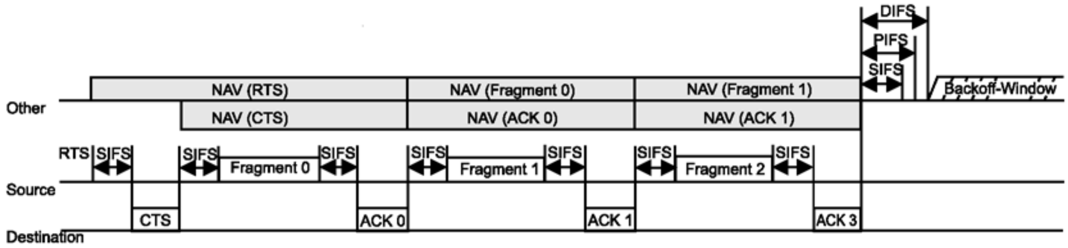


Figure 9-5—RTS/CTS with fragmented MSDU

Each frame contains information that defines the duration of the next transmission. The duration information from RTS frames shall be used to update the NAV to indicate busy until the end of ACK 0. The duration information from the CTS frame shall also be used to update the NAV to indicate busy until the end of ACK 0. Both Fragment 0 and ACK 0 shall contain duration information to update the NAV to indicate busy until the end of ACK 1. This shall be done by using the Duration/ID field in the Data and ACK frames. This shall continue until the last fragment, which shall have a duration of one ACK time plus one SIFS time, and its ACK, which shall have its Duration/ID field set to 0. Each fragment and ACK acts as a virtual RTS and CTS; therefore no further RTS/CTS frames need to be generated after the RTS/CTS that began the frame exchange sequence even though subsequent fragments may be larger than dot11RTSThreshold. At STAs using an FH PHY, when there is insufficient time before the next dwell boundary to transmit the subsequent fragment, the STA initiating the frame exchange sequence may set the Duration/ID field in the last data or management frame to be transmitted before the dwell boundary to the duration of one ACK time plus one SIFS time.

In the case where an acknowledgment is sent but not received by the source STA, STAs that heard the fragment, or ACK, mark the channel busy for the next frame exchange due to the NAV having been updated from these frames. This is the worst-case situation, and it is shown in Figure 9-6. If an acknowledgment is not sent by the destination STA, STAs that can only hear the destination STA do not update their NAV and may attempt to access the channel when their NAV updated from the previously received frame reaches 0. All STAs that hear the source are free to access the channel after their NAV updated from the transmitted fragment has expired.

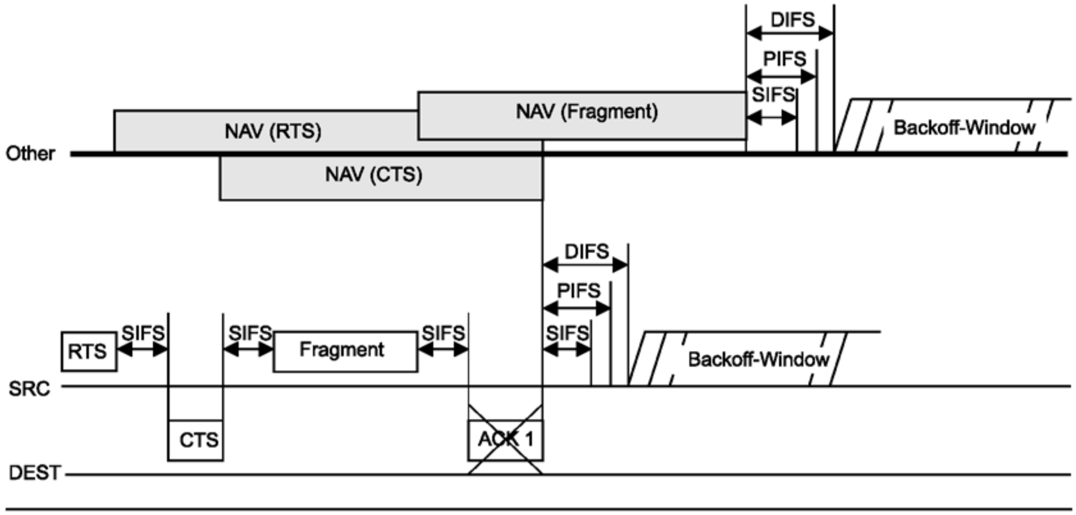


Figure 9-6—RTS/CTS with transmitter priority and missed acknowledgment

9.3.2.6 CTS procedure

A STA that is addressed by an RTS frame shall transmit a CTS frame after a SIFS period if the NAV at the STA receiving the RTS frame indicates that the medium is idle. If the NAV at the STA receiving the RTS indicates the medium is not idle, that STA shall not respond to the RTS frame. The RA field of the CTS frame shall be the value obtained from the TA field of the RTS frame to which this CTS frame is a response. The Duration field in the CTS frame shall be the duration field from the received RTS frame, adjusted by subtraction of aSIFSTime and the number of microseconds required to transmit the CTS frame at a data rate determined by the rules in 9.7.

After transmitting an RTS frame, the STA shall wait for a CTSTimeout interval, with a value of aSIFSTime + aSlotTime + aPHY-RX-START-Delay, starting at the PHY-TXEND.confirm primitive. If a PHY-RXSTART.indication primitive does not occur during the CTSTimeout interval, the STA shall conclude that the transmission of the RTS has failed, and this STA shall invoke its backoff procedure upon expiration of the CTSTimeout interval. If a PHY-RXSTART.indication primitive does occur during the CTSTimeout interval, the STA shall wait for the corresponding PHY-RXEND.indication primitive to determine whether the RTS transmission was successful. The recognition of a valid CTS frame sent by the recipient of the RTS frame, corresponding to this PHY-RXEND.indication primitive, shall be interpreted as successful response, permitting the frame sequence to continue (see Annex G). The recognition of anything else, including any other valid frame, shall be interpreted as failure of the RTS transmission. In this instance, the STA shall invoke its backoff procedure at the PHY-RXEND.indication primitive and may process the received frame.

9.3.2.7 Dual CTS protection

9.3.2.7.1 Dual CTS protection procedure

If the Dual CTS Protection field of the HT Operation element has value 1 in the Beacon frames transmitted by its AP, a non-AP HT STA shall start every TXOP with an RTS addressed to the AP. The RTS shall be an STBC frame if the STBC transmit and receive capabilities of the non-AP HT STA allow it to receive and transmit STBC frames using a single spatial stream; otherwise, the RTS shall be a non-STBC frame. The AP shall respond with a dual CTS (CTS1 followed by CTS2) separated by PIFS or SIFS. Table 9-2 describes the sequence of CTS transmissions and the required timing.

Table 9-2—Dual CTS rules

Type of RTS	CTS description	Timing
RTS (non-STBC frame)	CTS1: Same rate or MCS as the RTS (non-STBC frame) CTS2: Basic STBC MCS (STBC frame)	PIFS shall be used as the interval between CTS1 and CTS2. If the CS mechanism (see 9.3.2.1) indicates that the medium is busy at the TxPIFS slot boundary (defined in 9.3.7) following CTS1, CTS2 shall not be transmitted as part of this frame exchange.
RTS (STBC frame)	CTS1: Basic STBC MCS (STBC frame) CTS2: Lowest basic rate (non-STBC frame)	SIFS shall be used as the interval between CTS1 and CTS2. The STA resumes transmission a SIFS+CTS2+SIFS after receiving CTS1, instead of after SIFS.

The dual CTS response applies only to the AP; a non-AP STA shall respond to an RTS request with a single CTS.

If dual CTS Protection is enabled, the AP shall begin each EDCA TXOP with a CTS frame. This CTS frame uses STBC when the immediately following frame uses non-STBC and vice versa. The RA of this CTS shall be identical to the RA of the immediately following frame. The AP may continue a PIFS after the CTS, only if the CS mechanism (see 9.3.2.1) indicates that the medium is IDLE at the TxPIFS slot boundary (defined in 9.3.7) following the transmission of the CTS.

To avoid the resetting of NAV by STAs that have set their NAV due to the reception of a non-STBC RTS that is part of a dual CTS exchange, but then do not hear the CTS2, a non-AP HT STA may create a NAV that is not resettable according to the RTS NAV reset rule defined in 9.3.2.4 at the receiving STAs by initiating the TXOP with a non-STBC CTS addressed to the AP (known as *CTS-to-AP*).

NOTE—Sending a CTS-to-AP allows NAV protection to be established without causing the AP to update its NAV, as opposed to, for example, the sending of a CTS-to-self, which would potentially have caused the AP NAV to become set and then prevented it from responding to the subsequent RTS. The AP does not set a NAV in the CTS-to-AP case and is able to respond to the following RTS. The NAV at receiving STAs is not updated by the RTS because its duration does not exceed the duration of the preceding CTS, and subsequently, the NAV cannot be reset during CTS2.

An STBC CTS addressed to the AP may be transmitted prior to an STBC RTS to set a NAV that is not resettable according to the RTS NAV reset rule defined in 9.3.2.4 at receiving STAs.

NOTE—When an HT STA sends an RTS to the AP that is a non-STBC frame, the AP returns a CTS that is a non-STBC frame to the STA and then immediately transmits a CTS that is an STBC frame. The original non-AP STA is now free to transmit. But a non-HT STA that has set its NAV based on the original RTS might reset its NAV and then decrement its backoff counter, given that a SIFS + the duration of CTS2 is longer than a DIFS (i.e., the STA does not detect PHY-RXSTART.indication primitive within the period specified in 9.3.2.4). Thus, without sending a CTS-to-AP, the NAV reservation might not always work.

If dual CTS protection is enabled and a STA obtains a TXOP and does not have any frames to transmit before the expiry of the TXOP duration, the STA may indicate truncation of the TXOP provided that the remaining duration of the TXOP after the transmission of the last frame can accommodate the CF-End frame, a CF-End frame that is an STBC frame duration at the basic STBC MCS, a CF-End frame that is a non-STBC frame at the lowest basic rate, and three SIFS durations. The STA indicates truncation of the TXOP by transmitting a CF-End frame with TXVECTOR parameter restrictions as specified in 9.7.6.3.

On receiving a CF-End frame from a STA with a matching BSSID, an AP whose last transmitted HT Operation element contained the Dual CTS Protection field equal to 1 shall respond with dual CF-End frames, one CF-End frame that is an STBC frame at the basic STBC MCS and one CF-End frame that is a non-STBC frame at the lowest basic rate, after a SIFS duration. Dual CF-End frames eliminate unfairness towards STAs that are not of the same mode as the one that owns the TXOP being truncated.

If the TXOP is owned by the AP and dual CTS Protection is enabled in the system, the AP may send dual CF-End frames if it runs out of frames to transmit, provided that the remaining TXOP duration after the transmission of the last frame can accommodate a STBC CF-End frame duration at the lowest STBC basic rate, a CF-End frame that is a non-STBC frame at the lowest basic rate, and two SIFS durations.

The spacing between the dual CF-End frames sent by the AP shall be SIFS. The first CF-End frame shall use the same encoding (STBC frame versus non-STBC frame) used for transmissions in the TXOP being truncated, and the second CF-End frame shall use the other encoding.

An STBC-capable STA shall choose between control frame operation using either STBC frames or non-STBC frames. In the non-STBC frame case, it discards control frames that are STBC frames it receives. In the STBC frame case, it discards control frames that are non-STBC frames received from its own BSS. This choice is a matter of policy local at the STA.

9.3.2.7.2 Dual CTS protection examples

Figure 9-7 shows an example of the operation of the dual CTS protection mechanism. In this example, the initiating STA is an STBC non-AP STA.

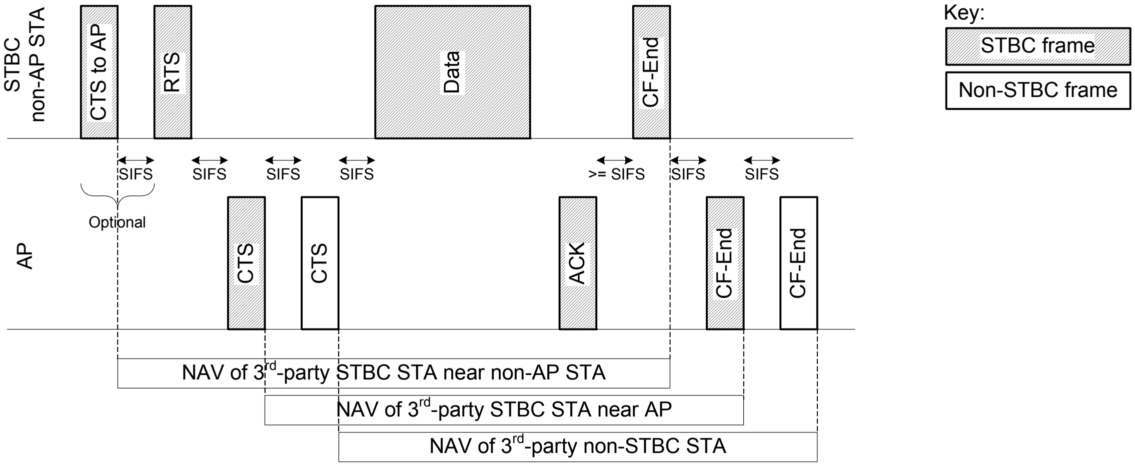


Figure 9-7—Example of dual CTS mechanism (STBC initiator)

Figure 9-8 shows an example of the operation of the dual CTS protection mechanism. In this example, the initiating STA is a non-STBC non-AP HT STA.

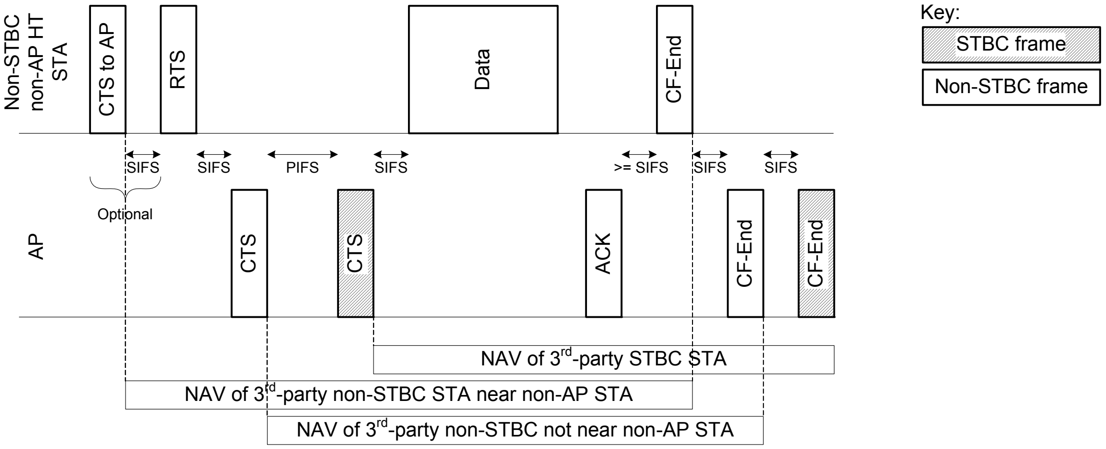


Figure 9-8—Example of the dual CTS mechanism (non-STBC initiator)

9.3.2.8 ACK procedure

The cases when an ACK frame can be generated are shown in the frame exchange sequences listed in Annex G.

On receipt of a management frame of subtype Action NoAck, a STA shall not send an ACK frame in response.

Upon successful reception of a frame of a type that requires acknowledgment with the To DS field set, an AP shall generate an ACK frame. An ACK frame shall be transmitted by the destination STA that is not an AP, when it successfully receives an individually addressed frame of a type that requires acknowledgment, but not if it receives a group addressed frame of such type. After a successful reception of a frame requiring acknowledgment, transmission of the ACK frame shall commence after a SIFS period, without regard to the busy/idle state of the medium. (See Figure 9-9.)

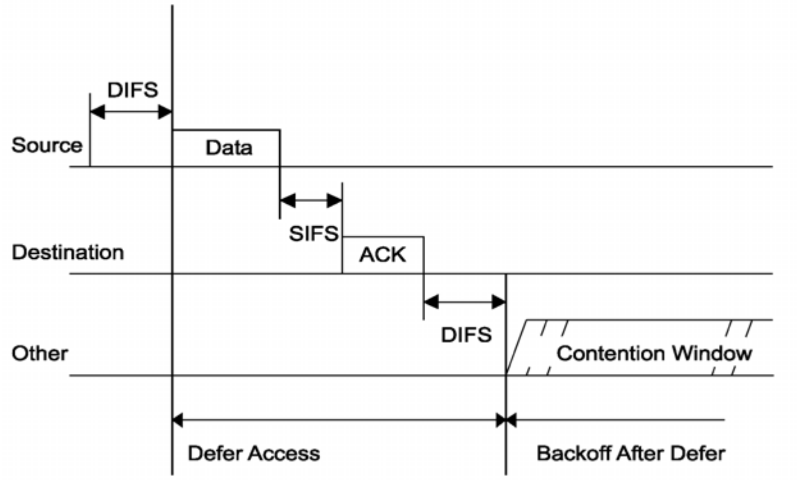


Figure 9-9—Individually addressed data/ACK MPDU

After transmitting an MPDU that requires an ACK frame as a response (see Annex G), the STA shall wait for an ACKTimeout interval, with a value of aSIFSTime + aSlotTime + aPHY-RX-START-Delay, starting at the PHY-TXEND.confirm primitive. If a PHY-RXSTART.indication primitive does not occur during the ACKTimeout interval, the STA concludes that the transmission of the MPDU has failed, and this STA shall invoke its backoff procedure upon expiration of the ACKTimeout interval. If a PHY-RXSTART.indication primitive does occur during the ACKTimeout interval, the STA shall wait for the corresponding PHY-RXEND.indication primitive to determine whether the MPDU transmission was successful. The recognition of a valid ACK frame sent by the recipient of the MPDU requiring acknowledgment, corresponding to this PHY-RXEND.indication primitive, shall be interpreted as successful acknowledgment, permitting the frame sequence to continue, or to end without retries, as appropriate for the particular frame sequence in progress. The recognition of anything else, including any other valid frame, shall be interpreted as failure of the MPDU transmission. In this instance, the STA shall invoke its backoff procedure at the PHY-RXEND.indication primitive and may process the received frame. An exception is that recognition of a valid data frame sent by the recipient of a PS-Poll frame shall also be accepted as successful acknowledgment of the PS-Poll frame.

9.3.2.9 BlockAck procedure

Upon successful reception of a frame of a type that requires an immediate BlockAck response, the receiving STA shall transmit a BlockAck frame after a SIFS period, without regard to the busy/idle state of the medium. The rules that specify the contents of this BlockAck frame are defined in 9.21.

9.3.2.10 Duplicate detection and recovery

Because MAC-level acknowledgments and retransmissions are incorporated into the protocol, there is the possibility that a frame may be received more than once. The procedures defined in this subclause attempt to filter out these duplicates. Additional duplicate filtering is performed during Receive Buffer Operation for frames that are part of a Block Ack agreement as described in 9.21.4 and 9.21.7.

Duplicate frame filtering is facilitated through the inclusion of a Sequence Control field (consisting of a sequence number and fragment number) within data and management frames as well as TID subfield in the

QoS Control field within QoS data frames. MPDUs that are part of the same MSDU or A-MSDU shall have the same sequence number, and different MSDUs or A-MSDUs have (with a high probability) a different sequence number.

A non-QoS STA shall assign sequence numbers to management frames and data frames (QoS subfield of the Subtype field is equal to 0) from a single modulo-4096 counter, starting at 0 and incrementing by 1, for each MSDU or MMPDU. A QoS STA operating as a non-QoS STA because it is in a non-QoS BSS or non-QoS IBSS shall assign sequence numbers to management frames and data frames (QoS subfield of the Subtype field is equal to 0) from a single modulo-4096 counter, starting at 0 and incrementing by 1, for each MSDU or MMPDU. A transmitting STA should cache the last used sequence number per RA for frames that are assigned sequence numbers from this counter and should ensure that the successively assigned sequence numbers for frames transmitted to a single RA do not have the same value by incrementing the counter by 2, if incrementing by 1 would have produced the same sequence number as is found in the cache for that RA.

A STA operating as a QoS STA shall maintain one modulo-4096 counter, per <Address 1, TID>, for individually addressed QoS Data frames. Sequence numbers for these frames are assigned using the counter identified by the Address 1 field and the TID subfield of the QoS Control field of the frame, and that counter is incremented by 1 for each MSDU or A-MSDU corresponding to that <Address 1, TID> tuple. Sequence numbers for management frames, QoS data frames with a group address in the Address 1 field, and all non-QoS data frames transmitted by QoS STAs shall be assigned using an additional single modulo-4096 counter, starting at 0 and incrementing by 1 for each such MSDU, A-MSDU, or MMPDU, except that a QoS STA may use values from additional modulo-4096 counters per <Address 1, TID> for sequence numbers assigned to time priority management frames. A transmitting STA should cache the last used sequence number per RA for frames that are assigned sequence numbers from this counter and should ensure that the successively assigned sequence numbers for frames transmitted to a single RA do not have the same value by incrementing the counter by 2, if incrementing by 1 would have produced the same sequence number as is found in the cache for that RA. Sequence numbers for QoS (+)Null frames may be set to any value.

A receiving STA shall keep a cache of recently received <Address 2, sequence-number, fragment-number> tuples from frames that are not QoS Data frames. The receiving STA shall keep at least the most recent cache entry per <Address 2> value in this cache. The receiving QoS STA shall also keep a cache of recently received <Address 2, TID, sequence-number, fragment-number> tuples from QoS Data frames from all STAs from which it has received QoS data frames. The receiving QoS STA shall keep at least the most recent cache entry per <Address 2, TID> pair in this cache. The receiving STA should maintain two additional caches, one containing entries of recently received <Address 2, sequence-number, fragment-number> tuples from received management frames that are not time priority management frames and the other containing entries of recently received <Address 2, sequence-number, fragment-number> tuples from received time priority management frames. The receiving STA should not include the entries in these two additional caches in any other caches. In each of these two caches, the receiving STA should keep at least the most recent cache entry per <Address 2> value. A receiving STA should omit tuples obtained from group addressed and ATIM frames from all caches.

A receiving STA shall reject as a duplicate frame any frame that is not a QoS Data frame in which the Retry bit in the Frame Control field is 1 and that matches an <Address 2, sequence-number, fragment-number> tuple of an entry in the cache that contains tuples of that format, unless the frame is a management frame and the STA is maintaining separate caches for <Address 2, sequence-number, fragment-number> tuples from received management frames. A receiving QoS STA shall also reject as a duplicate frame any QoS Data frame in which the Retry bit in the Frame Control field is 1 and that matches an <Address 2, TID, sequence-number, fragment-number> tuple of an entry in the cache that contains tuples of that format. A STA that is maintaining separate caches for <Address 2, sequence-number, fragment-number> tuples from received management frames shall reject as a duplicate frame any management frame that is not a time priority management frame in which the Retry bit in the Frame Control field is 1 and that matches an <Address 2, sequence-number, fragment-number> tuple of an entry in the management cache that contains tuples from frames that are not time priority management frames. A STA that is maintaining separate caches for <Address 2, sequence-number, fragment-number> tuples from received management frames shall reject as a duplicate frame any time priority management frame in which the Retry bit in the Frame Control field is 1 and that matches an <Address 2, sequence-number, fragment-number> tuple of an entry in the cache that contains tuples from time priority management frames.

There is a small possibility that a frame may be improperly rejected due to such a match; however, this occurrence would be rare and simply results in a lost frame (similar to an FCS error in other LAN protocols).

NOTE—The receiver STA performs the ACK and (for an AP) PS procedures on all successfully received frames requiring acknowledgment, even if the frame is discarded due to duplicate filtering.

9.3.2.11 NAV distribution

When a node needs to distribute NAV information, for instance, to reserve the medium for a transmission of a nonbasic rate frame (that may not be heard by other nodes in the BSS), the node may first transmit a CTS frame with the RA field equal to its own MAC address (CTS-to-self) and with a duration value that protects the pending transmission, plus possibly an ACK frame.

The CTS-to-self NAV distribution mechanism is lower in network overhead cost than is the RTS/CTS NAV distribution mechanism, but CTS-to-self is less robust against hidden nodes and collisions than RTS/CTS. STAs employing a NAV distribution mechanism should choose a mechanism such as CTS-to-self or RTS/CTS that is appropriate for the given network conditions. If errors occur when employing the CTS-to-self mechanism, STAs should switch to a more robust mechanism.

9.3.2.12 Operation of aSlotTime

STAs shall set the MAC variable aSlotTime to the short slot value upon transmission or reception of Beacon, Probe Response, Association Response, and Reassociation Response MMPDUs from the BSS that the STA has joined or started and that have the short slot subfield equal to 1 when dot11ShortSlotTimeOptionImplemented is true. STAs shall set the MAC variable aSlotTime to the long slot value upon transmission or reception of Beacon, Probe Response, Association Response, and Reassociation Response MMPDUs from the BSS that the STA has joined or started and that have the short slot subfield equal to 0 when dot11ShortSlotTimeOptionImplemented is true. STAs shall set the MAC variable aSlotTime to the long slot value at all times when dot11ShortSlotTime-OptionImplemented is false. When dot11ShortSlotTimeOptionImplemented is not present, or when the PHY supports only a single slot time value, then the STA shall set the MAC variable aSlotTime to the slot value appropriate for the attached PHY.

9.3.3 Random backoff time

A STA desiring to initiate transfer of data MPDUs and/or MMPDUs using the DCF shall invoke the CS mechanism (see 9.3.2.1) to determine the busy/idle state of the medium. If the medium is busy, the STA shall defer until the medium is determined to be idle without interruption for a period of time equal to DIFS when the last frame detected on the medium was received correctly, or after the medium is determined to be idle without interruption for a period of time equal to EIFS when the last frame detected on the medium was not received correctly. After this DIFS or EIFS medium idle time, the STA shall then generate a random backoff period (defined by Equation (9-1)) for an additional deferral time before transmitting, unless the backoff timer already contains a nonzero value, in which case the selection of a random number is not needed and not performed. This process minimizes collisions during contention between multiple STAs that have been deferring to the same event.

$$\text{Backoff Time} = \text{Random}() \times \text{aSlotTime} \tag{9-1}$$

where

Random() = Pseudorandom integer drawn from a uniform distribution over the interval [0,CW], where CW is an integer within the range of values of the PHY characteristics aCWmin and aCWmax, aCWmin ≤ CW ≤ aCWmax. It is important that designers recognize the need for statistical independence among the random number streams among STAs.

aSlotTime = The value of the correspondingly named PHY characteristic.

The contention window (CW) parameter shall take an initial value of aCWmin. Every STA shall maintain a STA short retry count (SSRC) as well as a STA long retry count (SLRC), both of which shall take an initial value of 0. The SSRC shall be incremented when any short retry count (SRC) associated with any MPDU of type Data is incremented. The SLRC shall be incremented when any long retry count (LRC) associated with any MPDU of type Data is incremented. The CW shall take the next value in the series every time an unsuccessful attempt to transmit an MPDU causes either STA retry counter to increment, until the CW reaches the value of aCWmax. A retry is defined as the entire sequence of frames sent, separated by SIFS intervals, in an attempt to deliver an MPDU, as described in Annex G. Once it reaches aCWmax, the CW shall remain at the value of aCWmax until the CW is reset. This improves the stability of the access protocol under high-load conditions. See Figure 9-10.

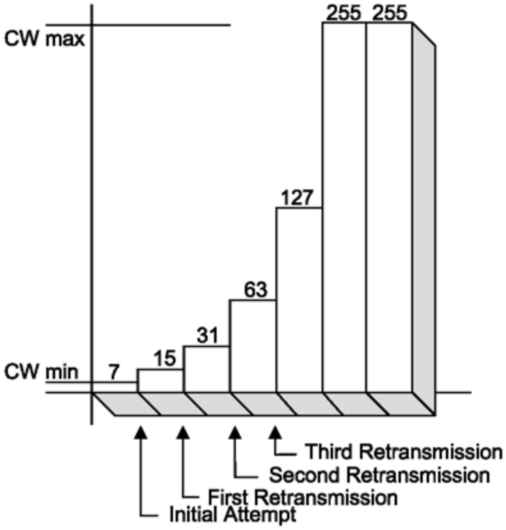


Figure 9-10—Example of exponential increase of CW

The CW shall be reset to aCWmin after every successful attempt to transmit a frame containing all or part of an MSDU or MMPDU, when SLRC reaches dot11LongRetryLimit, or when SSRC reaches dot11ShortRetryLimit. The SSRC shall be reset to 0 when a CTS frame is received in response to an RTS frame, when a BlockAck frame is received in response to a BlockAckReq frame, when an ACK frame is received in response to the transmission of a frame of length greater than dot11RTSThreshold containing all or part of an MSDU or MMPDU, or when a frame with a group address in the Address1 field is transmitted. The SLRC shall be reset to 0 when an ACK frame is received in response to transmission of a frame containing all or part of an MSDU or MMPDU of , or when a frame with a group address in the Address1 field is transmitted.

The set of CW values shall be sequentially ascending integer powers of 2, minus 1, beginning with a PHY-specific aCWmin value, and continuing up to and including a PHY-specific aCWmax value.

9.3.4 DCF access procedure

9.3.4.1 Introduction

The CSMA/CA access method is the foundation of the DCF. The operational rules vary slightly between the DCF and the PCF.

9.3.4.2 Basic access

Basic access refers to the core mechanism a STA uses to determine whether it may transmit using the DCF.

In general, a STA may transmit a pending MPDU when it is operating under the DCF access method, either in the absence of a PC, or in the CP of the PCF access method, when the STA determines that the medium is idle for greater than or equal to a DIFS period, or an EIFS period if the immediately preceding medium-busy event was caused by detection of a frame that was not received at this STA with a correct MAC FCS value. If, under these conditions, the medium is determined by the CS mechanism to be busy when a STA desires to initiate the initial frame of a frame exchange sequence (described in Annex G), exclusive of the CF period, the random backoff procedure described in 9.3.4.3 shall be followed. There are conditions, specified in 9.3.4.3 and 9.3.4.5, where the random backoff procedure shall be followed even for the first attempt to initiate a frame exchange sequence.

In a STA having an FH PHY, control of the channel is lost at the dwell time boundary and the STA shall have to contend for the channel after that dwell boundary. It is required that STAs having an FH PHY complete transmission of the entire MPDU and associated acknowledgment (if required) before the dwell time boundary. If, when transmitting or retransmitting an MPDU, there is not enough time remaining in the dwell to allow transmission of the MPDU plus the acknowledgment (if required), the STA shall defer the transmission by selecting a random backoff time, using the present CW (without advancing to the next value in the series). The short retry counter and long retry counter for the MSDU are not affected.

The basic access mechanism is illustrated in Figure 9-11.

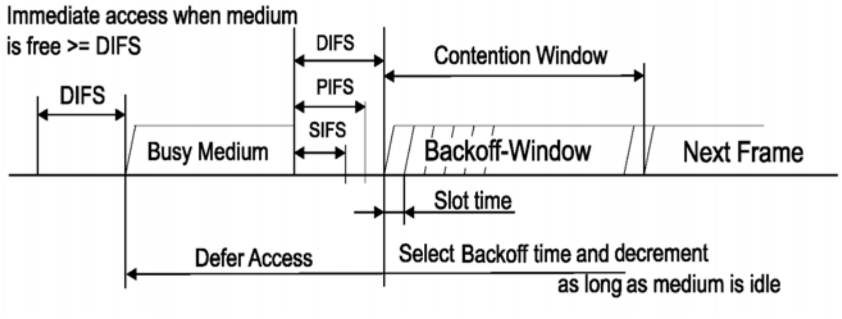


Figure 9-11—Basic access method

9.3.4.3 Backoff procedure for DCF

This subclause describes backoff procedure that is to be invoked when DCF is used. For the backoff procedure when EDCA is used, see 9.19.2.5.

The backoff procedure shall be invoked for a STA to transfer a frame when finding the medium busy as indicated by either the physical or virtual CS mechanism (see Figure 9-12). The backoff procedure shall also be invoked when a transmitting STA infers a failed transmission as defined in 9.3.2.6 or 9.3.2.8.

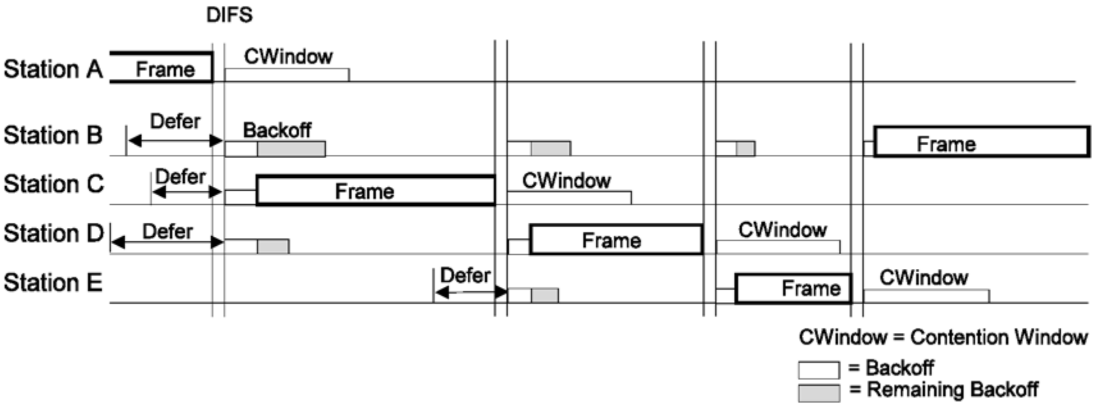


Figure 9-12—Backoff procedure

To begin the backoff procedure, the STA shall set its Backoff Timer to a random backoff time using the equation in 9.3.3. All backoff slots occur following a DIFS period during which the medium is determined to be idle for the duration of the DIFS period, or following an EIFS period during which the medium is determined to be idle for the duration of the EIFS period, as appropriate (see 9.3.2.3).

A STA performing the backoff procedure shall use the CS mechanism (see 9.3.2.1) to determine whether there is activity during each backoff slot. If no medium activity is indicated for the duration of a particular backoff slot, then the backoff procedure shall decrement its backoff time by aSlotTime.

If the medium is determined to be busy at any time during a backoff slot, then the backoff procedure is suspended; that is, the backoff timer shall not decrement for that slot. The medium shall be determined to be idle for the duration of a DIFS period or EIFS, as appropriate (see 9.3.2.3), before the backoff procedure is allowed to resume. Transmission shall commence when the Backoff Timer reaches 0.

A backoff procedure shall be performed immediately after the end of every transmission with the More Fragments bit equal to 0 of an MPDU of type Data, Management, or Control with subtype PS-Poll, even if no additional transmissions are currently queued. In the case of successful acknowledged transmissions, this backoff procedure shall begin at the end of the received ACK frame. In the case of unsuccessful transmissions requiring acknowledgment, this backoff procedure shall begin at the end of the ACKTimeout interval (as defined in 9.3.2.8). An unsuccessful transmission is one where an ACK frame is not received from the STA addressed by the RA field of the transmitted frame and the value of the RA field is an individual address. If the transmission is successful, the CW value reverts to aCWmin before the random backoff interval is chosen, and the SSRC and/or SLRC are updated as described in 9.3.3. The result of this procedure is that transmitted frames from a STA are always separated by at least one backoff interval.

The effect of this procedure is that when multiple STAs are deferring and go into random backoff, then the STA selecting the smallest backoff time using the random function wins the contention (assuming all of the contending STAs detect the same instances of WM activity at their respective receivers).

In an IBSS the backoff time for a pending non-Beacon or non-ATIM transmission shall not decrement in the period from the TBTT until the expiration of the ATIM window, and the backoff time for a pending ATIM management frame shall decrement only within the ATIM window. (See Clause 10.) Within an IBSS a separate backoff interval shall be generated to precede the transmission of a Beacon frame, as described in 10.1.3.3.

9.3.4.4 Recovery procedures and retransmit limits

Under DCF, error recovery is always the responsibility of the STA that initiates a frame exchange sequence (described in Annex G). Many circumstances may cause an error to occur that requires recovery. For example, the CTS frame might not be returned after an RTS frame is transmitted. This may happen due to a collision with another transmission, due to interference in the channel during the RTS or CTS frame, or because the STA receiving the RTS frame has an active virtual CS condition (indicating a busy medium time period).

Error recovery shall be attempted by retrying transmissions for frame exchange sequences that the initiating STA infers have failed. Retries shall continue, for each failing frame exchange sequence, until the transmission is successful, or until the relevant retry limit is reached, whichever occurs first. STAs shall maintain a SRC and a LRC for each MSDU or MMPDU awaiting transmission. These counts are incremented and reset independently of each other.

After an RTS frame is transmitted, the STA shall perform the CTS procedure, as defined in 9.3.2.6. If the RTS transmission fails, the SRC for the MSDU or MMPDU and the SSRC are incremented. This process shall continue until the number of attempts to transmit that MSDU or MMPDU reaches dot11ShortRetryLimit.

After transmitting a frame that requires acknowledgment, the STA shall perform the ACK procedure, as defined in 9.3.2.8. The SRC for an MPDU of type Data or MMPDU and the SSRC shall be incremented every time transmission of a MAC frame of length less than or equal to dot11RTSThreshold fails for that MPDU of type Data or MMPDU. This SRC and the SSRC shall be reset when a MAC frame of length less than or equal to dot11RTSThreshold succeeds for that MPDU of type Data or MMPDU. The LRC for an MPDU of type Data or MMPDU and the SLRC shall be incremented every time transmission of a MAC frame of length greater than dot11RTSThreshold fails for that MPDU of type Data or MMPDU. This LRC and the SLRC shall be reset when a MAC frame of length greater than dot11RTSThreshold succeeds for that MPDU of type Data or MMPDU. All retransmission attempts for an MPDU of type Data or MMPDU that has failed the ACK procedure one or more times shall be made with the Retry field set to 1 in the Data or Management type frame.

Retries for failed transmission attempts shall continue until the SRC for the MPDU of type Data or MMPDU is equal to dot11ShortRetryLimit or until the LRC for the MPDU of type Data or MMPDU is equal to dot11LongRetryLimit. When either of these limits is reached, retry attempts shall cease, and the MPDU of type Data (and any MSDU of which it is a part) or MMPDU shall be discarded.

A STA in PS mode, in an ESS, initiates a frame exchange sequence by transmitting a PS-Poll frame to request data from an AP. In the event that neither an ACK frame nor a data frame is received from the AP in response to a PS-Poll frame, then the STA shall retry the sequence, by transmitting another PS-Poll frame. If the AP sends a data frame in response to a PS-Poll frame, but fails to receive the ACK frame acknowledging this data frame, the next PS-Poll frame from the same STA may cause a retransmission of the last MSDU [26]. If the AP responds to a PS-Poll by transmitting an ACK frame, then responsibility for the data frame delivery error recovery shifts to the AP because the data are transferred in a subsequent frame exchange sequence, which is initiated by the AP. The AP shall attempt to deliver one MSDU to the STA that transmitted the PS-Poll, using any frame exchange sequence valid for an individually addressed MSDU. If the PS STA that transmitted the PS-Poll returns to Doze state after transmitting the ACK frame in response to successful receipt of this MSDU, but the AP fails to receive this ACK frame, then the AP retries transmission of this MSDU until the relevant retry limit is reached. See Clause 10 for details on filtering of extra PS-Poll frames.

9.3.4.5 Control of the channel

The SIFS is used to provide an efficient MSDU delivery mechanism. Once the STA has contended for the channel, that STA shall continue to send fragments until either all fragments of a single MSDU or MMPDU have been sent, an acknowledgment is not received, or the STA is restricted from sending any additional

[26] This duplicate MSDU is filtered at the receiving STA using the duplicate frame filtering mechanism.

fragments due to a dwell time boundary. Should the sending of the fragments be interrupted due to one of these reasons, when the next opportunity for transmission occurs the STA shall resume transmission. The algorithm by which the STA decides which of the outstanding MSDUs shall next be attempted after an unsuccessful transmission (as defined in 9.3.4.3) attempt is beyond the scope of this standard, but any such algorithm shall comply with the restrictions listed in 9.8.

Figure 9-13 illustrates the transmission of a multiple-fragment MSDU using the SIFS.

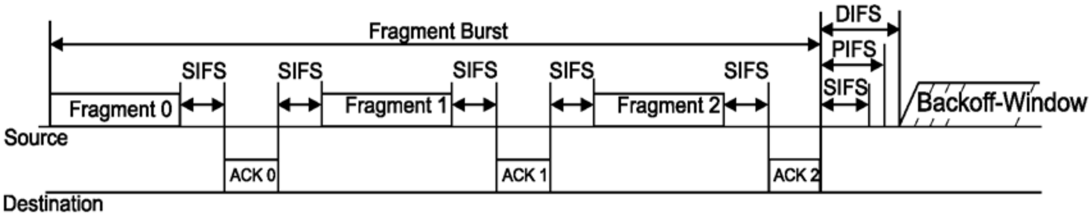


Figure 9-13—Transmission of a multiple-fragment MSDU using SIFS

When the source STA transmits a fragment, it shall release the channel, then immediately monitor the channel for an acknowledgment as described in 9.3.2.8.

When the destination STA has finished sending the acknowledgment, the SIFS following the acknowledgment shall be reserved for the source STA to continue (if necessary) with another fragment. The STA sending the acknowledgment shall not transmit on the channel immediately following the acknowledgment.

The process of sending multiple fragments after contending for the channel is defined as a fragment burst.

If the source STA receives an acknowledgment but there is not enough time to transmit the next fragment and receive an acknowledgment due to an impending dwell boundary, the source STA shall contend for the channel at the beginning of the next dwell time.

If the source STA does not receive an acknowledgment frame, it shall attempt to retransmit the failed MPDU or another eligible MPDU, as defined in 9.8, after performing the backoff procedure and the contention process.

After a STA contends for the channel to retransmit a fragment of an MSDU, it shall start with the last fragment that was not acknowledged. The destination STA shall receive the fragments in order (because the source sends them in order and they are individually acknowledged). It is possible, however, that the destination STA may receive duplicate fragments. It shall be the responsibility of the receiving STA to detect and discard duplicate fragments.

A STA shall transmit after the SIFS only under the following conditions during a fragment burst:

- The STA has just received a fragment that requires acknowledgment.
- The source STA has received an acknowledgment for a previous fragment, has more fragment(s) for the same MSDU to transmit, and there is enough time before the next dwell boundary to send the next fragment and receive its acknowledgment.

The following rules shall also apply:

- When a STA has transmitted a frame other than an initial or intermediate fragment, that STA shall not transmit on the channel following the acknowledgment for that frame, without performing the backoff procedure.

— When an MSDU has been successfully delivered or all retransmission attempts have been exhausted, and the STA has a subsequent MSDU to transmit, then that STA shall perform a backoff procedure.

9.3.5 Individually addressed MPDU transfer procedure

A STA using the DCF shall use an RTS/CTS exchange for individually addressed frames when the length of the PSDU is greater than the length threshold indicated by the dot11RTSThreshold attribute. A STA may also use an RTS/CTS exchange for individually addressed frames when it is necessary to distribute the NAV or when it is necessary to establish protection (see 9.23). Otherwise a STA using the DCF shall not use the RTS/CTS exchange.

The dot11RTSThreshold attribute is a managed object within the MAC MIB, and its value can be set and retrieved by the MLME. If dot11RTSThreshold is 0, all MPDUs shall be delivered with the use of RTS/CTS. If the value of dot11RTSThreshold is larger than the maximum PSDU length, all PSDUs shall be delivered without RTS/CTS exchanges.

When an RTS/CTS exchange is used, the PSDU shall be transmitted starting one SIFS period after the end of the CTS frame.

NOTE—No regard is given to the busy or idle status of the medium when transmitting this PSDU.

When an RTS/CTS exchange is not used, the PSDU shall be transmitted following the success of the basic access procedure. With or without the use of the RTS/CTS exchange procedure, the STA that is the destination of a data frame shall follow the ACK procedure.

9.3.6 Group addressed MPDU transfer procedure

In the absence of a PCF, when group addressed MPDUs in which the To DS field is 0 are transferred from a STA, only the basic access procedure shall be used. Regardless of the length of the frame, no RTS/CTS exchange shall be used. In addition, no ACK shall be transmitted by any of the recipients of the frame. Any group addressed MPDUs in which the To DS field is 1 transferred from a STA shall, in addition to conforming to the basic access procedure of CSMA/CA, obey the rules for RTS/CTS exchange and the ACK procedure because the MPDU is directed to the AP. When dot11SSPNInterfaceActivated is true, an AP shall distribute the group addressed message into the BSS only if dot11NonAPStationAuthSourceMulticast in the dot11InterworkingEntry identified by the source MAC address in the received message is true. When dot11SSPNInterfaceActivated is false, the group addressed message shall be distributed into the BSS. The STA originating the message receives the message as a group addressed message (prior to any filtering). Therefore, all STAs shall filter out group addressed messages that contain their address as the source address. When dot11SSPNInterfaceActivated is false, group addressed MSDUs shall be propagated throughout the ESS. When dot11SSPNInterfaceActivated is true, group addressed MSDUs shall be propagated throughout the ESS only if dot11NonAPStationAuthSourceMulticast in the dot11InterworkingEntry identified by the source MAC address in the received message is true.

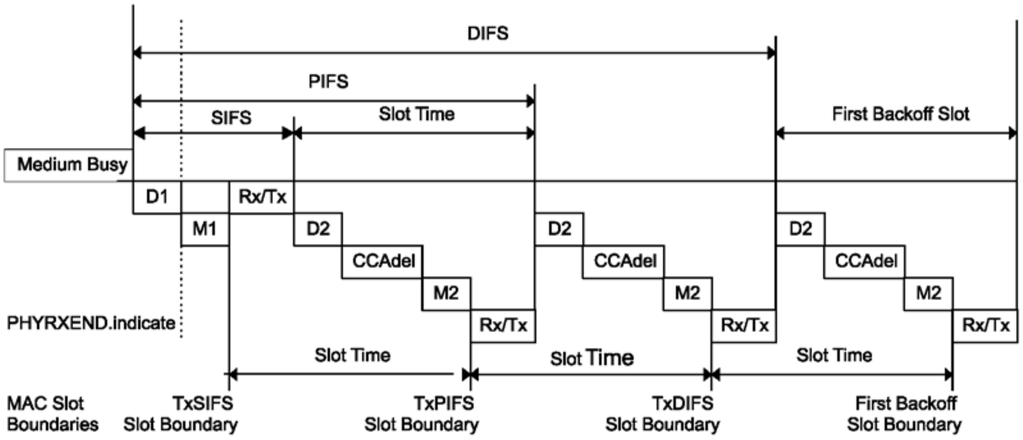
There is no MAC-level recovery on group addressed frames, except for those frames in which the To DS field is 1. As a result, the reliability of this traffic is reduced, relative to the reliability of individually addressed traffic, due to the increased probability of lost frames from interference, collisions, or time-varying channel properties.

An STBC-capable STA shall discard either all received group addressed data frames that are STBC frames or all received group addressed data frames that are non-STBC frames. How it makes this decision is outside the scope of this standard.

A STA shall discard an MPDU with a group address in the Address 1 field if the value in the Address 1 field does not match any value in the dot11GroupAddressesTable and does not match the Broadcast address value.

9.3.7 DCF timing relations

The relationships between the IFS specifications are defined as time gaps on the medium. The associated attributes are provided by the specific PHY. (See Figure 9-14.)



D1 = aRxRFDelay + aRxPLCPDelay (referenced from the end of the last symbol of a frame on the medium)
D2 = D1 + Air Propagation Time
Rx/Tx = aRXTXTurnaroundTime (begins with a PHYTXSTART.request)
M1 = M2 = aMACProcessingDelay
CCAdel = aCCA Time - D1

Figure 9-14—DCF timing relationships

All medium timings that are referenced from the end of the transmission are referenced from the end of the last symbol, or signal extension if present, of the PPDU. The beginning of transmission refers to the first symbol of the preamble of the next PPDU. All MAC timings are referenced from the PHY-TXEND.confirm, PHY-TXSTART.confirm, PHY-RXSTART.indication, and PHY-RXEND.indication primitives.

aSIFSTime and aSlotTime are determined per PHY, aSIFSTime is fixed, and aSlotTime can change dynamically as aAirPropagationTime changes (see 9.18.6).

aSIFSTime is: aRxRFDelay + aRxPLCPDelay + aMACProcessingDelay + aRxTxTurnaroundTime.

aSlotTime is: aCCATime + aRxTxTurnaroundTime + aAirPropagationTime
+ aMACProcessingDelay.

The PIFS and DIFS are derived by the Equation (9-2) and Equation (9-3), as illustrated in Figure 9-14.

$$\text{PIFS} = \text{aSIFSTime} + \text{aSlotTime} \tag{9-2}$$

$$\text{DIFS} = \text{aSIFSTime} + 2 \times \text{aSlotTime} \tag{9-3}$$

The EIFS is derived from the SIFS and the DIFS and the length of time it takes to transmit an ACK frame at the lowest PHY mandatory rate by Equation (9-4).

$$\text{EIFS} = \text{aSIFSTime} + \text{DIFS} + \text{ACKTxTime} \tag{9-4}$$

where

ACKTxTime is the time expressed in microseconds required to transmit an ACK frame, including preamble, PLCP header and any additional PHY dependent information, at the lowest PHY mandatory rate.

Figure 9-14 illustrates the relation between the SIFS, PIFS, and DIFS as they are measured on the medium and the different MAC slot boundaries TxSIFS, TxPIFS, and TxDIFS. These slot boundaries define when the transmitter shall be turned on by the MAC to meet the different IFS timings on the medium, after subsequent detection of the CCA result of the previous slot time.

Equation (9-5), Equation (9-6) and Equation (9-7) define the MAC Slot Boundaries, using attributes provided by the PHY, which are such that they compensate for implementation timing variations. The starting reference of these slot boundaries is again the end of the last symbol of the previous PPDU.

$$\text{TxSIFS} = \text{SIFS} - \text{aRxTxTurnaroundTime} \tag{9-5}$$

$$\text{TxPIFS} = \text{TxSIFS} + \text{aSlotTime} \tag{9-6}$$

$$\text{TxDIFS} = \text{TxSIFS} + 2 \times \text{aSlotTime} \tag{9-7}$$

The tolerances are specified in the physical layer management entity (PLME) SAP interface specification (see 6.5) and shall only apply to the SIFS specification so that tolerances shall not accumulate.

9.3.8 Signal Extension

Transmissions of frames with TXVECTOR parameter FORMAT of type NON_HT with NON_HT_MODULATION values of ERP-OFDM, DSSS-OFDM, and NON_HT_DUPOFDM and transmissions of frames with TXVECTOR parameter FORMAT with values of HT_MF and HT_GF include a period of no transmission of duration aSignalExtension, except for RIFS transmissions. The purpose of this signal extension is to enable the NAV value of Clause 17 STAs to be set correctly.

When an HT STA transmits a PPDU using a RIFS and with the TXVECTOR parameter FORMAT equal to NON_HT with the NON_HT_MODULATION parameter equal to one of ERP-OFDM, DSSS-OFDM, and NON_HT_DUPOFDM or a PPDU using a RIFS and with the TXVECTOR parameter FORMAT equal to HT_MF or HT_GF, it shall set the TXVECTOR parameter NO_SIG_EXTN to true. Otherwise, it shall set the TXVECTOR parameter NO_SIG_EXTN to false.

9.3.9 Determination of PLME aCWmin characteristics

In the case of the Clause 19 ERP, the aCWmin value is dependent on the requestor’s characteristic rate set. The characteristic rate set is equal to the IBSS’s supported rate set when the STA is operating as a member of an IBSS. It is equal to the AP’s supported rate set when the STA is associated with an AP. At all other times, it is equal to the STA’s mandatory rate set. The MAC variable aCWmin is set to aCWmin(0) if the characteristic rate set includes only rates in the set 1, 2, 5.5, 11; otherwise, aCWmin is set to aCWmin(1). If the returned value for aCWmin is a scalar, then the MAC always sets the variable aCWmin to the returned scalar value of aCWmin.

9.4 PCF

9.4.1 General

The PCF provides CF frame transfer. It is an option for an AP to be able to become the PC. A non-AP STA shall not become a PC. All STAs inherently obey the medium access rules of the PCF, because these rules are based on the DCF, and all STAs set their NAV at the beginning of each CFP. The operating characteristics of the PCF are such that all STAs are able to operate in the presence of a BSS in which a PC is operating, and, if associated with a point-coordinated BSS, are able to receive frames sent under PCF control. It is also an option for a STA to be able to respond to a CF-Poll received from a PC. A STA that is able to respond to CF-Polls is referred to as being CF-Pollable, and may request to be polled by an active PC. CF-Pollable STAs and the PC do not use RTS/CTS in the CFP. When polled by the PC, a CF-Pollable STA may transmit only one MPDU, which might be sent to the PC but may have any destination, and may "piggyback" the acknowledgment of a frame received from the PC using particular data frame subtypes for this transmission. If the data frame is not in turn acknowledged, the CF-Pollable STA shall not retransmit the frame unless it is polled again by the PC, or it decides to retransmit during the CP. If the addressed recipient of a CF transmission is not CF-Pollable, that STA acknowledges the transmission using the DCF acknowledgment rules, and the PC retains control of the medium. A PC may use CF frame transfer solely for delivery of frames to STAs, and never to poll CF-Pollable STAs.

A PC may perform a backoff on retransmission of an unacknowledged frame during the CFP. A PC that is maintaining a polling list may retry the unacknowledged frame the next time the particular AID is at the top of the polling list.

A PC may retransmit an unacknowledged frame during the CFP after a PIFS time.

When more than one point-coordinated BSS is operating on the same PHY channel in overlapping space, the potential exists for collisions between PCF transfer activities by the independent PCs. The rules under which multiple, overlapping point-coordinated BSSs may coexist are presented in 9.4.4.3. As shown in Figure 9-1 (in 9.2), the PCF is built on top of the CSMA/CA-based DCF, by utilizing the access priority provisions provided by this scheme. An active PC shall be located at an AP, which restricts PCF operation to infrastructure networks. PCF is activated at a PC-capable AP by setting the CFPMaxDuration parameter in the CF Parameter Set of the MLME-START.request primitive to a nonzero value.

Data frames sent by, or in response to polling by, the PC during the CFP shall use the appropriate data subtypes based upon the following usage rules:

- Data+CF-Poll, Data+CF-Ack+CF-Poll, CF-Poll, and CF-Ack+CF-Poll shall be sent only by a PC.
- Data, Data+CF-Ack, Null Function, and CF-Ack may be sent by a PC or by any CF-Pollable STA.

STAs receiving Data type frames shall consider the frame body as the basis of a possible indication to LLC only if the frame is of subtype Data, Data+CF-Ack, Data+CF-Poll, or Data+CF-Ack+CF-Poll. CF-Pollable STAs shall interpret all subtype bits of received Data type frames that contain the BSSID of the current BSS for CF purposes, but shall not inspect the frame body unless the frame is of subtype Data, Data+CF-Ack, Data+CF-Poll, or Data+CF-Ack+CF-Poll.

9.4.2 CFP structure and timing

The PCF controls frame transfers during a CFP. The CFP shall alternate with a CP, when the DCF controls frame transfers, as shown in Figure 9-15. Each CFP shall begin with a Beacon frame that contains a DTIM element (referred to as DTIM Beacon frame). The CFPs shall occur at a defined repetition rate, which shall be synchronized with the beacon interval as specified in the following paragraphs.

The PC generates CFPs at the CFP repetition interval (CFPPeriod), which is defined as a number of DTIM intervals. The PC shall determine the CFPPeriod (depicted as a repetition interval in the illustrations in Figure 9-15 and Figure 9-16) to use from dot11CFPPeriod. This value, in units of DTIM intervals, shall be communicated to other STAs in the BSS in the CFPPeriod field of the CF Parameter Set element of Beacon frames. The CF Parameter Set element shall be present only in Beacon and Probe Response frames transmitted by STAs containing an active PC.

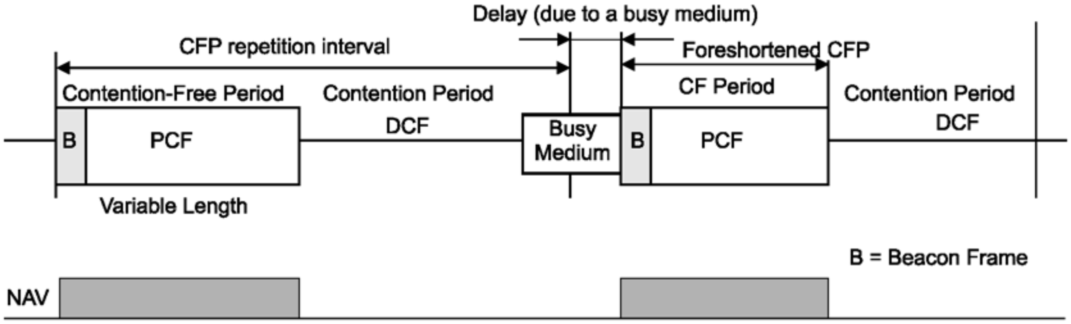


Figure 9-15—CFP/CP alternation

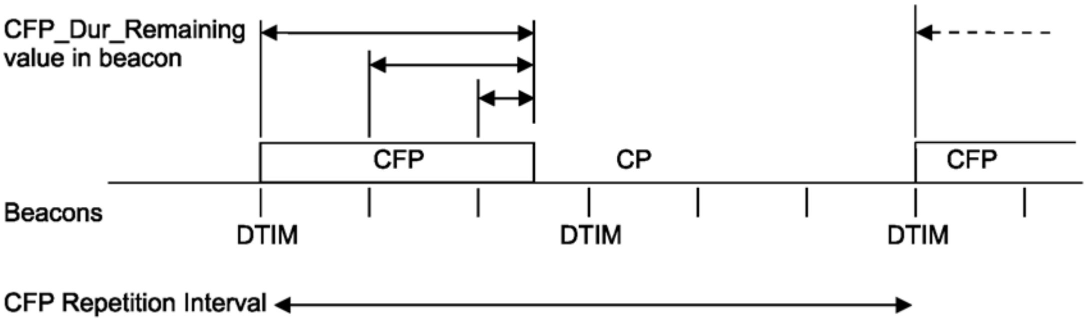


Figure 9-16—Beacon frames and CFPs

The length of the CFP is controlled by the PC, with maximum duration specified by dot11CFPMaxDuration. Neither the maximum duration nor the actual duration (signaled by transmission of a Control frame of subtype CF-End or CF-End+ACK by the PC) is constrained to be a multiple of the beacon interval. If the CFP duration is greater than the beacon interval, the PC shall transmit Beacon frames at the appropriate times during the CFP (subject to delay due to traffic at the nominal times, as with all Beacon frames). The CF Parameter Set element in all Beacon frames at the start of, or within, a CFP shall contain a nonzero value in the CFPDurRemaining field. This value, in units of TU, shall specify the maximum time from the most recent TBTT to the end of this CFP. The value of the CFPDurRemaining field shall be 0 in Beacon frames sent during the CP. An example of these relationships is illustrated in Figure 9-16, which shows a case where the CFPPeriod is two DTIM intervals, the DTIM interval is three beacon intervals, and the aCFPMaxDuration value is approximately 2.5 beacon intervals.

The PC may terminate any CFP at or before the limit given by dot11CFPMaxDuration, based on available traffic and size of the polling list. Because the transmission of any Beacon frame may be delayed due to a medium busy condition at the TBTT, a CFP may be shortened by the amount of the delay. In the case of a busy medium due to DCF traffic, the Beacon frame shall be delayed for the time required to complete the current DCF frame exchange. In cases where the Beacon frame transmission is delayed, the CFPDurRemaining value in the Beacon frame at the beginning of the CFP shall specify a time that causes the CFP to end no later than TBTT plus the value of aCFPMaxDuration. This is illustrated in Figure 9-17.

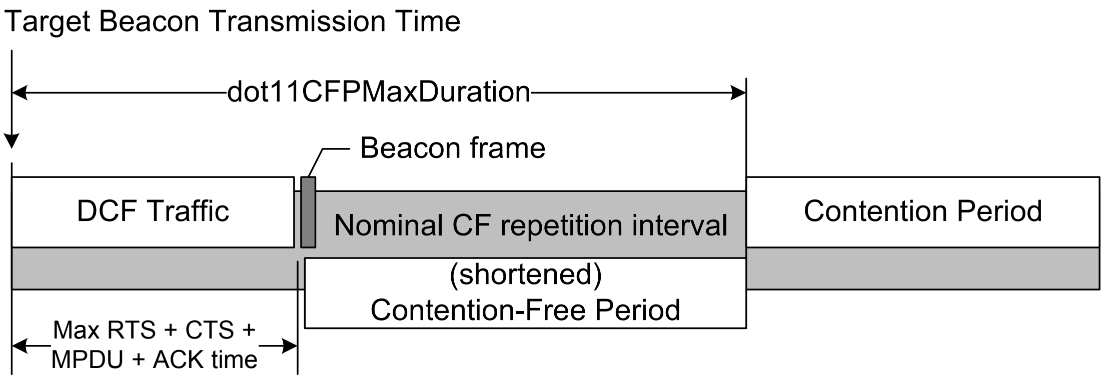


Figure 9-17—Example of delayed beacon and shortened CFP

9.4.3 PCF access procedure

9.4.3.1 General

The CF transfer protocol is based on a polling scheme controlled by a PC operating at the AP of the BSS. The PC gains control of the medium at the beginning of the CFP and attempts to maintain control for the entire CFP by waiting a shorter time between transmissions than the STAs using the DCF access procedure. All STAs that receive Beacon frames containing a CF Parameter Set element, including STAs not associated with the BSS, set their NAVs to the CFPMaxDuration value at the nominal start time of each CFP. This prevents most contention by preventing nonpolled transmissions by STAs regardless of whether they are CF-Pollable. Acknowledgment of frames sent during the CFP may be accomplished using Data+CF-ACK, CF-ACK, Data+CF-ACK+CF-Poll (only on frames transmitted by the PC), or CF-ACK+CF-Poll (only on frames transmitted by the PC) frames in cases where a Data (or Null) frame immediately follows the frame being acknowledged, thereby avoiding the overhead of separate ACK frames. Non-CF-Pollable or unpolled CF-Pollable STAs acknowledge frames during the CFP using the DCF ACK procedure.

9.4.3.2 Fundamental access

At the nominal beginning of each CFP, the PC shall sense the medium. When the medium is determined to be idle for one PIFS period, the PC shall transmit a Beacon frame containing the CF Parameter Set element and a DTIM element.

After the initial Beacon frame, the PC shall wait for one SIFS period, and then transmit one of the following: a data frame, a CF-Poll frame, a Data+CF-Poll frame, a management frame, or a CF-End frame. If the CFP is null, i.e., no traffic is buffered and no polls exist to send at the PC, a CF-End frame shall be transmitted immediately after the initial Beacon frame. If there are buffered group addressed MSDUs/MMPDUs, the PC shall transmit these prior to any individually addressed MSDUs/MMPDUs.

STAs receiving individually addressed, error-free frames from the PC are expected to respond after a SIFS period, in accordance with the transfer procedures defined in 9.4.4. If the recipient STA is not CF-Pollable, the response to receipt of an error-free data frame shall be an ACK frame.

9.4.3.3 NAV operation during the CFP

The mechanism for handling the NAV during the CFP is designed to facilitate the operation of overlapping CFP coordinated infrastructure BSSs. The mechanism by which infrastructure BSSs coordinate their CFPs is beyond the scope of this standard.

Each STA, except the STA with the PC, shall preset its NAV to the CFPMaxDuration value (obtained from the CF Parameter Set element in Beacon frames from this PC) at each TBTT (see Clause 10) at which a CFP is scheduled to start (based on the CFPCount field in the CF Parameter Set element of the Beacon frames from this PC). Each non-PC STA shall update its NAV using the CFPDurRemaining value in the CF Parameter Set element of any error-free Beacon frame that the STA receives. This includes CFPDurRemaining values in CF Parameter Set elements from Beacon frames received from other (overlapping) BSSs.

These actions prevent STAs from taking control of the medium during the CFP, which is especially important in cases where the CFP spans multiple medium-occupancy intervals, such as dwell periods of an FH PHY. This setting of the NAV also reduces the risk of hidden STAs determining the medium to be idle for a DIFS period during the CFP and possibly corrupting a transmission in progress.

A STA joining a BSS operating with a PC shall use the information in the CFPDurRemaining element of the CF parameter set of any received Beacon or Probe Response frames to update its NAV prior to initiating any transmissions.

The PC shall transmit a CF-End or CF-End+ACK frame at the end of each CFP. A STA that receives either of these frames, from any BSS, shall reset its NAV.

9.4.4 PCF transfer procedure

9.4.4.1 General

Frame transfers under the PCF may consist of frames alternately sent from the AP/PC and sent to the AP/PC. During the CFP, the ordering of these transmissions, and the STA allowed to transmit frames to the PC at any given point in time, shall be controlled by the PC. Figure 9-18 depicts frame transfer during a typical CFP. The rules under which this frame transfer takes place are detailed in 9.4.4.2 to 9.4.4.5.

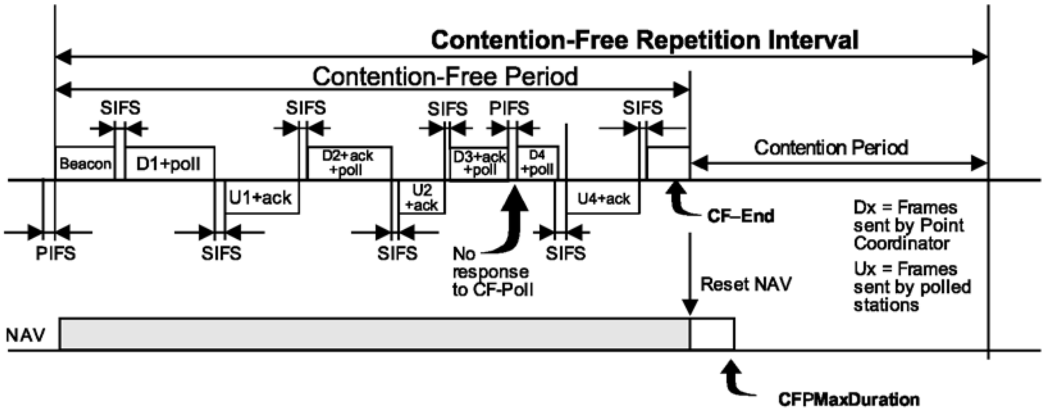


Figure 9-18—Example of PCF frame transfer

In a STA having an FH PHY, control of the channel is lost at a dwell time boundary. It is required that the current MPDU transmission and the accompanying acknowledgment of the MPDU be transmitted before the dwell time boundary. After having been polled by the PC, if there is not enough time remaining in the dwell to allow transmission of the MPDU plus the acknowledgment, the STA shall defer the transmission of the MPDU and shall transmit a Null frame or CF-ACK frame. The short retry counter and long retry counter for the MSDU shall not be affected.

MaxMPDUTime is the time to transmit the maximum-sized MAC frame, expanded by security mechanisms, plus the time to transmit the PHY preamble, header, trailer, and expansion bits, if any. In a STA having an FH

PHY, the PC shall not transmit a frame with any data subtype that includes CF-Poll to a STA if there is insufficient time remaining before the dwell boundary for the STA to respond with a Null frame or CF-ACK frame.

9.4.4.2 PCF transfers when the PC STA is transmitter or recipient

The PC shall transmit frames between the Beacon that starts the CFP and the CF-End using the SIFS except in cases where a transmission by another STA is expected by the PC and a SIFS period elapses without the receipt of the expected transmission. In such cases the PC may send its next pending transmission as soon as one PIFS after the end of its last transmission. This permits the PC to retain control of the medium in the presence of an overlapping BSS. The PC may transmit any of the following frame types to CF-Pollable STAs:

— Data, used to send data from the PC when the addressed recipient is not being polled and there is no previous frame to acknowledge;
— Data+CF-ACK, used to send data from the PC when the addressed recipient is not being polled or is not CF-Pollable or the DA is a group address and the PC needs to acknowledge the receipt of a frame received from a CF-Pollable STA a SIFS period before starting this transmission;
— Data+CF-Poll, used to send data from the PC when the addressed recipient is the next STA to be permitted to transmit during this CFP and there is no previous frame to acknowledge;
— Data+CF-ACK+CF-Poll, used to send data from the PC when the addressed recipient is the next STA to be permitted to transmit during this CFP and the PC needs to acknowledge the receipt of a frame received from a CF-Pollable STA a SIFS period before starting this transmission;
— CF-Poll, used when the PC is not sending data to the addressed recipient but the addressed recipient is the next STA to be permitted to transmit during this CFP and there is no previous frame to acknowledge;
— CF-ACK+CF-Poll, used when the PC is not sending data to the addressed recipient but the addressed recipient is the next STA to be permitted to transmit during this CFP and the PC needs to acknowledge the receipt of a frame from a CF-Pollable STA a SIFS period before starting this transmission;
— CF-ACK, used when the PC is not sending data to, or polling, the addressed recipient, but the PC needs to acknowledge receipt of a frame from a CF-Pollable STA a SIFS period before starting this transmission (useful when the next transmission by the PC is a management frame, such as a Beacon frame); or
— Any management frame that is appropriate for the AP to send under the rules for that frame type.

The PC may transmit data or management frames to non-CF-Pollable, non-PS STAs during the CFP. These STAs shall acknowledge receipt with ACK frames after a SIFS, as with the DCF. The PC may also transmit group addressed frames during the CFP. Because the Beacon frame that initiates the CFP contains a DTIM element, if there are associated STAs using PS mode, the buffered group addressed frames shall be sent immediately after any Beacon frame containing a TIM element with a DTIM count field with a value of 0.

A CF-Pollable STA that receives an individually addressed data frame of any subtype that includes CF-Poll may transmit one data frame a SIFS period after receiving the CF-Poll. CF-Pollable STAs shall ignore, but not reset, their NAV when performing transmissions in response to a CF-Poll.

Non-CF-Pollable STAs that receive an individually addressed frame during the CFP shall transmit an ACK, but shall not reset their NAV.

For frames that require MAC-level acknowledgment, CF-Pollable STAs that received a CF-Poll (of any type) may perform this acknowledgment using the Data+CF-ACK subtype in the response to the CF-Poll. For example, the U1 frame in Figure 9-18 contains the acknowledgment to the preceding D1 frame. The D2 frame contains the acknowledgment to the preceding U1 frame. The PC may use the CF-ACK subtypes to acknowledge a received frame even if the data frame sent with the CF-ACK subtype is addressed to a different

STA than the one being acknowledged. CF-Pollable STAs that are expecting an acknowledgment shall interpret the subtype of the frame (if any) sent by the PC a SIFS period after that STA's transmission to the PC. If a frame that requires MAC-level acknowledgment is received by a non-CF-Pollable STA, that STA shall not interpret the CF-Poll indication (if any), and shall acknowledge the frame by sending an ACK frame after a SIFS period.

The lengths of the frames may be variable, only bounded by the frame and/or fragment length limitations that apply for the BSS. If a CF-Pollable STA does not respond to a CF-Poll (of any type) within the SIFS period following a transmission from the PC, or a non-CF-Pollable STA does not return the ACK frame within a SIFS period following a transmission from the PC that requires acknowledgment, then the PC shall resume control and may transmit its next frame after a PIFS period from the end of the PC's last transmission.

A CF-Pollable STA shall respond to a frame with any data subtype that includes CF-Poll directed to its MAC address and received without error. If the STA has no frame to send when polled, the response shall be a Null frame. If the STA has no frame to send when polled, but an acknowledgment is required for the frame that conveyed the CF-Poll, the response shall be a CF-ACK (no data) frame. The null response is required to permit a "no-traffic" situation to be distinguished from a collision between overlapping PCs.

The CFP shall end when the CFPDurRemaining time has elapsed since the Beacon frame originating the CFP or when the PC has no further frames to transmit nor STAs to poll. In either case, the end of the CFP shall be signaled by the transmission of a CF-End by the PC. If there is a received frame that requires acknowledgment at the time the CF-End is to be transmitted, the PC shall transmit a CF-End+ACK frame instead. A STA receiving a CF-End or CF-End+ACK shall reset its NAV.

9.4.4.3 Operation with overlapping point-coordinated BSSs

Because the PCF operates without the CSMA/CA CW randomization and backoff of the DCF, there is a risk of repeated collisions if multiple, overlapping, point-coordinated BSSs are operating on the same PHY channel, and their CFP Rates and beacon intervals are approximately equal. To minimize the risk of significant frame loss due to CF collisions, the PC shall use a DIFS plus a random backoff delay (with CW in the range of 1 to aCWmin) to start a CFP when the initial Beacon frame is delayed because of deferral due to a busy medium. The PC may optionally use this backoff during the CFP prior to retransmitting an unacknowledged, individually addressed data or management frame.

To further reduce the susceptibility to inter-PC collisions, the PC shall require that the medium be determined as being idle for a DIFS period plus a random (over a range of 1 to aCWmin) number of slot times once every dot11MediumOccupancyLimit TU during the CFP. This results in loss of control of the medium to overlapping BSS or hidden STA traffic, because the STAs in this BSS are prevented from transmitting by their NAV setting to CFPMaxDuration or CFPDurRemaining. For operation of the PCF in conjunction with an FH PHY, dot11MediumOccupancyLimit shall be set equal to the dwell time. For operation in conjunction with other PHY types, dot11MediumOccupancyLimit may be set equal to CFPMaxDuration, unless extra protection against PCF collisions is desired. The dot11MediumOccupancyLimit is also useful for compliance in regulatory domains that impose limits on continuous transmission time by a single STA as part of a spectrum etiquette.

9.4.4.4 CFPMaxDuration limit

The value of CFPMaxDuration shall be limited to allow coexistence between contention and CF traffic.

The minimum value for CFPMaxDuration is two times MaxMPDUTime plus the time required to send the initial Beacon frame and the CF-End frame of the CFP. This may allow sufficient time for the AP to send one data frame to a STA, while polling that STA, and for the polled STA to respond with one data frame.

The maximum value for CFPMaxDuration is the duration of (BeaconPeriod × DTIMPeriod × CFPPeriod) minus [MaxMPDUTime plus (2 × aSIFSTime) plus (2 × aSlotTime) plus (8 × ACKSize)], expressed in microseconds. MaxMPDUTime is the time to transmit the maximum-sized MAC frame, expanded by security mechanisms, plus the time to transmit the PHY preamble, header, trailer, and expansion bits, if any. This allows sufficient time to send at least one data frame during the CP.

9.4.4.5 CF usage rules

A PC may send group addressed frames, and individually addressed data or management frames to any active STA, as well as to CF-Pollable PS STAs. During the CFP, CF-Pollable STAs shall acknowledge after a SIFS period, the receipt of each Data+CF-Poll frame or Data+CF-ACK+CF-Poll frame using Data+CF-Ack or CF-Ack (no data) frames, the receipt of each CF_Poll (no data) using Data or Null (no data), and the receipt of all other data and management frames using ACK frames. Non-CF-Pollable STAs shall acknowledge receipt of data and management frames using ACK frames sent after a SIFS period. This non-CF-Pollable operation is the same as that already employed by such STAs for DCF operation.

When polled by the PCF (Data+CF-Poll, Data+CF-ACK+CF-Poll, CF-Poll, or CF-ACK+CF-Poll) a CF-Pollable STA may send one data frame to any destination. Such a frame directed to or through the PC STA shall be acknowledged by the PC, using the CF-ACK indication (Data+CF-ACK, Data+CF-ACK+CF-Poll, CF-ACK, CF-ACK+CF-Poll, or CF-End+ACK) sent after a SIFS. Such a frame directed to a non-CF-Pollable STA shall be acknowledged using an ACK frame sent after a SIFS period. A polled CF-Pollable STA with neither a data frame nor an acknowledgment to send shall respond by transmitting a Null frame after a SIFS period. A polled CF-Pollable STA with insufficient time before the end of the CFP or current medium occupancy limit, to send its queued MPDU and receive an acknowledgment, shall respond by transmitting a Null frame, or a CF-ACK frame if polled using Data+CF-Poll or Data+CF-ACK+CF-Poll, after a SIFS period. The CF-Pollable STA may set the More Data bit to 1 in its response to permit the PC to distinguish between an empty STA queue and a response due to insufficient time to transfer an MPDU.

The PC shall not issue frames with a subtype that includes CF-Polls if insufficient time remains in the current CFP to permit the polled STA to transmit a data frame containing a minimum length MPDU.

9.4.5 CF polling list

9.4.5.1 General

If the PC supports use of the CFP for inbound frame transfer as well as for frame delivery, the PC shall maintain a “polling list” for use in selecting STAs that are eligible to receive CF-Polls during CFPs. The polling list functional characteristics are defined below. If the PC supports the use of the CFP solely for frame delivery, the PC does not require a polling list, and shall never generate data frames with a subtype that includes CF-Poll. The form of CF support provided by the PC is identified in the Capability Information field of Beacon, Association Response, Reassociation Response, and Probe Response management frames, which are sent from APs. Any such frames sent by STAs, as in noninfrastructure networks, shall have these bits set to 0.

The polling list is used to force the polling of CF-Pollable STAs, regardless of whether the PC has pending traffic to transmit to those STAs. The polling list may be used to control the use of Data+CF-Poll and Data+CF-ACK+CF-Poll types for transmission of data frames being sent to CF-Pollable STAs by the PC. The polling list is a *logical* construct, which is not exposed outside of the PC. A minimum set of polling list maintenance techniques are required to in order to provide interoperability of arbitrary CF-Pollable STAs in BSSs controlled by arbitrary APs with active PCs. APs may also implement additional polling list maintenance techniques that are outside the scope of this standard.

9.4.5.2 Polling list processing

The PC shall send a CF-Poll to at least one STA during each CFP when there are entries in the polling list. During each CFP, the PC shall issue polls to a subset of the STAs on the polling list in order by ascending AID value.

While time remains in the CFP, all CF frames have been delivered, and all STAs on the polling list have been polled, the PC may generate one or more CF-Polls to any STAs on the polling list. While time remains in the CFP, all CF frames have been delivered, and all STAs on the polling list have been polled, the PC may send data or management frames to any STAs.

In order to gain maximum efficiency from the CFP, and the ability to piggyback acknowledgments on successor data frames in the opposite direction, the PC should generally use Data+CF-Poll and Data+CF-ACK+CF-Poll types for each data frame transmitted while sufficient time for the potential response to the CF-Poll remains in the CFP.

9.4.5.3 Polling list update procedure

A STA indicates its CF-Pollability using the CF-Pollable subfield of the Capability Information field of Association Request and Reassociation Request frames.

NOTE—A STA might perform a reassociation in order to change the PC's record of its CF-Pollability.

During association, a CF-Pollable STA may request to be placed on the polling list, or to never be polled, by appropriate use of bits in the Capability Information field of the Associate Request or Reassociate Request frame, as shown in Table 8-35 (see 8.4.1.4).

CF-Pollable STAs that are not on the polling list, but did not request never to be polled during their most recent association, may be dynamically placed on the polling list by the PC to handle bursts of frame transfer activity by that STA.

9.5 Fragmentation

The MAC may fragment and reassemble individually addressed MSDUs or MMPDUs. The fragmentation and defragmentation mechanisms allow for fragment retransmission.

The length of each fragment shall be an equal number of octets for all fragments except the last, which may be smaller. The length of each fragment shall be an even number of octets, except for the last fragment of an MSDU or MMPDU, which may be either an even or an odd number of octets. The length of a fragment shall never be larger than dot11FragmentationThreshold unless security encapsulation is invoked for the MPDU. If security encapsulation is active for the MPDU, then the MPDU shall be expanded by the encapsulation overhead and this may result in a fragment larger than dot11FragmentationThreshold.

A fragment is an MPDU, the payload of which carries all or a portion of an MSDU or MMPDU. When data are to be transmitted, the number of octets in the fragment (before processing by the security mechanism) shall be determined by dot11FragmentationThreshold and the number of octets in the MPDU that have yet to be assigned to a fragment at the instant the fragment is constructed for the first time. Once a fragment is transmitted for the first time, its frame body content and length shall be fixed until it is successfully delivered to the immediate receiving STA. A STA shall be capable of receiving fragments of arbitrary length that is less than the maximum allowed MSDU size, plus any encapsulation headers.

If a fragment requires retransmission, its frame body content and length shall remain fixed for the lifetime of the MSDU or MMPDU at that STA. After a fragment is transmitted once, contents and length of that fragment are not allowed to fluctuate to accommodate the dwell time boundaries. Each fragment shall contain a

Sequence Control field, which is comprised of a sequence number and fragment number. When a STA is transmitting an MSDU or MMPDU, the sequence number shall remain the same for all fragments of that MSDU or MMPDU. The fragments shall be sent in order of lowest fragment number to highest fragment number, where the fragment number value starts at 0, and increases by 1 for each successive fragment. The Frame Control field also contains a bit, the More Fragments bit, that is equal to 0 to indicate the last (or only) fragment of the MSDU or MMPDU.

The source STA shall maintain a transmit MSDU timer for each MSDU being transmitted. The attribute dot11MaxTransmitMSDULifetime specifies the maximum amount of time allowed to transmit an MSDU. The timer starts on the initial attempt to transmit the first fragment of the MSDU. If the timer exceeds dot11MaxTransmitMSDULifetime, then all remaining fragments are discarded by the source STA and no attempt is made to complete transmission of the MSDU.

9.6 Defragmentation

Each fragment contains information to allow the complete MSDU or MMPDU to be reassembled from its constituent fragments. The header of each fragment contains the following information that is used by the destination STA to reassemble the MSDU or MMPDU:

- Frame type
- Address of the sender, obtained from the Address2 field
- Destination address
- *Sequence Control field:* This field allows the destination STA to check that all incoming fragments belong to the same MSDU or MMPDU, and the sequence in which the fragments should be reassembled. The sequence number within the Sequence Control field remains the same for all fragments of an MSDU or MMPDU, while the fragment number within the Sequence Control field increments for each fragment.
- *More Fragments indicator:* Indicates to the destination STA that this is not the last fragment of the MSDU or MMPDU. Only the last or sole fragment of the MSDU or MMPDU shall have this bit set to 0. All other fragments of the MSDU or MMPDU shall have this bit set to 1.

The destination STA shall reconstruct the MSDU or MMPDU by combining the fragments in order of fragment number subfield of the Sequence Control field. If security encapsulation has been applied to the fragment, it shall be deencapsulated and decrypted before the fragment is used for defragmentation of the MSDU or MMPDU. If the fragment with the More Fragments bit equal to 0 has not yet been received, then the destination STA knows that the MSDU or MMPDU is not yet complete. As soon as the STA receives the fragment with the More Fragments bit equal to 0, the STA knows that no more fragments may be received for the MSDU or MMPDU.

All STAs shall support the concurrent reception of fragments of at least three MSDUs or MMPDUs. Note that a STA receiving more than three fragmented MSDUs or MMPDUs concurrently may experience a significant increase in the number of frames discarded.

The destination STA shall maintain a Receive Timer for each MSDU or MMPDU being received, for a minimum of three MSDUs or MMPDUs. The STA may implement additional timers to be able to receive additional concurrent MSDUs or MMPDUs. The receiving STA shall discard all fragments that are part of an MSDU or MMPDU for which a timer is not maintained. There is also dot11MaxReceiveLifetime, that specifies the maximum amount of time allowed to receive an MSDU. The receive MSDU or MMPDU timer starts on the reception of the first fragment of the MSDU or MMPDU. If the receive MSDU timer exceeds dot11MaxReceiveLifetime, then all received fragments of this MSDU or MMPDU are discarded by the destination STA. If additional fragments of an individually addressed MSDU or MMPDU are received after its dot11MaxReceiveLifetime is exceeded, those fragments shall be acknowledged and discarded.

To properly reassemble MPDUs into an MSDU or MMPDU, a destination STA shall discard any duplicated fragments received. A STA shall discard duplicate fragments as described in 9.3.2.10. However, an acknowledgment shall be sent in response to a duplicate fragment of an individually addressed MSDU.

9.7 Multirate support

9.7.1 Overview

Some PHYs have multiple data transfer rate capabilities that allow implementations to perform dynamic rate switching with the objective of improving performance. The algorithm for performing rate switching is beyond the scope of this standard, but in order to provide coexistence and interoperability on multirate-capable PHYs, this standard defines a set of rules to be followed by all STAs.

Only the data transfer rates of the mandatory rate set of the attached PHY are guaranteed to be supported when a STA for which dot11OCBActivated is true transmits a management or data frame. Higher layer protocols may negotiate a rate outside the mandatory rate set.

A STA that transmits a frame shall select a rate defined by the rules for determining the rates of transmission of protection frames in 9.23 when the following conditions apply:

— The STA’s protection mechanism for non-ERP receivers is enabled.

— The frame is a protection mechanism frame.

— The frame initiates an exchange.

Otherwise, the frame shall be transmitted using a rate that is in accordance with rules defined in 9.7.5 and 9.7.6.

For the Clause 18, Clause 17, Clause 19, and Clause 20 PHYs, the time required to transmit a frame for use in calculating the value for the Duration/ID field is determined using the PLME-TXTIME.request primitive (see 6.5.7) and the PLME-TXTIME.confirm primitive (see 6.5.8), both defined in 18.4.3, 17.3.4, 19.8.3.2, 19.8.3.3, 19.8.3.4, or 20.4.3 depending on the PHY options. In QoS STAs, the Duration/ID field may cover multiple frames and may involve using the PLME-TXTIME.request primitive several times.

9.7.2 Basic MCS Set field

An AP that transmits a frame containing an HT Operation element with either the Dual Beacon field or the Dual CTS Protection field equal to 1 shall include at least one MCS that has only one spatial stream in the Basic MCS Set field of the HT Operation element of that frame.

9.7.3 Basic STBC MCS

The basic STBC MCS has the value null when any of the following conditions is true:

— The Dual Beacon field in the HT Operation element is equal to 0, and the Dual CTS Protection field in the HT Operation element is equal to 0.

— No HT Operation element is present in the most recently received Association Response frame that was addressed to this STA.

— The BSSBasicMCSSet is empty or does not exist.

— The lowest MCS of the BSSBasicMCSSet has NSS value greater than 1 (the mapping of MCS to NSS is PHY dependent, for the HT PHY see 20.6).

If none of the above conditions is true, then the basic STBC MCS is the lowest MCS index of the BSSBasicMCSSet parameter.

When an MCS from the basic STBC MCS is required in 9.7.5 and 9.7.6 but the basic STBC MCS has the value null, the STA shall select a mandatory MCS of the attached PHY.

9.7.4 Basic Rate Set and Basic MCS Set for mesh STA

A mesh STA shall not establish a mesh peering with a mesh STA using a different BSSBasicRateSet (see 13.2.7 and 13.2.8).

Mesh STAs should adopt the mandatory PHY rates as the default BSSBasicRateSet to reduce the risk that a candidate peer mesh STA utilizes a different BSSBasicRateSet. If the mesh STA is also an HT STA, it should adopt the MCSs of mandatory MCSs as the default BSSBasicMCSSet.

Once the mesh STA establishes a mesh peering with a mesh STA, it shall change neither the BSSBasicRateSet nor the BSSBasicMCSSet parameters.

9.7.5 Rate selection for data and management frames

9.7.5.1 Rate selection for non-STBC Beacon and non-STBC PSMP frames

If the BSSBasicRateSet parameter is not empty, a non-STBC PSMP frame that is not part of an FMS stream or a non-STBC Beacon shall be transmitted in a non-HT PPDU using one of the rates included in the BSSBasicRateSet parameter.

If the BSSBasicRateSet parameter is empty, the frame shall be transmitted in a non-HT PPDU using one of the mandatory PHY rates.

All non-STBC PSMP frames with a group address in the Address 1 field that are part of an FMS stream shall be transmitted using the rate chosen by the AP as described in 10.23.7.

9.7.5.2 Rate selection for STBC group addressed data and management frames

When a STA has dot11TxSTBCOptionActivated true, it shall use the basic STBC MCS when it transmits an STBC Beacon frame or when it transmits a group addressed data or management frame that is an STBC frame.

9.7.5.3 Rate selection for other group addressed data and management frames

This subclause describes the rate selection rules for group addressed data and management frames, excluding the following:

- Non-STBC Beacon and non-STBC PSMP frames
- STBC group addressed data and management frames
- Data frames located in an FMS stream (see 10.23.7)

If the BSSBasicRateSet parameter is not empty, a data or management frame (excluding the frames listed above) with a group address in the Address 1 field shall be transmitted in a non-HT PPDU using one of the rates included in the BSSBasicRateSet parameter or the rate chosen by the AP, described in 10.23.7, if the data frames are part of an FMS stream.

If the BSSBasicRateSet parameter is empty and the BSSBasicMCSSet parameter is not empty, the frame shall be transmitted in an HT PPDU using one of the MCSs included in the BSSBasicMCSSet parameter.

If both the BSSBasicRateSet parameter and the BSSBasicMCSSet parameter are empty (e.g., a scanning STA that is not yet associated with a BSS), the frame shall be transmitted in a non-HT PPDU using one of the mandatory PHY rates.

9.7.5.4 Rate selection for polling frames

A data frame of a subtype that includes CF-Poll that does not also include CF-Ack and that is sent in the CP shall be transmitted at a rate selected as follows:

a) If an initial exchange has already established protection and the Duration/ID field in the frame establishing protection covers the entire TXOP, the rate or MCS is selected according to the rules in 9.7.5.6.
b) Otherwise, the data frame shall be transmitted at a rate or MCS as defined in 9.7.5.3, treating the frame as though it has a group address in the Address 1 field, solely for the purpose of determining the appropriate rate or MCS.

9.7.5.5 Rate selection for +CF-Ack frames

For a frame of type (QoS) Data+CF-Ack, (QoS) Data+CF-Poll+CFAck, or (QoS) CF-Poll+CF-Ack, the rate or MCS and TXVECTOR parameter CH_BANDWIDTH used to transmit the frame shall be chosen from among those supported by both the addressed recipient STA and the STA to which the ACK frame is intended.

9.7.5.6 Rate selection for other data and management frames

A data or management frame not identified in 9.7.5.1 through 9.7.5.5 shall be sent using any data rate or MCS subject to the following constraints:

— A STA shall not transmit a frame using a rate or MCS that is not supported by the receiver STA or STAs, as reported in any Supported Rates element, Extended Supported Rates element, or Supported MCS field in management frames transmitted by the receiver STA.
— A STA shall not transmit a frame using a value for the CH_BANDWIDTH parameter of the TXVECTOR that is not supported by the receiver STA.
— A STA shall not initiate transmission of a frame at a data rate higher than the greatest rate in the OperationalRateSet or the HTOperationalMCSset, which are parameters of the MLME-JOIN.request primitive.

When the supported rate set of the receiving STA or STAs is not known, the transmitting STA shall transmit using a rate in the BSSBasicRateSet parameter, or an MCS in the BSSBasicMCSSet parameter, or a rate from the mandatory rate set of the attached PHY if both the BSSBasicRateSet and the BSSBasicMCSSet are empty.

The rules in this subclause also apply to A-MPDUs that aggregate MPDUs of type Data or Management with any other types of MPDU.

9.7.6 Rate selection for control frames

9.7.6.1 General rules for rate selection for control frames

Control frames carried in an A-MPDU shall be sent at a rate selected from the rules defined in 9.7.5.6.

NOTE—The rules defined in 9.7.6.2 through 9.7.6.5 apply only to control frames not carried in an A-MPDU.

The following rules determine whether a control frame is carried in an HT PPDU or non-HT PPDU:

a) A control frame shall be carried in an HT PPDU when the control frame meets any of the following conditions:
    1) The control frame contains an L-SIG duration value (see 9.23.5), or

    2) The control frame is sent using an STBC frame.

b) A control response frame shall be carried in an HT PPDU when the control frame is a response to a frame that meets any of the following conditions:
    1) The frame eliciting the response included an HT Control field with the TRQ field equal to 1 and the NDP Announcement subfield equal to 0, and this responder set the Implicit Transmit Beamforming Receiving Capable field to 1 in its last transmitted HT Capabilities element; or
    2) The frame eliciting the response was an RTS frame carried in an HT PPDU; or
    3) The frame eliciting the response was an STBC frame, and the Dual CTS Protection field was equal to 1 in the last HT Operation element received from its AP or transmitted by the STA (see 9.3.2.7).

c) A control frame may be carried in an HT PPDU when the control frame meets any of the following conditions:
    1) The control frame contains an HT Control field with the MRQ subfield equal to 1, or
    2) The control frame contains an HT Control field with the TRQ field equal to 1.

    NOTE—In these cases, requirements specified in 9.27, 9.28.2, and 9.29 further constrain the choice of non-HT or HT PPDU.

d) Otherwise, the control frame shall be carried in a non-HT PPDU.

Selection of channel width is defined in 9.7.6.6.

A control response frame is a control frame that is transmitted as a response to the reception of a frame a SIFS time after the PPDU containing the frame that elicited the response, e.g. a CTS in response to an RTS reception, an ACK in response to a DATA reception, a BlockAck in response to a BlockAckReq reception. In some situations, the transmission of a control frame is not a control response transmission, such as when a CTS is used to initiate a TXOP.

9.7.6.2 Rate selection for control frames that initiate a TXOP

This subclause describes the rate selection rules for control frames that initiate a TXOP and that are not carried in an A-MPDU.

If a control frame other than a Basic BlockAckReq or Basic BlockAck is carried in a non-HT PPDU, the transmitting STA shall transmit the frame using one of the rates in the BSSBasicRateSet parameter or a rate from the mandatory rate set of the attached PHY if the BSSBasicRateSet is empty.

If a Basic BlockAckReq or Basic BlockAck frame is carried in a non-HT PPDU, the transmitting STA shall transmit the frame using a rate supported by the receiver STA, if known (as reported in the Supported Rates element and/or Extended Supported Rates element in frames transmitted by that STA). If the supported rate set of the receiving STA or STAs is not known, the transmitting STA shall transmit using a rate from the BSSBasicRateSet parameter or using a rate from the mandatory rate set of the attached PHY if the BSSBasicRateSet is empty.

NOTE—Because of their utility in resolving contention and in establishing a NAV, most control subtype frames that initiate a frame exchange are subject to explicit limitations regarding the choice of transmission rate with the intent of ensuring maximum possible coverage and receivability of the frame. But the Basic BlockAckReq and Basic BlockAck frames are subject to fewer restrictions because their use at times mimics a typical data-ACK exchange, where no BSS BasicRateSet rate restriction exists on the data frame. In addition, the Basic BlockAck frame is significantly larger than the other control frames.

When L-SIG TXOP protection is not used for an HT PPDU, an HT STA shall select an MCS from the BSSBasicMCSSet parameter when protection is required (as defined in 9.23) and shall select an MCS from the SupportedMCSSet parameter of the intended receiver when protection is not required.

When L-SIG TXOP protection is used, an HT STA shall select an MCS from the SupportedMCSSet parameter of the intended receiver.

9.7.6.3 Rate selection for CF_End frames

If not operating during the 40 MHz phase of PCO, a STA that transmits a CF-End frame that is not at the end of a TXOP that was obtained through the use of the dual CTS mechanism shall transmit the frame using a rate in BSSBasicRateSet or from the mandatory rate set of the attached PHY if the BSSBasicRateSet is empty.

If operating during the 40 MHz phase of PCO, a STA that transmits a CF-End frame that is not at the end of a TXOP that was obtained through the use of the dual CTS mechanism shall transmit the frame using an MCS from the BSSBasicMCSSet parameter.

A STA that transmits a CF-End frame at the end of a TXOP that was obtained by a non-AP STA through the use of the dual CTS mechanism shall transmit the CF-End frame with the same value for the TXVECTOR parameter STBC, TXVECTOR parameter MCS (if present), and TXVECTOR parameter RATE as was used for the transmission of the matching control frame at the beginning of the TXOP. The matching control frame is defined as follows:

- For the first CF-End transmitted in the TXOP, the matching control frame is the first RTS transmitted in the TXOP.
- For the second CF-End transmitted in the TXOP, the matching control frame is the first CTS that follows the first RTS transmitted in the TXOP.
- For the third CF-End transmitted in the TXOP, the matching control frame is the second CTS that follows the first RTS transmitted in the TXOP.

A STA that transmits a CF-End frame at the end of a TXOP that was obtained by an AP through the use of the dual CTS mechanism shall transmit the CF-End frame with the same value for the TXVECTOR parameter STBC, TXVECTOR parameter MCS (if present), and TXVECTOR parameter RATE as was used for the transmission of the matching control frame at the beginning of the TXOP. The matching control frame is defined as follows:

- For the first CF-End transmitted in the TXOP, the matching control frame is the first CTS-to-self transmitted in the TXOP.
- For the second CF-End transmitted in the TXOP, the matching control frame is the first RTS transmitted in the TXOP.

9.7.6.4 Rate selection for control frames that are not control response frames

This subclause describes the rate selection rules for control frames that are not control response frames, are not the frame that initiates a TXOP, are not the frame that terminates a TXOP, and are not carried in an A-MPDU.

A frame other than a BlockAckReq or BlockAck that is carried in a non-HT PPDU shall be transmitted by the STA using a rate no higher than the highest rate in the BSSBasicRateSet parameter that is less than or equal to the rate or non-HT reference rate (see 9.7.9) of the previously transmitted frame that was directed to the same receiving STA. If no rate in the BSSBasicRateSet parameter meets these conditions, the control frame shall be transmitted at a rate no higher than the highest mandatory rate of the attached PHY that is less than or equal to the rate or non-HT reference rate (see 9.7.9) of the previously transmitted frame that was directed to the same receiving STA.

A BlockAckReq or BlockAck that is carried in a non-HT PPDU shall be transmitted by the STA using a rate supported by the receiver STA, as reported in the Supported Rates element and/or Extended Supported Rates element in frames transmitted by that STA. When the supported rate set of the receiving STA or STAs is not known, the transmitting STA shall transmit using a rate from the BSSBasicRateSet parameter or from the mandatory rate set of the attached PHY if the BSSBasicRateSet is empty.

A frame that is carried in an HT PPDU shall be transmitted by the STA using an MCS supported by the receiver STA, as reported in the Supported MCS field in the HT Capabilities element in management frames transmitted by that STA. When the supported rate set of the receiving STA or STAs is not known, the transmitting STA shall transmit using an MCS in the BSSBasicMCSSet parameter.

9.7.6.5 Rate selection for control response frames

9.7.6.5.1 Introduction

Subclauses 9.7.6.5.2 through 9.7.6.5.5 describe the rate selection rules for control response frames that are not carried in an A-MPDU.

9.7.6.5.2 Selection of a rate or MCS

To allow the transmitting STA to calculate the contents of the Duration/ID field, a STA responding to a received frame transmits its control response frame at a primary rate, or at an alternate rate, or at an MCS, as specified by the following rules:

- If a CTS or ACK control response frame is carried in a non-HT PPDU, the primary rate is defined to be the highest rate in the BSSBasicRateSet parameter that is less than or equal to the rate (or non-HT reference rate; see 9.7.9) of the previous frame. If no rate in the BSSBasicRateSet parameter meets these conditions, the primary rate is defined to be the highest mandatory rate of the attached PHY that is less than or equal to the rate (or non-HT reference rate; see 9.7.9) of the previous frame. The STA may select an alternate rate according to the rules in 9.7.6.5.4. The STA shall transmit the non-HT PPDU CTS or ACK control response frame at either the primary rate or the alternate rate, if one exists.
- If a BlockAck frame is sent as an immediate response to either an implicit BlockAck request or to a BlockAckReq frame that was carried in an HT PPDU and the BlockAck frame is carried in a non-HT PPDU, the primary rate is defined to be the highest rate in the BSSBasicRateSet parameter that is less than or equal to the rate (or non-HT reference rate; see 9.7.9) of the previous frame. If no rate in the BSSBasicRateSet parameter meets these conditions, the primary rate is defined to be the highest mandatory rate of the attached PHY that is less than or equal to the rate (or non-HT reference rate; see 9.7.9) of the previous frame. The STA may select an alternate rate according to the rules in 9.7.6.5.4. The STA shall transmit the non-HT PPDU BlockAck control response frame at either the primary rate or the alternate rate, if one exists.
- If a Basic BlockAck frame is sent as an immediate response to a BlockAckReq frame that was carried in a non-HT PPDU and the Basic BlockAck frame is carried in a non-HT PPDU, the primary rate is defined to be the same rate and modulation class as the BlockAckReq frame, and the STA shall transmit the Basic BlockAck frame at the primary rate.
- If a Compressed BlockAck frame is sent as an immediate response to a BlockAckReq frame that was carried in a non-HT PPDU and the Compressed BlockAck frame is carried in a non-HT PPDU, the primary rate is defined to be the highest rate in the BSSBasicRateSet parameter that is less than or equal to the rate (or non-HT reference rate; see 9.7.9) of the previous frame. If no rate in the BSSBasicRateSet parameter meets these conditions, the primary rate is defined to be the highest mandatory rate of the attached PHY that is less than or equal to the rate (or non-HT reference rate; see 9.7.9) of the previous frame. The STA may select an alternate rate according to the rules in 9.7.6.5.4. The STA shall transmit the non-HT PPDU Compressed BlockAck control response frame at either the primary rate or the alternate rate, if one exists.
- If the control response frame is carried in an HT PPDU, then it is transmitted at an MCS as determined by the procedure defined in 9.7.6.5.3.

The modulation class of the control response frame shall be selected according to the following rules:

- If the received frame is of a modulation class other than HT and the control response frame is carried in a non-HT PPDU, the control response frame shall be transmitted using the same modulation class as the received frame. In addition, the control response frame shall be sent using the same value for the TXVECTOR parameter PREAMBLE_TYPE as the received frame.
- If the received frame is of the modulation class HT and the control response frame is carried in a non-HT PPDU, the control response frame shall be transmitted using one of the ERP-OFDM or OFDM modulation classes.
- If the control response frame is carried in an HT PPDU, the modulation class shall be HT.

The selection of the value for the channel width (CH_BANDWIDTH parameter of the TXVECTOR) of the response transmission is defined in 9.7.6.6.

9.7.6.5.3 Control response frame MCS computation

If a control response frame is to be transmitted within an HT PPDU, the channel width (CH_BANDWIDTH parameter of the TXVECTOR) shall be selected first according to 9.7.6.6, and then the MCS shall be selected from a set of MCSs called the *CandidateMCSSet* as described in this subclause.

The Rx Supported MCS Set of the STA that transmitted the frame eliciting the response is determined from its Supported MCS Set field as follows:

- If a bit in the Rx MCS Bitmask subfield is equal to 0, the corresponding MCS is not supported.
- If a bit in the Rx MCS Bitmask subfield is equal to 1 and the integer part of the data rate (expressed in megabits per second) of the corresponding MCS is less than or equal to the rate represented by the Rx Highest Supported Data Rate subfield, then the MCS is supported by the STA on receive. If the Rx Highest Supported Data Rate subfield is equal to 0 and a bit in the Rx MCS Bitmask is equal to 1, then the corresponding MCS is supported by the STA on receive.

The CandidateMCSSet is determined using the following rules:

- If the frame eliciting the response was an STBC frame and the Dual CTS Protection bit is equal to 1, the CandidateMCSSet shall contain only the basic STBC MCS.
- If the frame eliciting the response had an L-SIG duration value (see 9.23.5) and initiates a TXOP, the CandidateMCSSet is the MCS Set consisting of the intersection of the Rx Supported MCS Set of the STA that sent the frame that is eliciting the response and the set of MCSs that the responding STA is capable of transmitting.
- If none of the above conditions is true, the CandidateMCSSet is the BSSBasicMCSSet parameter. If the BSSBasicMCSSet parameter is empty, the CandidateMCSSet shall consist of the set of mandatory HT PHY MCSs.

MCS values from the CandidateMCSSet that cannot be transmitted with the selected CH_BANDWIDTH parameter value shall be eliminated from the CandidateMCSSet.

The choice of a response MCS is made as follows:

a) If the frame eliciting the response is within a non-HT PPDU,
   1) Eliminate from the CandidateMCSSet all MCSs that have a data rate greater than the data rate of the received PPDU (the mapping of MCS to data rate is defined in 20.6).
   2) Find the highest indexed MCS from the CandidateMCSSet. The index of this MCS is the index of the MCS that is the primary MCS for the response transmission.
   3) If the CandidateMCSSet is empty, the primary MCS is the lowest indexed MCS of the mandatory MCSs.

b) If the frame eliciting the response is within an HT PPDU,

1) Eliminate from the CandidateMCSSet all MCSs that have an index that is higher than the index of the MCS of the received frame.

2) Determine the highest number of spatial streams (N_{SS}) value of the MCSs in the CandidateMCSSet that is less than or equal to the N_{SS} value of the MCS of the received frame. Eliminate all MCSs from the CandidateMCSSet that have an N_{SS} value that is not equal to this N_{SS} value. The mapping from MCS to N_{SS} is dependent on the attached PHY. For the HT PHY, see 20.6.

3) Find the highest indexed MCS of the CandidateMCSSet for which the modulation value of each stream is less than or equal to the modulation value of each stream of the MCS of the received frame and for which the coding rate value is less than or equal to the coding rate value of the MCS from the received frame. The index of this MCS is the index of the MCS that is the primary MCS for the response transmission. The mapping from MCS to modulation and coding rate is dependent on the attached PHY. For the HT PHY, see 20.6. For the purpose of comparing modulation values, the following sequence shows increasing modulation values: BPSK, QPSK, 16-QAM, 64-QAM.

4) If no MCS meets the condition in step 3), remove each MCS from the CandidateMCSSet that has the highest value of N_{SS} in the CandidateMCSSet. If the resulting CandidateMCSSet is empty, then set the CandidateMCSSet to the HT PHY mandatory MCSs. Repeat step 3) using the modified CandidateMCSSet.

Once the primary MCS has been selected, the STA may select an alternate MCS according to 9.7.6.5.4. The STA shall transmit the HT PPDU control response frame using either the primary MCS or the alternate MCS, if one exists.

9.7.6.5.4 Selection of an alternate rate or MCS for a control response frame

An alternate rate may be selected provided that all of the following conditions are met:

— The duration of frame at the alternate rate is the same as the duration of the frame at the primary rate determined by 9.7.6.5.2.

— The alternate rate is in either the BSSBasicRateSet parameter or is a mandatory rate of the attached PHY.

— The modulation class of the frame at the alternate rate is the same class as that of the primary rate selected by 9.7.6.5.2.

An alternate MCS may be selected provided that both of the following conditions are met:

— The duration of the frame at the alternate MCS is the same as the duration of the frame at the primary MCS.

— The alternate MCS is in the CandidateMCSSet that was generated according to the procedure of 9.7.6.5.3.

9.7.6.5.5 Control response frame TXVECTOR parameter restrictions

A STA shall not transmit a control response frame with TXVECTOR parameter GI_TYPE set to SHORT_GI unless it is in response to a reception of a frame with the RXVECTOR parameter GI_TYPE equal to SHORT_GI.

A STA shall not transmit a control response frame with TXVECTOR parameter FEC_CODING set to LDPC_CODING unless it is in response to a reception of a frame with the RXVECTOR parameter FEC_CODING equal to LDPC_CODING.

A STA shall not transmit a control response frame with the TXVECTOR parameter FORMAT set to HT_GF.

9.7.6.6 Channel Width selection for control frames

An HT STA that receives a frame that elicits a control frame transmission shall send the control frame response using a value for the CH_BANDWIDTH parameter that is based on the CH_BANDWIDTH parameter value of the received frame according to Table 9-3.

Table 9-3—CH_BANDWIDTH control frame response mapping

CH_BANDWIDTH RXVECTOR value	CH_BANDWIDTH TXVECTOR value
HT_CBW20	HT_CBW20 or NON_HT_CBW20
HT_CBW40	HT_CBW40 or NON_HT_CBW40
NON_HT_CBW20	HT_CBW20 or NON_HT_CBW20
NON_HT_CBW40	HT_CBW40 or NON_HT_CBW40

NOTE—This rule, combined with the rules in 9.7.6.1, determines the format of control response frames.

A frame that is intended to provide protection is transmitted using a channel width selected by the rules defined in 9.23.

An HT STA that uses a non-HT duplicate frame to establish protection of its TXOP shall send any CF-End frame using a non-HT duplicate frame except during the 40 MHz phase of PCO operation. During the 40 MHz phase of PCO operation, the rules in 10.16 apply.

9.7.6.7 Control frame TXVECTOR parameter restrictions

A STA shall not transmit a control frame that initiates a TXOP with the TXVECTOR parameter GI_TYPE set to a value of SHORT_GI.

A STA shall not transmit a control frame that initiates a TXOP with the TXVECTOR parameter FEC_CODING set to a value of LDPC_CODING.

9.7.7 Multiple BSSID Rate Selection

If the multiple BSSID capability is supported, Beacon frames shall be transmitted using any basic rate valid for all of the BSSs supported. If no such rate exists, then Beacon frames shall be transmitted using any mandatory PHY rate for any PHY type that all BSSs have in common.

9.7.8 Modulation classes

In order to determine the rules for response frames given in 9.7, the following modulation classes are defined in Table 9-4. Each row defines a modulation class. Modulations described within the same row have the same modulation class, while modulations described in different rows have different modulation classes. For Clause 20 PHY transmissions, the modulation class is determined by the FORMAT and NON_HT_MODULATION parameters of the TXVECTOR/RXVECTOR. Otherwise, the modulation class is determined by the clause or subclause number defining that modulation.

Table 9-4—Modulation classes

Modulation class	Description of modulation	Condition that selects this modulation class	
		Clause 14 to Clause 19 PHYs	Clause 20 PHY
1	Infrared (IR)	Clause 15 transmission	N/A
2	Frequency-hopping spread spectrum (FHSS)	Clause 14 transmission	N/A
3	DSSS and HR/DSSS	Clause 16 or Clause 17 transmission	FORMAT is NON_HT. NON_HT_MODULATION is ERP-DSSS or ERP-CCK.
4	ERP-PBCC	19.6 transmission	FORMAT is NON_HT. NON_HT_MODULATION is ERP-PBCC.
5	DSSS-OFDM The use of the DSSS-OFDM option is deprecated, and this option may be removed in a later revision of the standard.	19.7 transmission	FORMAT is NON_HT. NON_HT_MODULATION is DSSS-OFDM.
6	ERP-OFDM	19.5 transmission	FORMAT is NON_HT. NON_HT_MODULATION is ERP-OFDM.
7	OFDM	Clause 18 transmission	FORMAT is NON_HT. NON_HT_MODULATION is OFDM or NON_HT_DUP_OFDM.
8	HT	N/A	FORMAT is HT_MF or HT_GF.

9.7.9 Non-HT basic rate calculation

This subclause defines how to convert an HT MCS to a non-HT basic rate for the purpose of determining the rate of the response frame. It consists of two steps as follows:

a) Use the modulation and coding rate determined from the HT MCS (defined in 20.6) to a non-HT reference rate by lookup into Table 9-5.[27] In the case of an MCS with UEQM, the modulation of stream 1 is used.

b) The non-HT basic rate is the highest rate in the BSSBasicRateSet that is less than or equal to this non-HT reference rate.

[27] For example, if an HT PPDU transmission uses 64-QAM and coding rate of 3/4, the related non-HT reference rate is 54 Mb/s.

Table 9-5—Non-HT reference rate

Modulation	Coding rate (R)	Non-HT reference rate (Mb/s)
BPSK	1/2	6
BPSK	3/4	9
QPSK	1/2	12
QPSK	3/4	18
16-QAM	1/2	24
16-QAM	3/4	36
64-QAM	1/2	48
64-QAM	2/3	48
64-QAM	3/4	54
64-QAM	5/6	54

9.8 MSDU transmission restrictions

To avoid reordering MSDUs between pairs of LLC entities and/or unnecessarily discarding MSDUs, the following restrictions shall be observed by any STA that is able to concurrently process multiple outstanding MSDUs for transmission. The term *outstanding* refers to an MPDU containing all or part of an MSDU or MMPDU for which transmission has been started, and for which delivery of the MSDU or MMPDU has not yet been completed (i.e., an acknowledgement of the final fragment has not been received and the MSDU or MMPDU has not been discarded due to retries, lifetime, or for some other reason). A STA may have any number (greater than or equal to one) of eligible MSDUs outstanding concurrently, subject to the restrictions below.

A non-QoS STA shall not have more than one MSDU or MMPDU from a particular SA to a particular individual RA outstanding at a time.

NOTE—A simpler, more restrictive alternative to the rule in the above paragraph that may be used is that no more than one MSDU with a particular individual RA be outstanding at a time.

For frames that are not sent within the context of a Block Ack agreement, a QoS STA shall not have more than one MSDU or A-MSDU for each TID or MMPDU from a particular SA to a particular individual RA outstanding at any time.

NOTE—A simpler, more restrictive alternative to the rule in the above paragraph that may be used is that no more than one MSDU or A-MSDU with any particular TID with a particular individual RA be outstanding at any time.

In a STA where the optional StrictlyOrdered service class has been implemented, that STA shall not have any group addressed (multidestination) MSDU of the StrictlyOrdered service class outstanding from the SA of any other outstanding MSDU (either individual or group addressed). This is because a group addressed MSDU is implicitly addressed to a collection of peer STAs that could include any individual RA.

It is recommended that the STA select a value of dot11MaxTransmitMSDULifetime that is sufficiently large that the STA does not discard MSDUs or A-MSDUs due to excessive Transmit MSDU timeouts under normal operating conditions.

An A-MSDU shall contain only MSDUs of a single service class and inherits that service class for the purpose of the following rules. For MSDUs or A-MSDUs belonging to the service class of QoSAck when the receiver is a QoS STA, the QoS data frames that are used to send these MSDUs or A-MSDUs shall have the Ack Policy subfield in the QoS Control field set to Normal Ack, Block Ack, Implicit Block Ack Request, or PSMP Ack. For MSDUs or A-MSDUs belonging to the service class of QoSNoAck when the receiver is a QoS STA, the QoS data frames that are used to send these MSDUs or A-MSDUs shall have the Ack Policy subfield in the QoS Control field set to No Ack.

9.9 HT Control field operation

If the value of dot11HTControlFieldSupported is true, a STA shall set the +HTC Support subfield of the HT Extended Capabilities field of the HT Capabilities element to 1 in HT Capabilities elements that it transmits.

A STA that has a value of true for at least one of dot11RDResponderOptionImplemented and dot11MCSFeedbackOptionImplemented shall set dot11HTControlFieldSupported to true.

An HT Control field shall not be present in a frame addressed to a STA unless that STA declares support for +HTC in the HT Extended Capabilities field of its HT Capabilities element (see 8.4.2.58).

NOTE—An HT STA that does not support +HTC that receives a +HTC frame addressed to another STA still performs the CRC on the actual length of the MPDU and uses the Duration/ID field to update the NAV, as described in 9.3.2.4.

If the HT Control field is present in an MPDU aggregated in an A-MPDU, then all MPDUs of the same frame type (i.e., having the same value for the Type subfield of the Frame Control field) aggregated in the same A-MPDU shall contain an HT Control field. The HT Control field of all MPDUs containing the HT Control field aggregated in the same A-MPDU shall be set to the same value.

9.10 Control Wrapper operation

A STA supporting the HT Control field that receives a Control Wrapper frame shall process it as though it received a frame of the subtype of the wrapped frame.

NOTE—A STA supporting the HT Control field can reset the NAV set by a wrapped RTS frame following the rules defined in 9.3.2.4.

9.11 A-MSDU operation

An A-MSDU shall contain only MSDUs whose DA and SA parameter values map to the same RA and TA values.

The constituent MSDUs of an A-MSDU shall all have the same priority parameter value from the corresponding MA-UNITDATA.request primitive.

An A-MSDU shall be carried, without fragmentation, within a single QoS data MPDU.

The Address 1 field of an MPDU carrying an A-MSDU shall be set to an individual address.

The channel access rules for a QoS data MPDU carrying an A-MSDU are the same as a data MPDU carrying an MSDU (or fragment thereof) of the same TID.

The expiration of the A-MSDU lifetime timer occurs only when the lifetime timer of all of the constituent MSDUs of the A-MSDU have expired.

NOTE 1—This rule implicitly allows an MSDU that is a constituent of an A-MSDU to potentially be transmitted after the expiry of its lifetime.

NOTE 2—Selecting any other value for the timeout would result in loss of MSDUs. Selecting the maximum value avoids this loss of MSDUs at the cost of transmitting MSDUs that have exceeded their lifetime.

A STA that has a value of false for dot11HighthroughputOptionImplemented shall not transmit an A-MSDU. A STA shall not transmit an A-MSDU to a STA from which it has not received a frame containing an HT Capabilities element.

Support for the reception of an A-MSDU, where the A-MSDU is carried in a QoS data MPDU with Ack Policy equal to Normal Ack and the A-MSDU is not aggregated within an A-MPDU, is mandatory for an HT STA.

The use of an A-MSDU carried in a QoS data MPDU under a Block Ack agreement is determined per Block Ack agreement. A STA shall not transmit an A-MSDU within a QoS data MPDU under a Block Ack agreement unless the recipient indicates support for A-MSDU by setting the A-MSDU Supported field to 1 in its BlockAck Parameter Set field of the ADDBA Response frame.

A STA shall not transmit an A-MSDU to a STA that exceeds its maximum A-MSDU length capability.

NOTE—Support for A-MSDU aggregation does not affect the maximum size of MSDU transported by the MA-UNITDATA primitives.

9.12 A-MPDU operation

9.12.1 A-MPDU contents

According to its context (defined in Table 8-283), an A-MPDU shall be constrained so that it contains only MPDUs as specified in the relevant table referenced from Table 8-283.

When an A-MPDU contains multiple QoS Control fields, bits 4 and 8–15 of these QoS Control fields shall be identical.

9.12.2 A-MPDU length limit rules

An HT STA indicates a value in the Maximum A-MPDU Length Exponent field in its HT Capabilities element that defines the maximum A-MPDU length that it can receive. The encoding of this field is defined in Table 8-125. Using this field, the STA establishes at association the maximum length of A-MPDUs that can be sent to it. The STA shall be capable of receiving A-MPDUs of length up to the value indicated by this field.

An HT STA shall not transmit an A-MPDU that is longer than the value indicated by the Maximum A-MPDU Length Exponent field declared by the intended receiver.

NOTE—The A-MPDU length limit applies to the maximum length of the PSDU that might be received. If the A-MPDU includes any padding delimiters (i.e., delimiters with the Length field equal to 0) in order to meet the MPDU start spacing requirement, this padding is included in this length limit.

9.12.3 Minimum MPDU Start Spacing field

An HT STA shall not start the transmission of more than one MPDU within the time limit described in the Minimum MPDU Start Spacing field declared by the intended receiver. To satisfy this requirement, the number of octets between the start of two consecutive MPDUs in an A-MPDU, measured at the PHY SAP, shall be equal or greater than

$$t_{MMSS} \times r / 8$$

where

t_{MMSS} is the time (in microseconds) defined in the "Encoding" column of Table 8-125 for the value of the Minimum MPDU Start Spacing field

r is the value of the PHY Data Rate (in megabits per second) defined in 20.6 based on the TXVECTOR parameters: MCS, GI_TYPE, and CH_BANDWIDTH

If necessary, in order to satisfy this requirement, a STA shall add padding between MPDUs in an A-MPDU. Any such padding shall be in the form of one or more MPDU delimiters with the MPDU Length field set to 0.

9.12.4 A-MPDU aggregation of group addressed data frames

An HT STA that is neither an AP nor a mesh STA shall not transmit an A-MPDU containing an MPDU with a group addressed RA.

NOTE—An HT AP and an HT mesh STA can transmit an A-MPDU containing MPDUs with a group addressed RA.

An HT AP and an HT mesh STA shall not transmit an A-MPDU containing group addressed MPDUs if the HT Protection field is equal to non-HT mixed mode.

When an HT AP or an HT mesh STA transmits an A-MPDU containing MPDUs with a group addressed RA, both of the following shall apply:

- The value of maximum A-MPDU length exponent that applies is the minimum value in the Maximum A-MPDU Length Exponent subfield of the A-MPDU Parameters field of the HT Capabilities element across all HT STAs associated with the AP or all peer HT mesh STAs.
- The value of minimum MPDU start spacing that applies is the maximum value in the Minimum MPDU Start Spacing subfield of the A-MPDU Parameters field of the HT Capabilities element across all HT STAs associated with the AP or all peer HT mesh STAs.

9.12.5 Transport of A-MPDU by the PHY data service

An A-MPDU shall be transmitted in a PSDU associated with a PHY-TXSTART.request primitive with the TXVECTOR AGGREGATION parameter set to 1. A received PSDU is determined to be an A-MPDU when the associated PHY-RXSTART.indication primitive RXVECTOR AGGREGATION parameter is equal to 1.

9.13 PPDU duration constraint

An HT STA shall not transmit a PPDU that has a duration (as determined by the PHY-TXTIME.confirm primitive defined in 6.5.7) that is greater than aPPDUMaxTime.

9.14 LDPC operation

An HT STA shall not transmit a frame with the TXVECTOR parameter FORMAT set to HT_MF or HT_GF and the TXVECTOR parameter FEC_CODING set to LDPC_CODING unless the RA of the frame corresponds to a STA for which the LDPC Coding Capability subfield of the most recently received HT Capabilities element from that STA contained a value of 1 and dot11LDPCCodingOptionActivated is true.

Further restrictions on TXVECTOR parameter values may apply due to rules found in 9.23 and 9.7.

9.15 STBC operation

Only a STA that sets the Tx STBC subfield to 1 in the HT Capabilities element may transmit frames with a TXVECTOR parameter STBC set to a nonzero value to a STA from which the most recently received value of the Rx STBC field of the HT Capabilities element is nonzero.

9.16 Short GI operation

A STA may transmit a frame with TXVECTOR parameters CH_BANDWIDTH set to HT_CBW20 and GI_TYPE set to SHORT_GI only if all of the following conditions are met:

- The STA is an HT STA.
- The TXVECTOR parameter FORMAT is equal to HT_MF or HT_GF.
- The RA of the frame corresponds to a STA for which the Short GI for 20 MHz subfield of the most recently received HT Capabilities element contained a value of 1.
- dot11ShortGIOptionInTwentyActivated is present and is true.

A STA may transmit a frame with TXVECTOR parameters CH_BANDWIDTH set to HT_CBW40 and GI_TYPE set to SHORT_GI only if all of the following conditions are met:

- The STA is an HT STA.
- The TXVECTOR parameter FORMAT is equal to HT_MF or HT_GF.
- The RA of the frame corresponds to a STA for which the Short GI for 40 MHz subfield of the most recently received HT Capabilities element contained a value of 1.
- dot11ShortGIOptionInFortyActivated is present and is true.

An HT STA shall not transmit a frame with the TXVECTOR parameter FORMAT set to HT_GF and the GI_TYPE parameter set to SHORT_GI when the MCS parameter indicates a single spatial stream.

Further restrictions on TXVECTOR parameter values may apply due to rules found in 9.23 and 9.7.

9.17 Greenfield operation

An HT STA shall not transmit a frame with the TXVECTOR parameter FORMAT set to HT_GF unless the RA of the frame corresponds to a STA for which the HT-Greenfield subfield of the most recently received HT Capabilities element contained a value of 1 and dot11HTGreenfieldOptionActivated is true. Further restrictions may apply due to rules found in 9.23, 9.7, and 10.9.8.5.

9.18 Operation across regulatory domains

9.18.1 General

The PHY of a WLAN is subject to regulations that might vary significantly from one regulatory domain to another. This subclause provides the framework for operation across regulatory domains and describes the mechanism that supports cross-domain mobility and operation in multiple regulatory domains. When this mechanism is active, the dot11MultiDomainCapabilityActivated attribute shall be true.

NOTE—This subclause does not eliminate the need to obtain type acceptance, regulatory approval, equipment authorization, or equipment certification in each of the regulatory domains in which the equipment operates. The mechanisms described in this subclause provide the information to the STA to identify the regulatory domain in which it is located and to cease operation while in those domains for which it does not have type approval. It is incumbent upon the implementer to provide proof of compliance to the requirements of individual regulatory agencies.

The method for configuring individual STAs is outside the scope of this standard. A STA needs to be properly configured for operation in a particular regulatory domain prior to beginning normal operation. Particular care needs to be taken when operating in an IBSS configuration.

9.18.2 Operation upon entering a regulatory domain

A STA that is enabled for operation across regulatory domains uses passive scanning when it has lost connectivity with its ESS. Passive scanning is performed using only the receive capabilities of the STA and is, thus, compatible with regulatory requirements. The timeout for determining the loss of connectivity is system dependent and beyond the scope of this standard.

When a STA with dot11MultiDomainCapabilityActivated true enters a regulatory domain, before transmitting, it shall passively scan to learn at least one valid channel, i.e., a channel upon which it detects IEEE 802.11 frames. The Beacon frame contains information on the country code, the maximum allowable transmit power, and the channels that may be used for the regulatory domain. Optionally, the Beacon frame may also include in the Country element, on a periodic basis, the regulatory information that would be returned in a Probe Response frame. When DSE dependent STA operation is required in a regulatory domain, a dependent STA may be required to receive a Beacon frame signalling dependent enablement (10.12.5), and until this Beacon frame is received, the STA may continue passive scanning to receive such a Beacon frame directly from an enabling STA. Once the STA has acquired the information so that it is able to meet the transmit requirements of the regulatory domain, it shall transmit a Probe Request to an AP to gain the additional necessary regulatory domain information contained in the Probe Response frame, unless the information was previously received in a Beacon frame. The STA then has sufficient information available to configure its PHY for operation in the regulatory domain.

9.18.3 Determination of hopping patterns for FH PHYs

The mechanisms described in this subclause are obsolete. Consequently, this subclause may be removed in a later revision of the standard.

The Beacon may contain FH Parameters and/or FH Pattern Table elements. If the Beacon contains both FH Parameters and FH Pattern Table elements, both of these elements shall describe the same hopping pattern. Note that the information returned as a result of a Probe Request frame with a Request element may include the FH parameters and/or the FH Pattern Table possibly replicating optional elements identified by orders 12 and 13 in Table 8-27. When operating in a regulatory domain that does not have a method for determining a hopping pattern described in 14.7.8 or a hopping table in Annex I, hopping patterns shall be determined by an AP using HCCs or EHCCs. The HCC hopping sequences are derived from a simple formula that uses field operations on a group. For full details of the HCC placement operator function, please see the references [B11] and [B51]. The placement operator function shall be as shown in Equation (9-8).

$$y_{HCC}(k;a) = \frac{a}{k} \bmod N \qquad \text{for } k, a \in J'_N \tag{9-8}$$

where

N is the prime radix
a is the family index
k is in the group J'_N
J'_N is the group remaining when the element containing the value 0 is removed from the field J_N

Therefore, k does not take the value 0. The value $1/k$ is the multiplicative inverse of k on the field J_N. The multiplicative inverse of k on the group J_N is the integer value w, such that $(k \times w) \bmod N = 1$. The values computed for y_{HCC} are the channel numbers, a corresponds to the hopping pattern number, and k corresponds to the index into the hopping pattern. A code family is the set of *N*–1 hopping patterns generated for the prime radix *N*. There is no value equivalent to the hopping set of Clause 14. Each hopping pattern comprises *N*–1 channels.

As an example, consider a code family that supports 10 channels. The prime radix for such a family is 11. The code family generated by the HCC algorithm is shown below in Table 9-6.

Table 9-6—HCC family – *N* = 11; Family indices (SEQ) 1 to 10

INDEX (k) ---->

SEQ	1	2	3	4	5	6	7	8	9	10
1	1	6	4	3	9	2	8	7	5	10
2	2	1	8	6	7	4	5	3	10	9
3	3	7	1	9	5	6	2	10	4	8
4	4	2	5	1	3	8	10	6	9	7
5	5	8	9	4	1	10	7	2	3	6
6	6	3	2	7	10	1	4	9	8	5
7	7	9	6	10	8	3	1	5	2	4
8	8	4	10	2	6	5	9	1	7	3
9	9	10	3	5	4	7	6	8	1	2
10	10	5	7	8	2	9	3	4	6	1

The HCC method to calculate hopping sequences only generates sequences of a length that is one less than the prime radix. The EHCC algorithm extends the original HCC algorithm to support a larger number of possible hopping sequence lengths. The EHCC algorithm works through a process known as "deletion of the diagonals." This creates hopping patterns with *N*–2 and *N*–3 channels.

Using the same example in Table 9-6, a family of codes cannot be generated for code lengths of 9 or 8 by the HCC algorithm because neither 9 or 8 is equal to a prime number minus one. However, the diagonals of the array in Table 9-6 represent the end points of the group. Thus, code families for code lengths 9 and 8 can be easily generated from the table simply by removing the diagonals. Table 9-7 shows such a code family with a code length of 9 (constructed by removing the diagonal of 10s).

Table 9-7 now contains a near diagonal (upper left to lower right) consisting entirely of ones. This is a necessary mathematical property of the result of removing the initial diagonal of (p–1) in the previous operation, and is exhibited for any prime p.

Table 9-7—EHCC family – Code length = 9, *N* = 11; Family Indices (SEQ) 1 to 9

INDEX (k) ---->

SEQ	1	2	3	4	5	6	7	8	9
1	1	6	4	3	9	2	8	7	5
2	2	1	8	6	7	4	5	3	9
3	3	7	1	9	5	6	2	4	8
4	4	2	5	1	3	8	6	9	7
5	5	8	9	4	1	7	2	3	6
6	6	3	2	7	1	4	9	8	5
7	7	9	6	8	3	1	5	2	4
8	8	4	2	6	5	9	1	7	3
9	9	3	5	4	7	6	8	1	2

Extending the process, Table 9-8 shows a code family with a code length of 8 (constructed from Table 9-7 by removing the entry from each row that has a value of 1, subtracting 1 from each value remaining in the row, and discarding the last row).

Table 9-8—EHCC family – Code length = 8, *N* = 11; Family indices (SEQ) 1 to 8

INDEX (k) ---->

SEQ	1	2	3	4	5	6	7	8
1	5	3	2	8	1	7	6	4
2	1	7	5	6	3	4	2	8
3	2	6	8	4	5	1	3	7
4	3	1	4	2	7	5	8	6
5	4	7	8	3	6	1	2	5
6	5	2	1	6	3	8	7	4
7	6	8	5	7	2	4	1	3
8	7	3	1	5	4	8	6	2

To obtain a code family of length *N*–2, the code family of length *N*–1 calculated by the HCC algorithm shall be modified by deleting the diagonal of the code family with the value of *N*–1 and removing the row of the code family with the family coefficient of *N*–1. This results in a code family represented in a square (*N*–2) by (*N*–2) array.

To obtain a code family of length *N*–3, the code family of length *N*–1 calculated by the HCC algorithm shall be modified by deleting the diagonals of the code family with the values of 1 and *N*–1, removing the rows of the code family with the family coefficients of *N*–1 and *N*–2, and subtracting 1 from all remaining values in the code family array. This results in a code family represented in a square (*N*–3) by (*N*–3) array with array values of 1 to *N*–3.

When using hopping patterns calculated using the HCC or EHCC algorithms, the values in the FH parameter set element shall be set as follows:

a) The Hop Set field shall be 0; the value of 0 for the Hop Set field indicates that the HCC/EHCC algorithm is in use and that the Hop Pattern field contains the HCC/EHCC family index and the Hop Index field contains the HCC/EHCC index.

b) The family index of the code being used shall be placed in the Hop Pattern field.

c) The index shall be placed in the Hop Index field.

9.18.4 Hopping sequence generation using the Frequency Hopping and Hopping Pattern Table elements

The mechanisms described in this subclause are obsolete. Consequently, this subclause may be removed in a later revision of the standard.

Two equations are used to create a hopping sequence from the information in the Frequency Hopping element and the Hopping Pattern Table element. Equation (9-9), the Random Table Method, shall be used when the value of the Flag field of the Hopping Pattern Table element is 1. Equation (9-10), the Hop Index Method, shall be used when the value of the Flag field of the Hopping Pattern Table element is 0.

$$f_x(i) = [b(i) + x]\bmod(m) + q \tag{9-9}$$

$$f_x(i) = [(i-1) \times x]\bmod(m) + q \tag{9-10}$$

where

i is the index.
m is the modulus.
q is the offset
x is given by Equation (9-11)

$$x = n \times p + s - 1 \tag{9-11}$$

where

n is the number of sets.
p is the current pattern.
s is the current set number.

The values of *i, p,* and *s* are found in the Frequency Hopping element. The values of *m, n,* and *q* are found in the Hopping Pattern Table element.

9.18.5 Operation with operating classes

When dot11OperatingClassesImplemented is true, the following statements apply:

- When dot11OperatingClassesRequired is false, or where operating classes domain information is not present in a STA, that STA is not required to change its operation in response to an element or element-specific Information field that contains an operating class.
- When dot11OperatingClassesRequired is true, or where operating classes domain information is present in a STA, the STA shall indicate current operating class information in the Country element and Supported Operating Classes element.
- When dot11OperatingClassesRequired and dot11ExtendedChannelSwitchActivated are true and a STA is capable of operating as specified in more than one operating class, the STA shall include the Supported Operating Classes element in Association frames and Reassociation frames.
- When dot11OperatingClassesRequired is true, or where operating classes domain information is present and the STA parsing a Country element finds an invalid First Channel Number field or Operating Class field with a value that is reserved, the STA shall ignore the remainder of the Country element and shall parse any remaining management frame body for additional elements.

9.18.6 Operation with coverage classes

The default PHY parameters are based on aAirPropagationTime having a value of 1 µs or less, and aSlotTime and other MAC timing are based on the PHY timing parameters, as specified in 9.3.2.3 and 9.3.7. When dot11OperatingClassesRequired is true, it is possible to manage the MAC timing of STAs that can receive Beacon frames or Probe Response frames that contain the Country element (8.4.2.10), to increase fairness in contending for the medium. Radio waves propagate at 300 m/µs in free space, and, for example, 3 µs would be the ceiling for BSS maximum one-way distance of ~450 m (~900 m round trip). The Coverage Class field of the Country element indicates the new value of aAirPropagationTime (see Table 8-56), and the MAC can use the new value to calculate aSlotTime (see 9.3.7). When dot11OperatingClassesRequired and dot11ExtendedChannelSwitchActivated are true and Country elements have been received in Beacon frames or Probe Response frames, associated STAs and dependent STAs shall use MAC timing that corresponds to the new value of aAirPropagationTime (see 9.3.7).

Using the Country element, an AP can change coverage class and maximum transmit power level to enhance operation. When dot11OperatingClassesRequired and dot11ExtendedChannelSwitchActivated are true and the maximum transmit power level is different from the transmit power limit indicated by the operating class, the associated STA or dependent STA shall operate at a transmit power at or below that indicated by the lesser of the two limits.

9.19 HCF

9.19.1 General

Under HCF, the basic unit of allocation of the right to transmit onto the WM is the TXOP. Each TXOP is defined by a starting time and a defined maximum length. The TXOP may be obtained by a STA winning an instance of EDCA contention (see 9.19.2) during the CP or by a STA receiving a QoS (+)CF-Poll frame (see 9.19.3) during the CP or CFP. The former is called *EDCA TXOP*, while the latter is called *HCCA TXOP* or *polled TXOP*. An HCCA TXOP shall not extend across a TBTT. A TXOP shall not exceed dot11MaxDwellTime (if using an FH PHY). The occurrence of a TBTT implies the end of the HCCA TXOP, after which the regular channel access procedure (EDCA or HCCA) is resumed. It is possible that no frame was transmitted during the TXOP. The shortened termination of the HCCA TXOP does not imply an error condition.

9.19.2 HCF contention-based channel access (EDCA)

9.19.2.1 Reference implementation

The channel access protocol is derived from the DCF procedures described in 9.3.

A model of the reference implementation is shown in Figure 9-19 and illustrates a mapping from frame type or UP to AC: the four transmit queues and the four independent EDCAFs, one for each queue. The mapping of UP to the AC is described in 9.2.4.2 and Table 9-1. The mapping of frame types to ACs is described in 9.2.4.2.

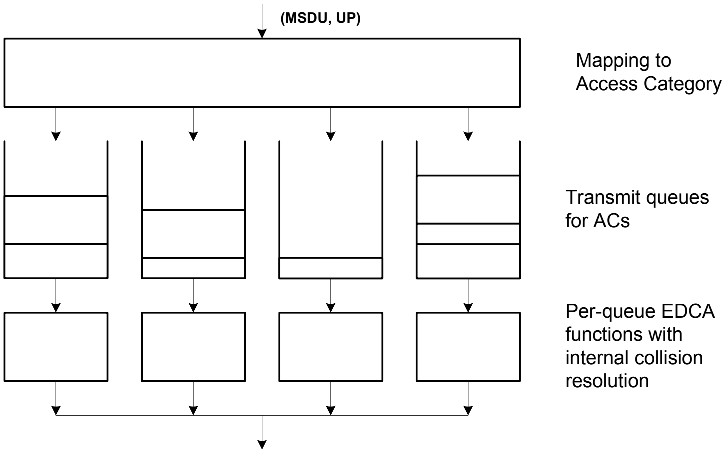

(MSDU, UP)
Mapping to Access Category
Transmit queues for ACs
Per-queue EDCA functions with internal collision resolution

Figure 9-19—Reference implementation model

9.19.2.2 EDCA TXOPs

There are two modes of EDCA TXOP defined, the initiation of the EDCA TXOP and the multiple frame transmission within an EDCA TXOP. An initiation of the TXOP occurs when the EDCA rules permit access to the medium. A multiple frame transmission within the TXOP occurs when an EDCAF retains the right to access the medium following the completion of a frame exchange sequence, such as on receipt of an ACK frame.

The TXOP limit duration values are advertised by the AP in the EDCA Parameter Set element in Beacon and Probe Response frames transmitted by the AP.

A TXOP limit value of 0 indicates that the TXOP holder may transmit or cause to be transmitted (as responses) the following within the current TXOP:

a) A single MSDU, MMPDU, A-MSDU, A-MPDU, or PS-Poll at any rate, subject to the rules in 9.7
b) Any required acknowledgments
c) Any frames required for protection, including one of the following:
    1) An RTS/CTS exchange
    2) CTS to itself
    3) Dual CTS as specified in 9.3.2.7
d) Any frames required for beamforming as specified in 9.27
e) Any frames required for link adaptation as specified in 9.28
f) Any number of BlockAckReq frames

NOTE 1—This is a rule for the TXOP holder. A TXOP responder need not be aware of the TXOP limit nor of when the TXOP was started.

NOTE 2—This rule prevents the use of RD when the TXOP limit is 0.

When dot11OCBActivated is true, TXOP limits shall be 0 for each AC.

STAs shall limit the duration of TXOPs obtained using the EDCA rules to the value specified by the TXOP limit. The duration of a TXOP is the duration during which the TXOP holder maintains uninterrupted control of the medium, and it includes the time required to transmit frames sent as an immediate response to the TXOP holder's transmissions.

When the TXOP limit is nonzero, a STA shall fragment an individually addressed MSDU so that the transmission of the first MPDU of the TXOP does not cause the TXOP limit to be exceeded at the PHY rate selected for the initial transmission attempt of that MPDU. The TXOP limit may be exceeded, when using a lower PHY rate than selected for the initial transmission attempt of the first MPDU, for a retransmission of an MPDU, for the initial transmission of an MPDU if any previous MPDU in the current MSDU has been retransmitted, or for group addressed MSDUs. When the TXOP limit is exceeded due to the retransmission of an MPDU at a reduced PHY rate, the STA shall not transmit more than one MPDU in the TXOP.

It should be noted, that when transmitting multiple frames in a TXOP using acknowledgment mechanisms other than Normal Ack, a protective mechanism should be used (such as RTS/CTS or the protection mechanism described in 9.23). A QoS AP or a mesh STA may send group addressed frames without using any protection mechanism. In a QoS IBSS, group addressed frames shall be sent one at a time, and backoff shall be performed after the transmission of each of the group addressed frames. In an MBSS, a mesh STA may send multiple group addressed frames in a TXOP, bounded by the TXOP limit, without performing backoff after the TXOP is obtained.

A STA shall save the TXOP holder address for the BSS in which it is associated, which is the MAC address from the Address 2 field of the frame that initiated a frame exchange sequence except when this is a CTS frame, in which case the TXOP holder address is the Address 1 field. If an RTS frame is received with the RA address matching the MAC address of the STA and the MAC address in the TA field in the RTS frame matches the saved TXOP holder address, then the STA shall send the CTS frame after SIFS, without regard for, and without resetting, its NAV. When a STA receives a frame addressed to it that requires an immediate response, except in the case of an RTS, it shall transmit the response independent of its NAV. The saved TXOP holder address shall be cleared when the NAV is reset or when the NAV counts down to 0.

9.19.2.3 Obtaining an EDCA TXOP

Each channel access timer shall maintain a backoff function (timer), which has a value measured in backoff slots.

The duration AIFS[AC] is a duration derived from the value AIFSN[AC] by the relation

$$\text{AIFS[AC]} = \text{AIFSN[AC]} \times \text{aSlotTime} + \text{aSIFSTime}.$$

The value of AIFSN[AC] shall be greater than or equal to 2. In an infrastructure BSS, AIFSN[AC] is advertised by the AP in the EDCA Parameter Set element in Beacon and Probe Response frames transmitted by the AP. The value of AIFSN[AC] shall be greater than or equal to 1 for APs. An EDCA TXOP is granted to an EDCAF when the EDCAF determines that it shall initiate the transmission of a frame exchange sequence. Transmission initiation shall be determined according to the following rules:

On specific slot boundaries, each EDCAF shall make a determination to perform one and only one of the following functions:

- Initiate the transmission of a frame exchange sequence for that access function.
- Decrement the backoff timer for that access function.

- Invoke the backoff procedure due to an internal collision.
- Do nothing for that access function.

The specific slot boundaries at which exactly one of these operations shall be performed are defined as follows, for each EDCAF:

a) Following AIFSN[AC] × aSlotTime – aRxTxTurnaroundTime of idle medium after SIFS (not necessarily idle medium during the SIFS duration) after the last busy medium on the antenna that was the result of a reception of a frame with a correct FCS.

b) Following EIFS – DIFS + AIFSN[AC] × aSlotTime + aSIFSTime – aRxTxTurnaroundTime of idle medium after the last indicated busy medium as determined by the physical CS mechanism that was the result of a frame reception that has resulted in FCS error, or PHY-RXEND.indication (RXERROR) primitive where the value of RXERROR is not NoError.

c) When any other EDCAF at this STA transmitted a frame requiring acknowledgment, the earlier of

   1) The end of the ACK-Timeout interval timed from the PHY_TXEND.confirm primitive, followed by AIFSN[AC] x aSlotTime + aSIFSTime – aRxTxTurnaroundTime of idle medium, and

   2) The end of the first AIFSN[AC] × aSlotTime – aRxTxTurnaroundTime of idle medium after SIFS (not necessarily medium idle during the SIFS duration, the start of the SIFS duration implied by the length in the PLCP header of the previous frame) when a PHY-RXEND.indication primitive occurs as specified in 9.3.2.8.

d) Following AIFSN[AC] × aSlotTime – aRxTxTurnaroundTime of idle medium after SIFS (not necessarily medium idle during the SIFS duration) after the last busy medium on the antenna that was the result of a transmission of a frame for any EDCAF and which did not require an acknowledgment.

e) Following AIFSN[AC] × aSlotTime + aSIFSTime – aRxTxTurnaroundTime of idle medium after the last indicated idle medium as indicated by the CS mechanism that is not covered by a) to d).

f) Following aSlotTime of idle medium, which occurs immediately after any of these conditions, a) to f), is met for the EDCAF.

At each of the above-described specific slot boundaries, each EDCAF shall initiate a transmission sequence if

- There is a frame available for transmission at that EDCAF, and
- The backoff timer for that EDCAF has a value of 0, and
- Initiation of a transmission sequence is not allowed to commence at this time for an EDCAF of higher UP.

At each of the above-described specific slot boundaries, each EDCAF shall decrement the backoff timer if the backoff timer for that EDCAF has a nonzero value.

At each of the above-described specific slot boundaries, each EDCAF shall report an internal collision (which is handled in 9.19.2.5) if

- There is a frame available for transmission at that EDCAF, and
- The backoff timer for that EDCAF has a value of 0, and
- Initiation of a transmission sequence is allowed to commence at this time for an EDCAF of higher UP.

An example showing the relationship between AIFS, AIFSN, DIFS, and slot times immediately following a medium busy condition (and assuming that medium busy condition was not caused by a frame in error) is shown in Figure 9-20. In this case, with AIFSN = 2, the EDCAF may decrement the backoff counter for the first time at 2 × aSlotTime following the TXSIFS (where TXSIFS is the time at which the MAC responds to the end of the medium busy condition if it is going to respond “after SIFS”). If, in this example, the backoff counter contained a value of 1 at the time the medium became idle, transmission would start as a result of an EDCA TXOP on-air at a time

aSIFSTime + 3 × aSlotTime

following the end of the medium busy condition.

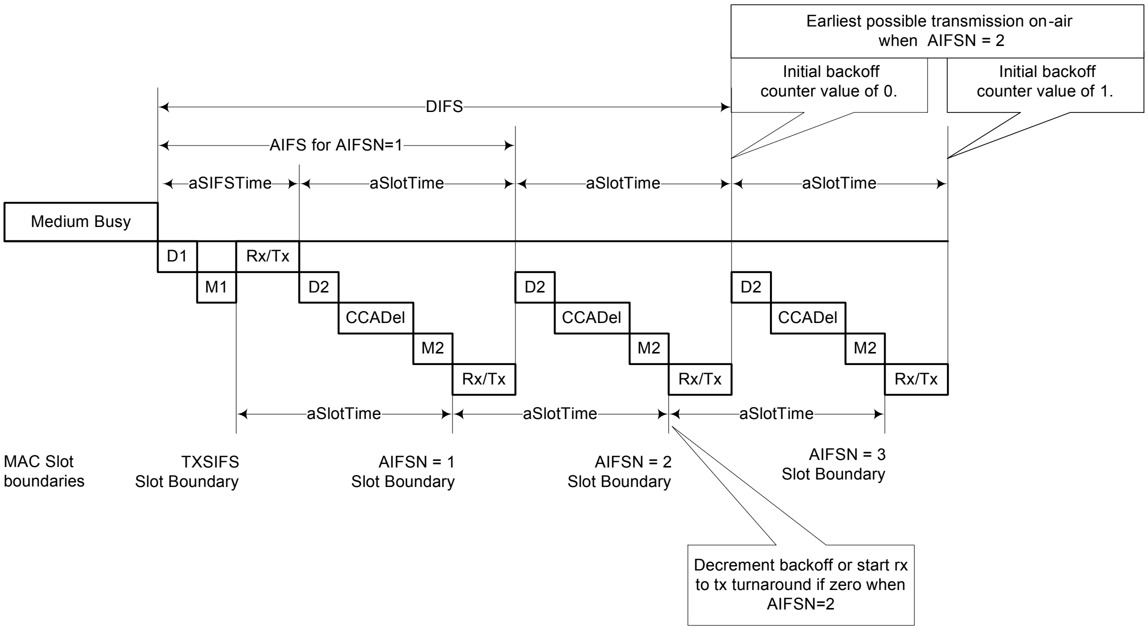


Figure 9-20—EDCA mechanism timing relationships

9.19.2.4 Multiple frame transmission in an EDCA TXOP

Multiple frames may be transmitted in an EDCA TXOP that was acquired following the rules in 9.19.2.3 if there is more than one frame pending in the AC for which the channel has been acquired. However, those frames that are pending in other ACs shall not be transmitted in this EDCA TXOP. If a TXOP holder has in its transmit queue an additional frame of the same AC as the one just transmitted and the duration of transmission of that frame plus any expected acknowledgment for that frame is less than the remaining TXNAV timer value, then the STA may commence transmission of that frame a SIFS (or RIFS, under the conditions defined in 9.3.2.3.2) after the completion of the immediately preceding frame exchange sequence, subject to the TXOP limit restriction as described in 9.19.2.2. An HT STA that is a TXOP holder may transmit multiple MPDUs of the same AC within an A-MPDU as long as the duration of transmission of the A-MPDU plus any expected BlockAck response is less than the remaining TXNAV timer value.

NOTE—An RD responder can transmit multiple MPDUs as described in 9.25.4

The TXNAV timer is a timer that is initialized with the duration from the Duration/ID field in the frame most recently successfully transmitted by the TXOP holder. The TXNAV timer begins counting down from the end of the transmission of the PPDU containing that frame. Following the BlockAck response, the HT STA may start transmission of another MPDU or A-MPDU a SIFS after the completion of the immediately preceding frame exchange sequence. The HT STA may retransmit unacknowledged MPDUs within the same TXOP or in a subsequent TXOP.

After a valid response to the initial frame of a TXOP, if the Duration/ID field is set for multiple frame transmission and there is a subsequent transmission failure, the corresponding channel access function may transmit after the CS mechanism (see 9.3.2.1) indicates that the medium is idle at the TxPIFS slot boundary (defined in 9.3.7) before the expiry of the TXNAV timer. At the expiry of the TXNAV timer, if the channel access function has not regained access to the medium, then the EDCAF shall invoke the backoff procedure that is described in 9.19.2.5. Transmission failure is defined in 9.19.2.5.

All other channel access functions at the STA shall treat the medium as busy until the expiry of the TXNAV timer.

A frame exchange may be a group addressed frame, a frame transmitted with No Ack policy (for which there is no expected acknowledgment), or an individually addressed frame followed by a correctly received ACK frame transmitted by a STA (either a non-AP STA or an AP).

Note that, as for an EDCA TXOP, a multiple frame transmission is granted to an EDCAF, not to a STA, so that the multiple frame transmission is permitted only for the transmission of a frame of the same AC as the frame that was granted the EDCA TXOP, unless the EDCA TXOP obtained is used by an AP for a PSMP sequence. In such a case, this AC transmission restriction does not apply to either the AP or the STAs participating in the PSMP sequence, but the specific restrictions on transmission during a PSMP sequence described in 9.26 do apply.

9.19.2.5 EDCA backoff procedure

Each EDCAF shall maintain a state variable CW[AC], which shall be initialized to the value of the parameter CWmin[AC].

For the purposes of this subclause, successful transmission and transmission failure are defined as follows:

— After transmitting an MPDU (regardless of whether it is carried in an A-MPDU) that requires an immediate frame as a response, the STA shall wait for a timeout interval of duration of aSIFSTime + aSlotTime + aPHY-RX-START-Delay, starting at the PHY-TXEND.confirm primitive. If a PHY-RXSTART.indication primitive does not occur during the timeout interval, the STA concludes that the transmission of the MPDU has failed.

— If a PHY-RXSTART.indication primitive does occur during the timeout interval, the STA shall wait for the corresponding PHY-RXEND.indication primitive to determine whether the MPDU transmission was successful. The recognition of a valid response frame sent by the recipient of the MPDU requiring a response, corresponding to this PHY-RXEND.indication primitive, shall be interpreted as a successful response.

— The recognition of anything else, including any other valid frame, shall be interpreted as failure of the MPDU transmission. The recognition of a valid data frame sent by the recipient of a PS-Poll frame shall also be accepted as successful acknowledgment of the PS-Poll frame. A transmission that does not require an immediate frame as a response is defined as a successful transmission.

The backoff procedure shall be invoked for an EDCAF when any of the following events occurs:

a) A frame with that AC is requested to be transmitted, the medium is busy as indicated by either physical or virtual CS, and the backoff timer has a value of 0 for that AC.

b) The final transmission by the TXOP holder initiated during the TXOP for that AC was successful and the TXNAV timer has expired.

c) The transmission of the initial frame of a TXOP of that AC fails.

d) An internal collision is reported for that EDCAF (see 9.19.2.3).

In addition, the backoff procedure may be invoked for an EDCAF when the transmission of a non-initial frame by the TXOP holder fails.

NOTE—A STA can perform a PIFS recovery, as described in 9.19.2.4, or perform a backoff, as described in the previous paragraph, as a response to transmission failure within a TXOP. How it chooses between these two is implementation dependent.

A STA that performs a backoff within its existing TXOP shall not extend the TXNAV timer value.

NOTE—In other words, the backoff is a continuation of the TXOP, not the start of a new TXOP.

If the backoff procedure is invoked for reason a) above, the value of CW[AC] shall be left unchanged. If the backoff procedure is invoked because of reason b) above, the value of CW[AC] shall be reset to CWmin[AC].

If the backoff procedure is invoked because of a failure event [reason c) or d) above or the transmission failure of a non-initial frame by the TXOP holder], the value of CW[AC] shall be updated as follows before invoking the backoff procedure:

— If the QSRC[AC] or the QLRC[AC] for the QoS STA has reached dot11ShortRetryLimit or dot11LongRetryLimit respectively, CW[AC] shall be reset to CWmin[AC].
— Otherwise,
— If CW[AC] is less than CWmax[AC], CW[AC] shall be set to the value (CW[AC] + 1)×2 – 1.
— If CW[AC] is equal to CWmax[AC], CW[AC] shall remain unchanged for the remainder of any retries.

The backoff timer is set to an integer value chosen randomly with a uniform distribution taking values in the range [0,CW[AC]] inclusive.

All backoff slots occur following an AIFS[AC] period during which the medium is determined to be idle for the duration of the AIFS[AC] period, or following an EIFS – DIFS + AIFS[AC] period during which the medium is determined to be idle for the duration of the EIFS – DIFS + AIFS[AC] period, as appropriate (see 9.3.2.3), except as defined in 9.19.2.3, which allows the medium to be busy during the initial aSIFSTime of this period under certain conditions.

If the backoff procedure is invoked following the transmission of a 40 MHz mask PPDU, the backoff counter shall be decremented based on a medium busy indication that ignores activity in the secondary channel. Additional 40 MHz mask PPDU backoff rules are found in 10.15.9.

9.19.2.6 Retransmit procedures

QoS STAs shall maintain a short retry counter and a long retry counter for each MSDU, A-MSDU, or MMPDU that belongs to a TC requiring acknowledgment. The initial value for the short and long retry counters shall be 0. QoS STAs also maintain a short retry counter and a long retry counter for each AC. They are defined as QSRC[AC] and QLRC[AC], respectively, and each is initialized to a value of 0.

After transmitting a frame that requires an immediate acknowledgment, the STA shall perform either of the acknowledgment procedures, as appropriate, that are defined in 9.3.2.8 and 9.21.3. The short retry count for an MSDU or A-MSDU that is not part of a Block Ack agreement or for an MMPDU shall be incremented every time transmission of a frame of length less than or equal to dot11RTSThreshold fails for that MSDU, A-MSDU, or MMPDU. QSRC[AC] shall be incremented every time transmission of an A-MPDU or frame of length less than or equal to dot11RTSThreshold fails. This short retry count and the QoS STA QSRC[AC] shall be reset when an A-MPDU or frame of length less than or equal to dot11RTSThreshold succeeds. The long retry count for an MSDU or A-MSDU that is not part of a Block Ack agreement or for an MMPDU shall be incremented every time transmission of a MAC frame of length greater than dot11RTSThreshold fails for that MSDU, A-MSDU, or MMPDU. QLRC[AC] shall be incremented every time transmission of an A-MPDU or frame of length greater than or equal to dot11RTSThreshold fails. This long retry count and the QLRC[AC] shall be reset when an A-MPDU or frame of length greater than dot11RTSThreshold succeeds. All retransmission attempts for an MPDU that is not sent under a Block Ack agreement and that has failed the acknowledgment procedure one or more times shall be made with the Retry field set to 1 in the data or management frame.

Retries for failed transmission attempts shall continue until the short retry count for the MSDU, A-MSDU, or MMPDU is equal to dot11ShortRetryLimit or until the long retry count for the MSDU, A-MSDU, or MMPDU is equal to dot11LongRetryLimit. When either of these limits is reached, retry attempts shall cease, and the MSDU, A-MSDU, or MMPDU shall be discarded.

For internal collisions occurring with the EDCA access method, the appropriate retry counters (short retry counter for MSDU, A-MSDU, or MMPDU and QSRC[AC] or long retry counter for MSDU, A-MSDU, or MMPDU and QLRC[AC]) are incremented. For transmissions that use Block Ack, the rules in 9.21.3 also apply. STAs shall retry failed transmissions until the transmission is successful or until the relevant retry limit is reached.

With the exception of a frame belonging to a TID for which Block Ack is set up, a QoS STA shall not initiate the transmission of any management or data frame to a specific RA while the transmission of another management or data frame with the same RA and having been assigned its sequence number from the same sequence counter has not yet completed to the point of success, retry fail, or other MAC discard (e.g., lifetime expiry).

QoS STAs shall maintain a transmit MSDU timer for each MSDU passed to the MAC. dot11EDCATableMSDULifetime specifies the maximum amount of time allowed to transmit an MSDU for a given AC. The transmit MSDU timer shall be started when the MSDU is passed to the MAC. If the value of this timer exceeds the appropriate entry in dot11EDCATableMSDULifetime, then the MSDU, or any remaining, undelivered fragments of that MSDU, shall be discarded by the source STA without any further attempt to complete delivery of that MSDU.

When A-MSDU aggregation is used, the HT STA maintains a single timer for the whole A-MSDU. The timer is restarted each time an MSDU is added to the A-MSDU. The result of this procedure is that no MSDU in the A-MSDU is discarded before a period of dot11EDCATableMSDULifetime has elapsed.

9.19.2.7 Truncation of TXOP

When a STA gains access to the channel using EDCA and empties its transmission queue, it may transmit a CF-End frame provided that the remaining duration is long enough to transmit this frame. By transmitting the CF-End frame, the STA is explicitly indicating the completion of its TXOP.

A TXOP holder that transmits a CF-End frame shall not initiate any further frame exchange sequences within the current TXOP.

A non-AP STA that is not the TXOP holder shall not transmit a CF-End frame.

A STA shall interpret the reception of a CF-End frame as a NAV reset, i.e., it resets its NAV timer to 0 at the end of the PPDU containing this frame. After receiving a CF-End frame with a matching BSSID, an AP may respond by transmitting a CF-End frame after SIFS.

NOTE—The transmission of a single CF-End frame by the TXOP holder resets the NAV of STAs hearing the TXOP holder. There may be STAs that could hear the TXOP responder that had set their NAV that do not hear this NAV reset. Those STAs are prevented from contending for the medium until the original NAV reservation expires.

Figure 9-21 shows an example of TXOP truncation. In this example, the STA accesses the medium using EDCA channel access and then transmits a nav-set sequence (e.g., RTS/CTS) (using the terminology of Annex G). After a SIFS, it then transmits an initiator-sequence, which may involve the exchange of multiple PPDUs between the TXOP holder and a TXOP responder. At the end of the second sequence, the TXOP holder has no more data available to send that fits within the TXOP; therefore, it truncates the TXOP by transmitting a CF-End frame.

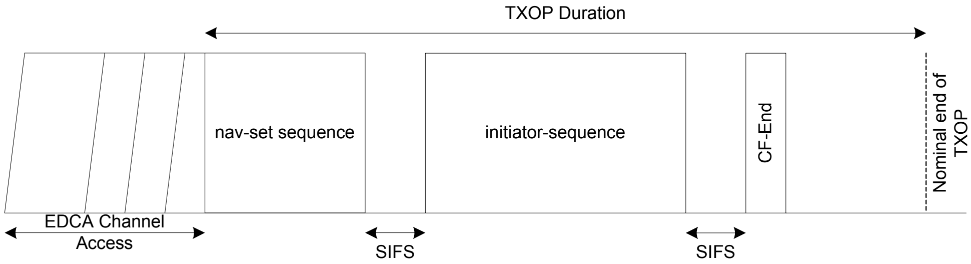


Figure 9-21—Example of TXOP truncation

STAs that receive the CF-End frame reset their NAV and can start contending for the medium without further delay.

TXOP truncation shall not be used in combination with L-SIG TXOP protection when the HT Protection field of the HT Operation element is equal to nonmember protection mode or non-HT mixed mode.

9.19.3 HCCA

9.19.3.1 General

The HCCA mechanism manages access to the WM using an HC that has higher medium access priority than non-AP STAs. This allows it to transfer MSDUs to STAs and to allocate TXOPs to STAs.

The HC is a type of centralized coordinator, but differs from the PC used in PCF in several significant ways, although it may optionally implement the functionality of a PC. Most important is that HCF frame exchange sequences may be used among STAs associated in an infrastructure BSS during both the CP and the CFP. Another significant difference is that the HC grants a STA a polled TXOP with duration specified in a QoS (+)CF-Poll frame. STAs may transmit multiple frame exchange sequences within given polled TXOPs, subject to the limit on TXOP duration.

All STAs inherently obey the NAV rules of the HCF because each frame transmitted under HCF contains a duration value chosen to cause STAs in the BSS to set their NAVs to protect the expected subsequent frames.

All non-AP QoS STAs shall be able to respond to QoS (+)CF-Poll frames received from an HC with the Address 1 field matching their own addresses.

The HC shall perform delivery of buffered group addressed MSDUs/MMPDUs following DTIM Beacon frames. The HC may also operate as a PC, providing (non-QoS) CF-Polls to associated CF-Pollable STAs using the frame formats, frame exchange sequences, and other applicable rules for PCF specified in 9.4.[28]

An HC may perform a backoff following an interruption of a frame exchange sequence due to lack of an expected response under the rules described in 9.19.3.2.4, using the parameters dot11HCCWmin, dot11HCCWmax, and dot11HCCAIFSN and the backoff rules in 9.2 and 9.19.2.5. The decision to perform a backoff by the HC is dependent on conditions such as interference from an overlapping BSS. The mechanism to detect the interference from an overlapping BSS and the decision to perform a backoff, DFS (such as in 11.6), or other techniques (such as inter-BSS scheduling) is beyond the scope of this standard.

[28] Attempting to intersperse HCF frame exchange sequences and PCF frame exchange sequences in a single CFP might be extremely complex.

9.19.3.2 HCCA procedure

9.19.3.2.1 General

The HC gains control of the WM as needed to send QoS traffic and to issue QoS (+)CF-Poll frames to STAs by waiting a shorter time between transmissions than the STAs using the EDCA procedures. The duration values used in QoS frame exchange sequences reserve the medium to permit completion of the current sequence.

The HC may include a CF Parameter Set element in the Beacon frames it generates. This causes the BSS to appear to be a point-coordinated BSS to STAs. This causes STAs to set their NAVs to the CFPDurRemaining value in the CF Parameter Set element value at TBTT, as specified in 9.4.4.3. This prevents most contention in the CFP by preventing nonpolled transmissions by STAs regardless of whether they are CF-Pollable.

9.19.3.2.2 CFP generation

The HC may function as a PC that uses the CFP for delivery, generating a CFP as shown in Figure 9-15, with the restriction that the CFP initiated by an HC shall end with a CF-End frame. The HC may also issue QoS (+)CF-Poll frames to associated STAs during the CFP. However, because the HC can also grant polled TXOPs, by sending QoS (+)CF-Poll frames, during the CP, it is not mandatory for the HC to use the CFP for QoS data transfers.

Only an AP that also issues non-QoS CF-Poll frames to associated CF-Pollable STAs may end a CFP with a CF-End+CF-Ack frame and only when the CF-End+CF-Ack is acknowledging a reception from a CF-Pollable non-QoS STA. The use of a non-QoS CF-Poll frame by an AP to a QoS STA is deprecated (for further discussion; see 8.4.1.4).

9.19.3.2.3 CAP generation

When the HC needs access to the WM to start a CFP or a TXOP in CP, the HC shall sense the WM. When the WM is determined to be idle at the TxPifs slot boundary as defined in 9.3.7, the HC shall transmit the first frame of any permitted frame exchange sequence, with the duration value set to cover the CFP or the TXOP. The first permitted frame in a CFP after a TBTT is the Beacon frame. CAPs along with the CFPs and the CPs are illustrated in Figure 9-22.

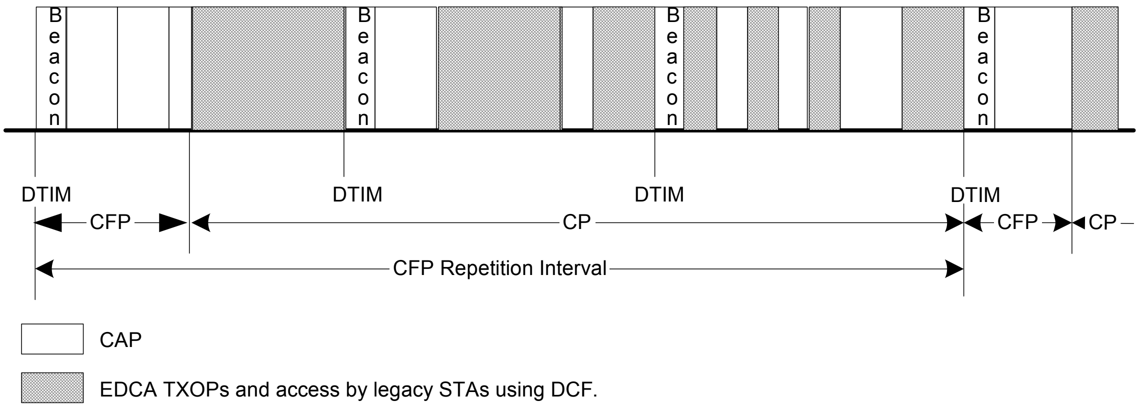


Figure 9-22—CAP/CFP/CP periods

After the last frame of all other nonfinal frame exchange sequences (e.g., sequences that convey individually addressed QoS data or management frames) during a TXOP, the holder of the current TXOP shall wait for one SIFS period before transmitting the first frame of the next frame exchange sequence. The HC may sense the channel and reclaim the channel if the WM is determined to be idle at the TxPifs slot boundary after the TXOP as defined in 9.3.7 A CAP ends when the HC does not reclaim the channel at the TxPifs slot boundary after the end of a TXOP.

9.19.3.2.4 Recovery from the absence of an expected reception

This subclause describes recovery from the absence of an expected reception in a CAP. It should be noted that the recovery rules from the absence of an expected reception are different from EDCA because in this case the NAVs of all the STAs in the BSS have already been set up by the transmissions by the HC. The recovery rules for the multiple frame transmission are different because a STA may always be hidden and may have not set its NAV due to the transmission by another STA. Finally, since an HC is collocated with the AP, the AP may recover using the rules described in this subclause even if the recovery is from the absence of an expected reception.

The beginning of reception of an expected response is detected by the occurrence of PHY-CCA.indication(BUSY,channel-list) primitive at the STA that is expecting the response where

- The channel-list parameter is absent, or
- The channel-list is equal to {primary} and the HT STA expected to transmit the expected response supports 20 MHz operation only, or
- The channel-list is equal to either {primary} or {primary, secondary} and the HT STA expected to transmit the expected response supports both 20 MHz and 40 MHz operation (see 10.15.2).

If the beginning of such reception does not occur during the first slot time following a SIFS, then:[29]

a) If the transmitting STA is the HC, it may initiate recovery by transmitting at the TxPifs slot boundary after the end of the HC's last transmission only if the PHY-CCA.indication primitive is clear during the CCAdel period preceding the TxPifs slot boundary as shown in 9-14.
b) If the transmitting STA is a non-AP QoS STA, and there is an MPDU for transmission, it shall initiate recovery by transmitting at a PIFS after the end of the last transmission, if PHY-CCA.indication primitive is clear, the polled TXOP limit is greater than 0 and at least one frame (re)transmissions can be completed within the remaining duration of a nonzero polled TXOP limit.

If the transmitted frame is not of type QoS (+)CF-Poll and the expected response frame is not received correctly, regardless of the occurrence of the PHY-RXSTART.indication primitive, the QoS STA may initiate recovery following the occurrence of PHY-CCA.indication(idle) primitive so that a SIFS time interval occurs between the last energy on the air and the transmission of the recovery frame.

When there is a transmission failure within a polled TXOP, the long or short retry counter (as described in 9.19.2.6) corresponding to the AC of the failed MSDU or MMPDU shall be incremented. An MPDU belonging to a TC is subject to the respective retry limit as well as the dot11EDCATableMSDULifetime and is discarded when either of them is exceeded. An MPDU belonging to a TS with a specified delay bound is subject to delay bound and is discarded if the MPDU could not be transmitted successfully since it has been delivered to the MAC. An MPDU belonging to a TS with an unspecified delay is subject to dot11MaxTransmitMSDULifetime and is discarded when it is exceeded.

Non-AP STAs that receive a QoS (+)CF-Poll frame shall respond within a SIFS period, regardless of the NAV setting. If a response is not received but a PHY-CCA.indication(busy) primitive occurs during the slot following a SIFS and is followed by a PHY-RXSTART.indication or PHY-RXEND.indication primitive prior to a PHY-CCA.indication(idle) primitive, then the HC assumes that the transmitted QoS (+)CF-Poll frame was successfully received by the polled STA. In the cases of QoS Data+CF-Poll, QoS Data+CF-Ack+CF-Poll, or QoS CF-Ack+CF-Poll, the PHY-CCA.indication(busy) primitive is used only to determine whether the transfer of control of the channel has been successful. The PHY-CCA.indication(busy) primitive is not used for

[29] This restriction is intended to avoid collisions due to inconsistent CCA reports in different STAs, not to optimize the bandwidth usage efficiency.

determining the success or failure of the transmission. If the CF-Poll is piggybacked onto a QoS data frame, the HC may have to retransmit that QoS data frame subsequently.

If an HC receives a frame from a STA with a duration/ID covering only the response frame, the HC shall assume that the STA is terminating its TXOP, and the HC may initiate other transmissions, send a CF-End frame (see 9.19.3.4), or allow the channel to go into the CP.

If a polled QoS STA has no queued traffic to send or if the MPDUs available to send are too long to be transmitted within the specified TXOP limit, the QoS STA shall send a QoS (+)Null frame. In the case of no queued traffic, this QoS (+)Null frame shall have a QoS Control field that reports a queue size of 0 for any TID with the duration/ID set to the time required for the transmission of one ACK frame, plus one SIFS interval. In the case of insufficient TXOP size, such as when the maximum MSDU size is not specified, this QoS (+)Null frame shall have a QoS Control field that contains the TID and TXOP duration or a nonzero queue size needed to send the MPDU that is ready for transmission. When a queue size is transmitted, the HC shall combine the queue size information with the rate of the received QoS (+)Null frame to determine the required size of the requested TXOP.

Within a polled TXOP, the unused portion of TXOPs shall not be used by the STA and may be reallocated by the HC as follows:

a) The recipient of the final frame, with the Ack Policy subfield equal to Normal Ack, shall be the HC if there will be time remaining in the TXOP after the transmission of the final frame and its expected ACK response frame. If there are no frames to be sent to the HC, then the QoS STA shall send to the HC a QoS Null with the Queue Size subfield in the QoS Control field set to 0.
    1) If a PHY-CCA.indication(busy) primitive occurs at the STA that is expecting the ACK response frame during the first slot following a SIFS after the end of the transmission of the final frame, it shall be interpreted as indicating that the channel control has been successfully transferred and no further frames shall be transmitted by the STA in the TXOP, even though the ACK frame from HC may be incorrectly received.[30]
    2) If the beginning of the reception of an expected ACK response frame to the final frame does not occur, detected as the nonoccurrence of PHY-CCA.indication(busy) primitive at the QoS STA that is expecting the response during the first slot time following a SIFS, the QoS STA shall retransmit the frame or transmit a QoS Null frame, with the Ack Policy subfield set to Normal Ack and the Queue Size subfield set to 0, after PIFS from the end of last transmission, until such time that it receives an acknowledgment or when there is not enough time remaining in the TXOP for sending such a frame. [31]

b) If there is not enough time within the unused portion of the TXOP to transmit either the QoS Null frame or the frame with the Duration/ID field covering only the response frame, then the STA shall cease control of the channel.[32]

9.19.3.3 TXOP structure and timing

Any QoS data frame of a subtype that includes CF-Poll contains a TXOP limit in its QoS Control field. The ensuing polled TXOP is protected by the NAV set by the Duration field of the frame that contained the QoS (+)CF-Poll function, as shown in Figure 9-23. Within a polled TXOP, a STA may initiate the transmission of one or more frame exchange sequences, with all such sequences nominally separated by a SIFS interval. The STA shall not initiate transmission of a frame unless the transmission and any acknowledgment or other immediate response expected from the peer MAC entity are able to complete prior to the end of the remaining TXOP duration. All transmissions, including the response frames, within the polled TXOP are considered to be

[30] Note that while PHY-CCA.indication(busy) primitive is used in this instance to determine the control of the channel, it is not used for determining the success or failure of the transmission.

[31] This is to avoid the situation where the HC may not receive the frame and may result in an inefficient use of the channel.

[32] In this case the channel is not accessed until the NAVs expire at all the STAs.

the part of the TXOP, and the HC shall account for these when setting the TXOP limit. If the TXOP Limit subfield in the QoS Control field of the QoS data frame that includes CF-Poll is equal to 0, then the STA to which the frame is directed to shall respond with either one MPDU or one QoS Null frame.

A TXOP or transmission within a TXOP shall not extend across TBTT, dot11CFPMaxDuration (if during CFP), dot11MaxDwellTime (if using an FH PHY), or dot11CAPLimit. The HC shall verify that the full duration of any granted TXOP meets these requirements so that STAs may use the time prior to the TXOP limit of a polled TXOP without checking for these constraints. Subject to these limitations, all decisions regarding what MSDUs, A-MSDUs, and/or MMPDUs are transmitted during any given TXOP are made by the STA that holds the TXOP.[33, 34]

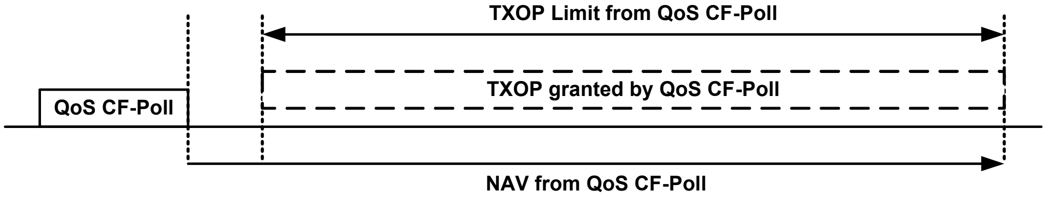


Figure 9-23—Polled TXOP

9.19.3.4 NAV operation during a TXOP

An HC shall set its own NAV to prevent it from transmitting during a TXOP that it has granted to a STA through an HCCA poll. However, the HC may reclaim the TXOP if a STA is not using it or ends the TXOP early (see 9.19.3.2.4).

In a CFP or CP, if the HC has no more STAs to poll and it has no more data, management, BlockAckReq, or BlockAck frames to send, it may reset the NAVs of all QoS STAs in the BSS by sending a QoS CF-Poll frame with the RA matching its own MAC address and with the Duration/ID field set to 0. When the AP contains a PC, during the CFP, it may reset the NAVs of all receiving STAs by sending a CF-End frame, regardless of how the NAVs have been originally set.

A STA that receives a QoS (+)CF-Poll frame with a MAC address in the Address 1 field that matches the HC's MAC address and the Duration/ID field value equal to 0 shall reset the NAV value to 0.

NOTE—A STA resets its NAV when it receives a CF-End or CF-End+ACK frame according to the procedures described in 9.4.3.3.

When a STA receives a QoS (+)CF-Poll frame containing the BSSID of the BSS in which the STA is associated, that STA shall update the NAV if necessary and shall save the TXOP holder address for that BSS, which is the MAC address from the Address 1 field of the frame containing the QoS (+)CF-Poll. If an RTS frame is received with the RA address matching the MAC address of the STA and the MAC address in the TA field in the RTS frame matches the saved TXOP holder address, then the STA shall send the CTS frame after

[33] In certain regulatory domains, channel sensing needs to be done at periodic intervals (for example, in Japan, this period is 4 ms). This means that the duration of a TXOP in these regulatory domains might not be more than this periodic interval. If longer durations are desired, then the TXOP holder needs to sense the channel at least once in the limit imposed in the regulatory domain, by waiting for at least for the duration of one PIFS during which it senses the channel. If it does not detect any energy, it may continue by sending the next frame. In other words, the total TXOP size assigned should include an extra time allocated (i.e., *n* × aSlotTime, where *n* is the number of times the STA needs to sense the channel and is given by Floor(TXOP limit/limit imposed in the regulatory domain).

[34] The TID value in the QoS Control field of a QoS Data+CF-Poll frame pertains only to the MSDU or fragment thereof in the Frame Body field of that frame. This TID value does not pertain to the TXOP limit value and does not place any constraints on what frame(s) the addressed STA may send in the granted TXOP.

SIFS, without regard for, and without resetting, its NAV. The saved TXOP holder address shall be cleared when the NAV is reset or when the NAV counts down to 0.

When a STA receives a frame addressed to it and requires an acknowledgment, it shall respond with an ACK or QoS +CF-Ack frame independent of its NAV. A non-AP STA shall accept a polled TXOP by initiating a frame exchange sequence independent of its NAV.

9.19.3.5 HCCA transfer rules

9.19.3.5.1 General

A TXOP obtained by receiving a QoS (+)CF-Poll frame uses the specified TXOP limit consisting of one or more frame exchange sequences with the sole time-related restriction being that the final sequence shall end not later than the TXOP limit. In QoS CF-Poll and QoS CF-Ack+CF-Poll frames, the TID subfield in the QoS Control field indicates the TS for which the poll is intended. The requirement to respond to that TID is nonbinding, and a STA may respond with any frame. Upon receiving a QoS (+)CF-Poll frame, a STA may send any frames, i.e., QoS data frames belonging to any TID as well as management frames in the obtained TXOP. MSDUs may be fragmented in order to fit within TXOPs.

The QoS CF-Poll frames shall be sent only by an HC. Non-AP STAs are not allowed to send QoS (+)CF-Poll frames. STAs shall not send QoS (+)Data frames in response to any data frame other than the QoS (+)CF-Poll frames.

The TXOP limit is inclusive of the PHY and IFS overhead, and an AP should account for the overhead when granting TXOPs.

If a STA has set up at least one TS for which the Aggregation subfield in the associated TSPEC is equal to 0, the AP shall use only QoS CF-Poll or QoS CF-Ack+CF-Poll frames to poll the STA and shall never use QoS (+)Data+CF-Poll to poll the STA. It should be noted that although QoS (+)CF-Poll is a data frame, but it should be transmitted at one of the rates in the BSSBasicRateSet parameter in order to set the NAV of all STAs that are not being polled (see 9.7). If a CF-Poll is piggybacked with a QoS data frame, then the frame containing all or part of an MSDU or A-MSDU may be transmitted at a rate that is below the negotiated minimum PHY rate in the TSPEC related to that data.

QoS STAs shall use QoS data frames for all MSDU transfers to another QoS STA. The TID in the QoS Control fields of these frames shall indicate the TC or TS to which the MPDU belongs. Furthermore, either the Queue Size subfield shall indicate the amount of queued traffic present in the output queue that the STA uses for traffic belonging to this TC or TS, or the TXOP Duration Requested subfield shall indicate the duration that the STA desires for the next TXOP for traffic belonging to this TC or TS. The queue size value reflects the amount on the appropriate queue not including the present MPDU. A STA that wishes to inform the HC of queue status may use the QoS Null frame indicating the TID and the queue size or TXOP duration request (also see 9.19.3.5.2).

QoS STAs shall be able to receive QoS +CF-Ack frames. The HC may use QoS Data+CF-Ack frames to send frames to the same STA a SIFS after receiving the final transmission of the previous TXOP. The HC may also use QoS Data+CF-Ack frames to send frames to any other STA a SIFS after receiving the final transmission of the previous TXOP, if the STA that sent the final transmission of the previous TXOP has set the Q-Ack subfield in the QoS Capability element in the (Re)Association Request frame to 1. In both CFP and CP, STAs shall respond to QoS data frames having the Ack Policy subfield in the QoS Control field equal to Normal Ack with an ACK frame, unless the acknowledgment is piggybacked in which case it shall use a QoS +CF-Ack frame. Piggybacked frames are allowed only in CFP or within TXOPs initiated by the HC. The HC shall not send a QoS data frame containing a +CF-Ack with an Address 1 that does not correspond to the address of the STA for which the +CF-Ack is intended, unless the STA to which the +CF-Ack is intended, sets the Q-Ack subfield in the QoS Capability element in the (Re)Association Request frame. STAs are not required to be able to transmit QoS data frames with subtypes that include +CF-Ack.

An AP that has dot11QAckOptionImplemented true may allow associations with STAs advertising support for the Q-Ack option. Such associations do not require the AP to employ piggyback acknowledgments directed toward that associated STA in frames that are not directed to that associated STA. QoS STAs shall be able to process received QoS data frames with subtypes that include +CF-Ack when the STA to which the acknowledgment is directed is the same as the STA addressed by the Address 1 field of that data frame. A STA that does not set the Q-Ack subfield to 1 in the QoS Capability element in the (Re)Association Request frame is not required to handle the received QoS (+)Data+CF-Ack frames that are addressed to other STAs. The net effect of these restrictions on the use of QoS +CF-Ack frames is that the principal QoS +CF-Ack subtype that is useful is the QoS Data+CF-Ack frame, which can be sent by a STA as the first frame in a polled TXOP when that TXOP was conveyed in a QoS Data+CF-Poll(+CF-Ack) frame and the outgoing frame is directed to the HC's STA address. QoS (Data+)CF-Poll+CF-Ack frames are useful to allow the HC to grant another TXOP to the same STA a SIFS after receiving the final transmission of that STA's previous TXOP. QoS (Data+)CF-Poll+CF-Ack frames are also useful to allow the HC to grant another TXOP to a different STA a SIFS after receiving the final transmission of a STA's previous TXOP, if the STA that sent the final transmission of the previous TXOP has set the Q-Ack subfield in the QoS Capability element in the (Re)Association Request frame to 1.

HCF contention-based channel access shall not be used to transmit MSDUs belonging to an established TS (with the HC's acceptance of the associated TSPEC), unless the granted TSPEC indicates it is permitted to do so when the Access Policy subfield of the TS Info field is equal to "HCCA, EDCA mixed mode" (HEMM), the polled STA utilized the full TXOP provided by the HC, and it has more MPDUs to send. When this STA sends frames belonging to a TS using contention-based channel access, it shall encode the TID subfield in the QoS data frame with the TID associated with the TS. When the AP grants a TSPEC with the Access Policy subfield equal to HEMM and if the corresponding AC needs admission control, the AP shall include the medium time that specifies the granted time for EDCA access in the ADDTS Response frame.

9.19.3.5.2 TXOP requests

STAs may send TXOP requests during polled TXOPs or EDCA TXOPs using the QoS Control field in a QoS data frame or a QoS Null frame directed to the HC, with the TXOP Duration Requested or Queue Size subfield value and TID subfield value indicated to the HC. APs indicate whether they process TXOP request or queue size in the QoS Info field in the Beacon, Probe Response, and (Re)Association Response frames. APs shall process requests in at least one format. The AP may reallocate TXOPs if the request belongs to TS or update the EDCA parameter set if the above request belongs to TC. STAs shall use only the request format that the AP indicates it can process.

TXOP Duration Requested subfield values are not cumulative. A TXOP duration requested for a particular TID supersedes any prior TXOP duration requested for that TID. A value of 0 in the TXOP Duration Requested subfield may be used to cancel a pending unsatisfied TXOP request when its MSDU is no longer queued for transmission. The TXOP duration requested is inclusive of the PHY and IFS overhead, and a STA should account for this when attempting to determine whether a given transmission fits within a specified TXOP duration.

The AP may choose to assign a TXOP duration shorter than that requested in the TXOP Duration Requested subfield.

Even if the value of TXOP Duration Requested subfield or Queue Size subfield in a QoS data frame is 0, the HC shall continue to poll according to the negotiated schedule.

9.19.3.5.3 Use of RTS/CTS

STAs may send an RTS frame as the first frame of any frame exchange sequence for which improved NAV protection is desired, during either the CP or CFP, and without regard for dot11RTSThreshold.[35]

If a QoS STA sends an RTS frame and does not receive an expected CTS frame, then the recovery rules are as specified in 9.19.3.2.4.

If NAV protection is desired for a transmission to the AP in response to a QoS data frame with a subtype that includes CF-Poll, the polled STA is allowed to send a CTS frame (as a CTS frame is shorter than a QoS data frame and has a higher probability that it will be received by other STAs) with the RA containing its own MAC address in order to set the NAV in its own vicinity without the extra time to send an RTS frame.[36]

9.19.4 Admission Control at the HC

9.19.4.1 General

An IEEE 802.11 network may use admission control to administer policy or regulate the available bandwidth resources. Admission control is also required when a STA desires a guarantee of the amount of time that it has available to access the channel. The HC, which is in the AP, is used to administer admission control in the network. As the QoS facility supports two access mechanisms, there are two distinct admission control mechanisms: one for contention-based access and another for controlled access.

Admission control, in general, depends on vendors' implementation of the scheduler, available channel capacity, link conditions, retransmission limits, and the scheduling requirements of a given stream. All of these criteria affect the admissibility of a given stream. If the HC has admitted no streams that require polling, it may not find it necessary to perform the scheduler or related HC functions.

9.19.4.2 Contention-based admission control procedures

9.19.4.2.1 General

A non-AP STA may support admission control procedures in 9.19.4.2.3 to send frames in the AC where admission control is mandated; but, if it does not support that procedure and dot11RejectUnadmittedTraffic is false or not present, it shall use EDCA parameters of a lower priority AC, as indicated in Table 9-1, that does not require admission control. When a STA uses the EDCA parameters of a lower AC for this purpose, it affects only the EDCA parameters used for channel access, i.e., it has no effect on the contents of the transmitted frame. APs shall support admission control procedures, at least to the minimal extent of advertising that admission is not mandatory on its ACs.

The AP uses the ACM (admission control mandatory) subfields advertised in the EDCA Parameter Set element to indicate whether admission control is required for each of the ACs. While the CWmin, CWmax, AIFS, and TXOP limit parameters may be adjusted over time by the AP, the ACM bit shall be static for the duration of the lifetime of the BSS. An ADDTS Request frame shall be transmitted by a STA to the HC in order to request admission of traffic in any direction (i.e., uplink, downlink, direct, or bidirectional) employing an AC that requires admission control. The ADDTS Request frame shall contain the UP associated with the traffic and

[35] The sending of an RTS frame during the CFP is usually unnecessary, but might be used to confirm that the addressed recipient QoS STA is within range and awake and to elicit a CTS response that sets the NAV at STAs in the vicinity of the addressed recipient. This is useful when there are nearby STAs that are members of other BSSs and are out of range to receive Beacon frames from this BSS. Sending an RTS frame during the CFP is useful only when the recipient is a QoS STA, because a non-QoS STA in the same BSS has its NAV set to protect the CFP, which renders those non-QoS STAs unable to respond. Using the same duration calculation during the CFP as specified for the CP is directly applicable for all cases except when the RTS frame is sent by the HC and the following frame includes a QoS (+)CF-Poll.

[36] This is unnecessary because the NAVs in the vicinity of the QoS AP were set by the QoS (+)CF-Poll frame.

shall indicate EDCA as the access policy. The AP shall associate the received UP of the ADDTS Request frame with the appropriate AC per the UP-to-AC mappings described in 9.2.4.2. The STA may transmit unadmitted traffic for the ACs for which the AP does not require admission control. If a STA desires to send data without admission control using an AC that mandates admission control, the STA shall use EDCA parameters that correspond to a lower priority and do not require admission control unless dot11RejectUnadmittedTraffic is true. When a STA uses a lower priority AC for this purpose, the lower priority AC affects only the EDCA parameters used for channel access, i.e., it has no effect on the contents of the transmitted frame. All ACs with priority higher than that of an AC with an ACM flag equal to 1 should have the ACM flag set to 1. The HC contained within an AP when dot11SSPNInterfaceActivated is true shall admit a non-AP STA's request based on the value of dot11NonAPStationAuthAccessCategories stored in that non-AP STA's dot11InterworkingEntry, which is part of the dot11InterworkingTable. The dot11InterworkingEntry specifies the EDCA access classes and throughput limitations on each access class for which a non-AP STA is permitted to transmit.

9.19.4.2.2 Procedures at the AP

Regardless of the AC’s ACM setting, the AP shall respond to an ADDTS Request frame with an ADDTS Response frame that may be to accept or deny the request. On receipt of an ADDTS Request frame from a non-AP STA, the AP shall make a determination about whether to:

a) Accept the request, or

b) Deny the request.

The algorithm used by the AP to make this determination is implementation dependent. An AP when dot11SSPNInterfaceActivated is true shall use the policies delivered by the SSPN that are stored in the dot11InterworkingEntry, which is part of the dot11InterworkingTable. If the AP decides to accept the request, the AP shall also derive the medium time from the information conveyed in the TSPEC element in the ADDTS Request frame. The AP may use any algorithm in deriving the medium time, but N.2.2 provides a procedure that may be used. Having made such a determination, the AP shall transmit a TSPEC element to the requesting non-AP STA contained in an ADDTS Response frame. If the AP is accepting the request, the Medium Time field shall be specified. If the AP is accepting a request for a downlink TS, the Medium Time field shall be set to 0. If the AP is accepting a request corresponding to an AC for which ACM is 0 (e.g., the TSPEC is to change APSD behavior), the Medium Time field shall be set to 0.

9.19.4.2.3 Procedure at non-AP STAs

Each EDCAF shall maintain two variables: admitted_time and used_time.

The admitted_time and used_time shall be set to 0 at the time of (re)association. The STA may subsequently decide to explicitly request medium time for the AC that is associated with the specified priority.

In order to make such a request, the STA shall transmit a TSPEC element contained in an ADDTS Request frame with the following fields specified (i.e., nonzero): Nominal MSDU Size, Mean Data Rate, Minimum PHY Rate, Inactivity Interval, and Surplus Bandwidth Allowance. The Medium Time field is not used in the request frame and shall be set to 0.

On receipt of a TSPEC element contained in a ADDTS Response frame indicating that the request has been accepted, the STA shall recompute the admitted_time for the specified EDCAF as follows:

admitted_time = admitted_time + dot11EDCAAveragingPeriod × (medium time of TSPEC).

The STA may choose to tear down the explicit request at any time. For the teardown of an explicit admission, the STA shall transmit a DELTS frame containing the TSID and direction that specify the TSPEC to the AP.

If the STA sends or receives a DELTS frame, it shall recompute the admitted_time for the specified EDCAF as follows:

admitted_time = admitted_time – dot11EDCAAveragingPeriod × (medium time of TSPEC).

To describe the behavior at the STA, two parameters are defined. The parameter used_time signifies the amount of time used, in units of 32 µs, by the STA in dot11EDCAAveragingPeriod. The parameter admitted_time is the medium time allowed by the AP, in units of 32 µs, in dot11EDCAAveragingPeriod. The STA shall update the value of used_time:

a) At dot11EDCAAveragingPeriod second intervals

   used_time = max((used_time – admitted_time), 0)

b) After each successful or unsuccessful MPDU (re)transmission attempt,

   used_time = used_time + MPDUExchangeTime

The MPDUExchangeTime equals the time required to transmit the MPDU sequence. For the case of an MPDU transmitted with Normal Ack policy and without RTS/CTS protection, this equals the time required to transmit the MPDU plus the time required to transmit the expected response frame plus one SIFS. Frame exchange sequences for Management frames are excluded from the used_time update. If the used_time value reaches or exceeds the admitted_time value, the corresponding EDCAF shall no longer transmit QoS Data MPDUs or QoS Null MPDUs using the EDCA parameters for that AC as specified in the QoS Parameter Set element. However, a STA may choose to temporarily replace the EDCA parameters for that EDCAF with those specified for an AC of lower priority, if no admission control is required for those ACs.

NOTE—When a frame is transmitted using temporary EDCA parameters, the TID field of that frame is not modified.

If, for example, a STA has made and had accepted an explicit admission for a TS and the channel conditions subsequently worsen, possibly including a change in PHY data rate so that it requires more time to send the same data, the STA may make a request for more admitted_time to the AP and at the same time downgrade the EDCA parameters for that AC for short intervals in order to send some of the traffic at the admitted priority and some at the unadmitted priority, while waiting for a response to the admission request.

9.19.4.3 Controlled-access admission control

This subclause describes the schedule management of the admitted HCCA streams by the HC. When the HC provides controlled channel access to STAs, it is responsible for granting or denying polling service to a TS based on the parameters in the associated TSPEC. If the TS is admitted, the HC is responsible for scheduling channel access to this TS based on the negotiated TSPEC parameters. The HC should not initiate a modification of TSPEC parameters of an admitted TS unless requested by the STA. The HC should not tear down a TS unless explicitly requested by the STA or at the expiry of the inactivity timer. The polling service based on admitted TS provides a “guaranteed channel access” from the scheduler in order to have its QoS requirements met. This is an achievable goal when the WM operates free of external interference (such as operation within the channel by other technologies and co-channel overlapping BSS interference). The nature of wireless communications may preclude absolute guarantees to satisfy QoS requirements. However, in a controlled environment (e.g., no interference), the behavior of the scheduler can be observed and verified to be compliant to meet the service schedule.

The normative behavior of the scheduler is as follows:

- The scheduler shall be implemented so that, under controlled operating conditions, all STAs with admitted TS are offered TXOPs that satisfy the service schedule.
- Specifically, if a TS is admitted by the HC, then the scheduler shall service the STA during an SP. An SP is a contiguous time during which a set of one or more downlink individually addressed frames and/or one or more polled TXOPs are granted to the STA. An SP starts at fixed intervals of time specified in Service Interval field. The first SP starts when the lower order 4 octets of the TSF

timer equals the value specified in Service Start Time field.[37] Additionally, the minimum TXOP duration shall be at least the time to transmit one maximum MSDU size successfully at the minimum PHY rate specified in the TSPEC. If maximum MSDU size is not specified in the TSPEC, then the minimum TXOP duration shall be at least the time to transmit one nominal MSDU size successfully at the minimum PHY rate. The vendors are free to implement any optimized algorithms, such as reducing the polling overheads, increasing the TXOP duration, etc., within the parameters of the transmitted schedule.

— The HC shall admit its request based on Infrastructure Authorization Information in dot11InterworkingEntry, which is part of the dot11InterworkingTable. The dot11InterworkingEntry specifies whether a non-AP STA is permitted to use HCCA, its throughput limitation and its minimum delay bound.

When the HC aggregates the admitted TS, it shall set the Aggregation field in the granted TSPEC to 1. An AP shall schedule the transmissions in HCCA TXOPs and communicate the service schedule to the STA. The HC shall provide an aggregate service schedule if the STA sets the Aggregation field in its TSPEC request. If the AP establishes an aggregate service schedule for a STA, it shall aggregate all HCCA streams for the STA. The service schedule is communicated to the STA in a Schedule element contained in an ADDTS Response frame. In the ADDTS Response frame, the modified service start time shall not exceed the requested service start time, if specified in ADDTS Request frame, by more than one maximum service interval (SI). The HC uses the maximum SI for the initial scheduling only as there may be situations that HC may not be able to service the TS at the scheduled timing, due to an EDCA or DCF transmission or other interferences interrupting the schedule. The Service Interval field value in the Schedule element shall be greater than the minimum SI. The service schedule could be subsequently updated by an AP as long as it meets TSPEC requirements.

The HC may update the service schedule at any time by sending a Schedule element in a Schedule frame. The updated schedule is in effect when the HC receives the ACK frame for the Schedule frame. The service start time in the Schedule element in the Schedule frame shall not exceed the beginning of the immediately previous SP by more than the maximum SI. The service start time shall not precede the beginning of the immediately previous SP by more than the minimum SI.

A STA may affect the service schedule by modifying or deleting its existing TS as specified in 10.4.

N.3.1 provides guidelines for deriving an aggregate service schedule for a single STA from the STA's admitted TS. The schedule shall meet the QoS requirements specified in the TSPEC.

During any time interval [$t1$, $t2$] including the interval that is greater than the specification interval, the cumulative TXOP duration shall be greater than the time required to transmit all MSDUs (of nominal MSDU size) arriving at the mean data rate for the stream, over the period [$t1$, $t2 - D$]. The parameter D is set to the specified maximum SI in the TSPEC. If maximum SI is not specified, then D is set to the delay bound in the TSPEC.

The HC shall use the minimum PHY rate in calculating TXOPs if the minimum PHY rate is present in the TSPEC field in the ADDTS response. Otherwise, the HC may use an observed PHY rate in calculating TXOPs. STAs may have an operational rate lower than the minimum PHY rate due to varying conditions on the channel for a short time and still may be able to sustain the TS without changing the minimum PHY rate in the TSPEC.

A minimum set of TSPEC parameters shall be specified during the TSPEC negotiation. The specification of a minimum set of parameters is required so that the scheduler can determine a schedule for the stream that is to be admitted. These parameters are Mean Data Rate, Nominal MSDU Size, Minimum PHY Rate, Surplus

[37] The lower order 4 octets of the TSF timer cover a span of around 71 min. Due to TSF timer wrapover and due to the possibility of receiving the schedule frame after the indicated start time timer, ambiguity may occur. This ambiguity is resolved by using the nearest absolute TSF timer value in past or future when the lower order 4 octets match the Start Time field in the Schedule element.

Bandwidth Allowance, and at least one of Maximum Service Interval and Delay Bound in the ADDTS Request frame. In the ADDTS Response frame, these parameters are Mean Data Rate, Nominal MSDU Size, Minimum PHY Rate, Surplus Bandwidth Allowance, and Maximum Service Interval and shall be nonzero when a stream is admitted.

If any of the elements in the minimum set of parameters does not have the required nonzero value, as specified above in this subclause, in the ADDTS Request frame, the HC may replace the unspecified parameters with nonzero values and admit the stream, or it may reject the stream. If the HC admits the stream with the alternative set of TSPEC parameters, these parameters are indicated to the STA through the ADDTS Response frame. If both maximum SI and delay bound are specified, the HC may use only the maximum SI. If any other parameter is specified in the TSPEC element, the scheduler may use it when calculating the schedule for the stream. The HC may also use the UP value in the TS Info field for admission control or scheduling purposes; however, this decision is outside the scope of this standard. The mandatory set of parameters might be set by any higher layer entity or may be generated autonomously by the MAC.

If a STA specifies a nonzero minimum SI and if the TS is admitted, the HC shall generate a schedule that conforms to the specified minimum SI.

A reference design for a sample scheduler and admission control unit is provided in Annex N. A sample use of the TSPEC for admission control is also described in Annex N.

9.20 Mesh coordination function (MCF)

9.20.1 General

Under MCF, the basic unit of allocation of the right to transmit onto the WM is the TXOP. Each TXOP is defined by a starting time and a defined maximum length.

There are two types of TXOP in MCF: EDCA TXOPs and MCCA TXOPs. The EDCA TXOP is obtained by a mesh STA winning an instance of EDCA contention (see 9.19.2). The MCCA TXOP is obtained by a mesh STA gaining control of the WM during an MCCAOP. The MCCAOP is an interval of time for frame transmissions that has been reserved by means of the exchange of MCCA frames (see 9.20.3). Neither EDCA TXOPs nor MCCA TXOPs shall exceed dot11MaxDwellTime (if using an FH PHY).

EDCA TXOPs of a mesh STA that has dot11MCCAActivated true shall not overlap with the time periods of any of its tracked MCCAOP reservations.

The process of tracking MCCAOP reservations involves the recording of the MCCAOP reservations and the data that structure the MCCAOP advertisements of these reservations, namely the advertisement set sequence number, advertisement elements bitmap, and the advertisement element indexes, in a local database, and the updating of this database on the basis of received advertisements as described in 9.20.3.7.5.

9.20.2 MCF contention-based channel access

MCF implements the same EDCA (see 9.19.2) as does HCF.

9.20.3 MCF controlled channel access (MCCA)

9.20.3.1 General

MCF controlled channel access (MCCA) is an optional access method that allows mesh STAs to access the WM at selected times with lower contention than would otherwise be possible. This standard does not require all mesh STAs to use MCCA. MCCA might be used by a subset of mesh STAs in an MBSS. However, MCCAOP reservations shall only be set up among mesh STAs that have dot11MCCAActivated true and that operate on the same channel. The performance of MCCA might be impacted by STAs that do not respect MCCAOP reservations.

MCCA enabled mesh STAs use management frames to make reservations for transmissions. The mesh STA transmitting an MCCA Setup Request frame to initiate a reservation becomes the MCCAOP owner of the MCCAOP reservation. The receivers of the MCCA Setup Request frame are the MCCAOP responders. The MCCAOP owner and the MCCAOP responders advertise this MCCAOP reservation to their neighbors via an MCCAOP advertisement. The MCCA enabled neighbor mesh STAs that could cause interference to transmissions during these reserved time periods, or that would experience interference from them, shall not initiate a transmission during these reserved time periods. During its MCCAOP, the MCCAOP owner obtains a TXOP by winning an instance of EDCA contention. Because of its reservation, the MCCAOP owner experiences no competition from other MCCA enabled neighbor mesh STAs. At the start of an MCCAOP, the EDCAF of the MCCAOP owner replaces the AIFSN, CWmin, and CWmax value of its dot11EDCATable with MCCA access parameters.

In order to use MCCA, a mesh STA maintains synchronization with its neighboring mesh STAs. Mesh STAs that use MCCA shall use a DTIM interval with a duration of $2^n \times 100$ TU with n being a non-negative integer less than or equal to 17. Additionally, a mesh STA shall track the reservations of its neighboring mesh STAs.

NOTE 1—The DTIM interval of this form was chosen so that the starting times of the reservations do not change relative to each other between consecutive DTIM intervals. The restriction that n be less than or equal to 17 was chosen for compatibility with the maximum DTIM interval as well as the compatibility of the reservation's MCCAOP offset range (see 8.4.2.108.2) with the maximal DTIM interval length.

NOTE 2—It is allowed that a different value for the DTIM interval is used for mesh STAs that use MCCA in an MBSS that is centrally controlled and the central authority provides a coordination of the DTIM interval of the mesh STAs that use MCCA in the MBSS.

9.20.3.2 MCCA activation

When it receives an MLME-ACTIVATEMCCA.request primitive from its SME, a mesh STA shall set the MCCA Enabled subfield of the Mesh Capability field in the Mesh Configuration element to 1 in Beacon and Probe Response frames it transmits. It shall not initiate or accept MCCA Setup Request frames for dot11MCCAScanDuration TUs after the receipt of the MLME-ACTIVATEMCCA.request primitive.

During the dot11MCCAScanDuration waiting period, the mesh STA learns its neighborhood MCCAOP periods by receiving Beacon, Probe Response, or MCCA Advertisement frames from neighboring mesh STAs.

After this period, the mesh STA may initiate and accept MCCA Setup Request frames as per 9.20.3.6.

9.20.3.3 MCCAOP reservations

An MCCAOP reservation specifies a schedule for frame transmissions. The time periods scheduled for frame transmissions in the reservation are called MCCAOPs. The schedule is set up between an MCCAOP owner and one (for individually addressed frames) or more (for group addressed frames) MCCAOP responders. MCCAOPs are set up by means of the procedure defined in 9.20.3.6. Once an MCCAOP reservation is set:

— Access to the channel by MCCA enabled mesh STAs is governed by the procedures in 9.20.3.9.

— The MCCAOP reservation is advertised according to the procedures in 9.20.3.7.

The schedule is defined by means of the MCCAOP Reservation field defined in 8.4.2.108.2. An MCCAOP reservation schedules a series of MCCAOPs with a common duration given in the MCCAOP Duration subfield of the MCCAOP Reservation field. This series is started after the first DTIM Beacon following the successful completion of the MCCAOP setup procedure and terminated when the MCCAOP reservation is torn down.

The reservation defines a regular schedule of MCCAOPs in the DTIM interval of the MCCAOP owner. The number of MCCAOPs in the DTIM interval is given by the value of the MCCAOP Periodicity subfield of the MCCAOP Reservation field. The MCCAOP Offset subfield specifies the offset of the first scheduled MCCAOP of the transmission schedule relative to the beginning of the DTIM interval of the MCCAOP owner. The following MCCAOPs are separated by a time interval with a duration equal to the length of the DTIM period divided by the value in the MCCAOP Periodicity subfield.

An example of an MCCAOP reservation schedule is shown in Figure 9-24. In this example, the MCCAOP Periodicity equals two, so that there are two MCCAOPs in each DTIM interval. As further illustrated in the figure, the MCCAOP Offset value indicates the beginning of the first MCCAOP in each DTIM interval.

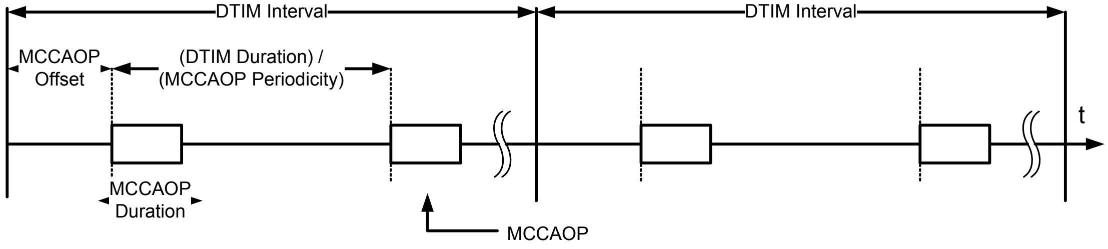


Figure 9-24—Example MCCAOP reservation with MCCAOP Periodicity equal to 2

If a mesh STA adjusts its TBTT, e.g., in response to a TBTT adjustment request, it shall adjust the MCCAOP reservations by modifying the MCCAOP Offset of each MCCAOP reservation.

An MCCAOP reservation is identified by an MCCAOP reservation ID. The MCCAOP owner shall select an MCCAOP reservation ID that is unique among all of its MCCAOP reservations. The MCCAOP reservation ID and MAC address of the MCCAOP owner uniquely identify the MCCAOP reservation in the mesh BSS. The MCCAOP reservation ID is an 8-bit unsigned integer and included in the MCCAOP Reservation ID field of an MCCAOP Setup Request element. If this MCCAOP setup request is for an individually addressed transmission, the MCCAOP Reservation ID is between 0 and 127. If this MCCAOP setup request is for a group addressed transmission, the MCCAOP Reservation ID is between 128 and 254. The value 255 is not used to identify a specific MCCAOP reservation but is reserved for usage in the MCCAOP teardown procedure as described in 9.20.3.8.

A mesh STA with dot11MCCAActivated equal to true shall be able to track at least dot11MCCAMinTrackStates MCCAOP reservations, including its own reservations. If the number of tracked MCCAOP reservations is less than dot11MCCAMaxTrackStates, the mesh STA shall be able to track, set up, and accept additional reservations. In this case, the mesh STA shall set the Accept Reservations subfield in the Flags field to 1 in the MCCAOP Advertisement Overview elements it transmits.

If the number of tracked MCCAOP reservations is equal to or greater than dot11MCCAMaxTrackStates, the mesh STA shall not track, set up, or accept additional reservations. In this case, the mesh STA shall set the Accept Reservations subfield in the Flags field to 0 in the MCCAOP Advertisement Overview elements it transmits. Moreover, it shall reply to MCCA Setup Request frames with an MCCA Setup Reply frame with the MCCA Reply Code field in the MCCAOP Setup Reply element equal to 3: Reject: MCCAOP track limit exceeded.

The tracked MCCAOP reservations are advertised as described in 9.20.3.7. How to access the medium during the tracked MCCAOP reservations is specified in 9.20.3.9.

9.20.3.4 Neighborhood MCCAOP periods at a mesh STA

The set of MCCAOP reservations in which a mesh STA is involved as an MCCAOP owner or an MCCAOP responder and that are used for individually addressed transmissions are referred to as the TX-RX periods of this mesh STA.

The set of MCCAOP reservations in which a mesh STA is involved as an MCCAOP owner or an MCCAOP responder and that are used for group addressed transmissions are referred to as the broadcast periods of this mesh STA. Optionally, the broadcast periods of a mesh STA includes known Target Beacon Transmission Time of Beacon frames for which this mesh STA is either the transmitter or the receiver, and transmission or reception periods of a STA that is collocated with the reporting mesh STA, for example, beacon or HCCA times of a collocated AP.

The interference periods of a mesh STA comprise the TX-RX periods and the broadcast periods of its neighbor mesh STAs in which the mesh STA is not involved as the owner or as a responder. The TX-RX periods, the broadcast periods, and the interference periods of a mesh STA shall not be used for a new MCCAOP reservation with the mesh STA as transmissions in these periods may experience interference from the transmissions in the new MCCAOPs or may cause interference to them.

The interference periods are directly derived from the TX-RX Periods Report field and Broadcast Periods Report field of the MCCAOP Advertisement elements transmitted by the neighbor mesh STAs. The Interference Periods Report reflects the latest TX-RX Periods Reports and Broadcast Periods Reports received from the neighbor mesh STAs.

The MCCAOP reservations of a mesh STA and its neighbors define a set of MCCAOPs that are already reserved for frame transmissions in the mesh neighborhood of a mesh STA. This set of MCCAOPs is referred to as the neighborhood MCCAOP periods for the mesh STA. Thus, neighborhood MCCAOP periods at a mesh STA include all MCCAOPs for which the mesh STA or one of its neighbors, including neighbors from other MBSSs, is either transmitter or receiver.

9.20.3.5 MCCA access fraction (MAF)

The MCCA access fraction at a mesh STA is the ratio of the time reserved for MCCAOPs in the DTIM interval of this mesh STA to the duration of the DTIM interval. This parameter is reported in the MCCA Access Fraction field of the MCCAOP Advertisement Overview elements. The maximum value for the MAF that is allowed at a mesh STA is specified by dot11MAFlimit. The dot11MAFlimit is copied into the MAF Limit field of the MCCAOP Advertisement Overview element as described in 8.4.2.110.

The MAF and the MAF Limit may be used to limit the use of MCCA in the mesh neighborhood of a mesh STA, as specified in 9.20.3.6. Before attempting to set up an MCCAOP reservation with a neighbor peer mesh STA, a mesh STA shall verify that the new MCCAOP reservation does not cause its MAF to exceed its MAF Limit and that the new MCCAOP reservation does not cause the MAF of any of its neighbor peer mesh STAs to exceed their MAF Limit. An MCCAOP setup request shall be refused by the intended MCCAOP responder if the MAF limit of one of its neighbors is exceeded due to the new setup.

9.20.3.6 MCCAOP setup procedure

The setup of an MCCAOP reservation is initiated by the MCCAOP owner, and is accepted or rejected by the MCCAOP responder. The setup procedure for an MCCAOP reservation is as follows:

a) The MCCAOP owner shall build a map of the neighborhood MCCAOP periods in the DTIM interval after hearing advertisements from all of its neighbor mesh STAs with the MCCA Enabled subfield of the Mesh Capability field in the Mesh Configuration element equal to 1. It shall request an MCCAOP advertisement, as described in 9.20.3.7.8, from each neighbor mesh STA from which no advertisement was heard in the last dot11MCCAAdvertPeriodMax DTIM intervals.

b) The MCCAOP owner shall determine the MCCAOP reservation. The MCCAOP parameters shall be chosen in such a way that they satisfy the following conditions:
   1) The reservation shall not overlap with the neighborhood MCCAOP periods of the MCCAOP owner.
   2) The reservation shall not overlap with the interference periods of the intended MCCAOP responder or responders.
   3) The reservation shall not cause the MAF limit to be exceeded for either itself or its neighbor mesh STAs.
   4) The Accept Reservations subfield of the Flags field equals 1 in the most recent MCCAOP Advertisement Overview element received from all intended MCCAOP responders.

c) If the conditions in item b) are satisfied, the MCCAOP owner shall transmit an MCCAOP Setup Request element to the intended MCCAOP responder with the chosen MCCAOP parameters.

d) The MCCAOP responder shall verify the following conditions:
   1) The reservation does not overlap with its neighborhood MCCAOP periods.
   2) The reservation does not cause the MAF limit to be exceeded for itself or its neighbor mesh STAs.
   3) The number of reservations in its neighborhood MCCAOP periods does not exceed dot11MCCAMaxTrackStates.

e) If the conditions in item d) are satisfied, the responder shall send an MCCA Setup Reply frame to the MCCAOP owner with the MCCA Reply Code field in the MCCAOP Setup Reply element equal to 0: Accept, as defined in Table 8-183.

f) If the conditions in item d) are satisfied and the MCCAOP request has been intended for group addressed transmissions, the responder shall include the reservation in its MCCAOP advertisement only after the MCCAOP advertisement from the MCCAOP owner is received.

g) If not all of the conditions in item d) are satisfied and the MCCAOP request is intended for individually addressed transmissions, the responder shall transmit to the MCCAOP owner an MCCA Setup Reply frame that is constructed as follows:
   1) If the condition in item d)1) is not satisfied and both conditions in item d)2) and item d)3) are satisfied, the responder may calculate an alternative MCCAOP reservation and include it in the MCCAOP Reservation field of the MCCAOP Setup Reply element. It shall set the MCCA Reply Code field of the MCCAOP Setup Reply element to 1: Reject: MCCAOP reservation conflict, as defined in Table 8-183.
   2) If the condition in item d)2) is not satisfied, it shall set the MCCA Reply Code field of the MCCAOP Setup Reply element to 2: Reject: MAF limit exceeded, as defined in Table 8-183.
   3) If the condition in item d)2) is satisfied and the condition in item d)3) is not satisfied, it shall set the MCCA Reply Code field of the MCCAOP Setup Reply element to 3: Reject: MCCAOP track limit exceeded, as defined in Table 8-183.

h) If not all of the conditions in item d) are satisfied and the MCCAOP request is intended for group addressed transmissions, the responder shall send an MCCA Setup Reply frame to the MCCAOP owner with the MCCA Reply Code field in the MCCAOP Setup Reply element equal to 1: Reject: MCCAOP reservation conflict.

i) If the MCCAOP owner receives an MCCA Setup Reply frame with MCCA Reply Code equal to Accept, the MCCAOP reservation is established. Otherwise, the mesh STA may repeat the MCCAOP setup procedure using a modified MCCAOP Setup Request. If an alternative MCCAOP reservation is included in the MCCAOP Setup Reply element, the mesh STA may consider this alternative in its modified MCCAOP Setup Request.

9.20.3.7 MCCAOP advertisement

9.20.3.7.1 General

A mesh STA with dot11MCCAActivated equal to true tracks MCCAOP reservations. The tracked MCCAOP reservations contain the neighborhood MCCAOP periods and optionally other periodic transmission of itself or of neighboring STAs.

The MCCAOP advertisement set contains all MCCAOP reservations tracked by the mesh STA. The MCCAOP advertisement set is represented by an MCCAOP Advertisement Overview element and zero (if the MCCAOP advertisement set is empty) or more (if the MCCAOP advertisement set is nonempty) MCCAOP Advertisement elements. An MCCAOP Advertisement element contains one or more tracked MCCAOP reservations.

The mesh STA advertises its MCCAOP advertisement set to its neighbor mesh STAs.

This subclause describes how the mesh STA constructs the MCCAOP Advertisement Overview element and the MCCAOP Advertisement elements. Further, this subclause describes the procedure to advertise an MCCAOP advertisement set, the procedure to request an MCCAOP advertisement from a neighboring mesh STA, and the procedure to process a received MCCAOP advertisement.

9.20.3.7.2 Construction of an MCCAOP advertisement set

Each MCCAOP reservation tracked by a mesh STA is one of the following types:

- MCCAOP TX-RX period:
  - An MCCAOP reservation for individually addressed frames for which the mesh STA is the MCCAOP owner or the MCCAOP responder.
- MCCAOP broadcast period:
  - An MCCAOP reservation for group addressed frames for which the mesh STA is the MCCAOP owner or the MCCAOP responder.
  - Optionally, a known Target Beacon Transmission Time of Beacon frames for which the mesh STA is either the transmitter or the receiver.
  - Optionally, a transmission or reception period of a STA that is collocated with the mesh STA, for example, beacon or HCCA times of a collocated AP.
- MCCAOP interference period:
  - A TX-RX or a broadcast period reported by a neighbor peer mesh STAs of the mesh STA excluding those periods for which this mesh STA is either the MCCAOP owner or the MCCAOP responder.
  - Optionally, a TX-RX or a broadcast period reported by neighbor nonpeer mesh STAs of the mesh STA.

The MCCAOP reservations are grouped into the following sets:

- MCCAOP TX-RX advertisement set

- MCCAOP broadcast advertisement set
- MCCAOP interference advertisement set

These three sets constitute the MCCAOP advertisement set. The mesh STA uses the MCCAOP Overview element and MCCAOP Advertisement elements to advertise its MCCAOP advertisement set to its neighbor mesh STAs.

The mesh STA acts as follows to construct the MCCAOP Overview elements and the MCCAOP Advertisement elements:

a) If the MCCAOP advertisement set is nonempty, the mesh STA constructs one or more MCCAOP reports according to the format described in 8.4.2.111.3 as follows:
    1) If the MCCAOP TX-RX advertisement set is nonempty, the mesh STA constructs one or more TX-RX reports according to the format described in 8.4.2.111.3 such that each reservation in the MCCAOP TX-RX advertisement set occurs exactly in one TX-RX report.
    2) If the MCCAOP broadcast advertisement set is nonempty, the mesh STA constructs one or more broadcast reports according to the format described in 8.4.2.111.3 such that each reservation in the MCCAOP broadcast advertisement set occurs exactly in one broadcast report.
    3) If the MCCAOP interference advertisement set is nonempty, the mesh STA constructs one or more interfering reports according to the format described in 8.4.2.111.3 such that each reservation in the MCCAOP interference advertisement set occurs exactly in one interfering report.

b) If the MCCAOP advertisement set is nonempty, the mesh STA constructs one or more MCCAOP Advertisement elements as follows:
    1) The MCCAOP Advertisement Set Sequence Number field is set to the MCCAOP advertisement set sequence number as explained in 9.20.3.7.3.
    2) The MCCAOP Advertisement Element Index subfield is set to an identifier that uniquely identifies the MCCAOP Advertisement element in the MCCAOP advertisement set.
    3) Each MCCAOP Advertisement element includes at least one of the TX-RX reports, broadcast reports, or interfering reports. Moreover, it includes at most one of the TX-RX reports, at most one of the broadcast reports, and at most one of the interfering reports. In case the MCCAOP Advertisement element contains a TX-RX report, the TX-RX Report Present subfield of the MCCAOP Advertisement Element Information field is set to 1; otherwise this subfield is set to 0. In case the MCCAOP Advertisement element contains a broadcast report, the Broadcast Report Present subfield of the MCCAOP Advertisement Element Information field is set to 1; otherwise this subfield is set to 0. In case the MCCAOP Advertisement element contains an interfering report, the Interference Report Present subfield of the MCCAOP Advertisement Element Information field is set to 1; otherwise, this subfield is set to 0.
    4) Each report as constructed in step a) is present in exactly one MCCAOP Advertisement element.

c) The mesh STA constructs one MCCAOP Advertisement Overview element such that
    1) The MCCAOP Advertisement Set Sequence Number field is set to the advertisement set sequence number as explained in 9.20.3.7.3.
    2) The Medium Access Fraction field is set to the medium access fraction.
    3) The MAF limit field is set to the value of dot11MAFlimit.
    4) The Accept Reservations field is set to 1 if the number of tracked reservations of this mesh STA is less than dot11MCCAMaxTrackStates, and set to 0 otherwise.
    5) Bit *i* of the Advertisement Elements Bitmap field is set to 1 if an MCCAOP Advertisement element with the MCCAOP Advertisement Element Index subfield equal to *i* is part of the representation of this MCCAOP advertisement set, and set to 0 otherwise.

9.20.3.7.3 Setting the MCCAOP advertisement set sequence number

The MCCAOP advertisement set sequence number identifies an MCCAOP advertisement set. Mesh STAs with dot11MCCAActivated equal to true assign MCCAOP advertisement set sequence numbers from a single modulo-256 counter. The MCCAOP advertisement set sequence number is initialized to 0. The MCCAOP advertisement set sequence number shall be incremented by 1 if one of the following conditions holds:

a) The mesh STA sets the bit for an MCCAOP Advertisement element in the Advertisement Elements Bitmap from 0 to 1 and this bit has been set to 1 under the same MCCAOP Advertisement Sequence Number before.

b) The bit of the Advertisement Elements Bitmap corresponding to an MCCAOP Advertisement element is equal to 1 and the content of this MCCAOP Advertisement element changes.

However, the MCCAOP advertisement set sequence number may remain unchanged if

— The mesh STA changes a bit in the Advertisement Element Bitmap from 0 to 1 and this bit has not been set to 1 under the same MCCAOP Advertisement Sequence Number before, or

— The mesh STA changes a bit in the Advertisement Elements Bitmap from 1 to 0.

NOTE—The Advertisement Set Sequence Number identifies the current distribution of the MCCAOP advertisement set over the MCCAOP Advertisement elements. Using a new MCCAOP advertisement set sequence number signals a new, (possibly) completely different distribution of the MCCAOP advertisement set over the MCCAOP Advertisement elements, and requires an advertisement of all reservations of the MCCAOP advertisement set. Leaving the MCCAOP advertisement set sequence number unchanged as in the previous MCCAOP Advertisement Overview element indicates MCCAOP Advertisement elements that have previously been advertised are not changed and remain current. This enables a limited advertisement procedure in which only new MCCAOP Advertisement elements are advertised. Additionally, this enables mesh STAs that operate in light or deep sleep mode to request a limited update of the MCCAOP advertisement set of a neighboring mesh STA in which only new MCCAOP Advertisement elements are included.

9.20.3.7.4 Advertisement procedure

To advertise its MCCAOP advertisement set, the mesh STA constructs a representation of the MCCAOP advertisement set as described in 9.20.3.7.2. The MCCAOP advertisement set is advertised by transmitting an MCCAOP Advertisement Overview element and zero or more MCCAOP Advertisement elements (see 9.20.3.7.2) to neighbor peer mesh STAs. The MCCAOP Advertisement Overview element and the MCCAOP Advertisement elements are transmitted in Beacon frames, Probe Response frames, or MCCA Advertisement frames.

The mesh STA shall advertise its MCCAOP advertisement set according to the following rules:

a) The mesh STA shall advertise at least one MCCAOP Advertisement Overview element in every dot11MCCAAdvertPeriodMax DTIM intervals.

b) The mesh STA shall advertise its MCCAOP Advertisement Overview element and any new MCCAOP Advertisement elements at the latest with the transmission of its next Beacon frame after its MCCAOP advertisement set has changed.

c) The mesh STA shall advertise the requested MCCAOP Advertisement elements as described in 9.20.3.7.8 if the mesh STA receives an MCCA Advertisement Request frame.

9.20.3.7.5 Receipt of an MCCAOP advertisement

Upon receipt of an MCCAOP advertisement a mesh STA with dot11MCCAActivated shall compare the Advertisement Set Sequence Number contained in the MCCAOP Advertisement Overview element of the received MCCAOP advertisement with the last advertisement set sequence number that this mesh STA tracked for the sender of the received MCCAOP advertisement.

If the tracked advertisement set sequence number does not equal the Advertisement Set Sequence Number of the received MCCAOP advertisement, the mesh STA shall perform the procedure described in 9.20.3.7.6.

If the tracked advertisement set sequence number equals the Advertisement Set Sequence Number of the received MCCAOP advertisement, the mesh STA shall compare the Advertisement Elements Bitmap contained in the received MCCAOP Advertisement Overview element with the last Advertisement Elements Bitmap that this mesh STA tracked for the sender of the received MCCAOP advertisement. If the tracked Advertisement Elements Bitmap does not equal the Advertisement Elements Bitmap of the received MCCAOP advertisement, the mesh STA shall perform the procedure described in 9.20.3.7.7.

NOTE—If both the tracked advertisement set sequence number equals the Advertisement Set Sequence Number of the received MCCAOP advertisement and the tracked Advertisement Elements Bitmap equals the Advertisement Elements Bitmap of the received MCCAOP advertisement, the MCCAOP advertisement set of the sender of the MCCAOP advertisement tracked by the mesh STA is current, and no update of the MCCAOP advertisement set is needed.

9.20.3.7.6 Complete update of the tracked MCCAOP reservations of a neighbor mesh STA

The mesh STA performed the steps in 9.20.3.7.5 and detected that the MCCAOP advertisement set sequence number has been updated. Consequently, the mesh STA shall operate as follows.

The mesh STA shall discard all MCCAOP reservations that it tracked for the sender of the received MCCAOP advertisement. The mesh STA shall record the Advertisement Set Sequence Number and the source address (SA) of the received MCCAOP advertisement. The mesh STA shall record all reservations in the MCCAOP Advertisement elements of the received MCCAOP advertisement.

If the mesh STA does not receive all MCCAOP Advertisement elements of the sender of the MCCAOP advertisement before a frame exchange sequence on the wireless medium causes the mesh STA to set its NAV, the mesh STA shall perform the MCCAOP advertisement request procedure as described in 9.20.3.7.8.

9.20.3.7.7 Partial update of the tracked MCCAOP reservations of a neighbor mesh STA

The mesh STA performed the steps in 9.20.3.7.5 and detected that part of the MCCAOP advertisement set of the sender of the MCCAOP advertisement has been updated. Consequently, the mesh STA shall operate as follows for each bit in the Advertisement Elements Bitmap contained in the MCCAOP Advertisement Overview element of the received MCCAOP advertisement.

If the bit in position n of the Advertisement Elements Bitmap in the received MCCAOP Advertisement is equal to 0 and if the bit in position n of the Advertisement Elements Bitmap tracked for the sender of the received MCCAOP advertisement is equal to 1, the mesh STA shall delete the reservations with the same Advertisement Sequence Number and the same MCCAOP Advertisement Element Index received from the same sender from its tracked reservations.

If the bit in position n of the Advertisement Elements Bitmap in the received MCCAOP Advertisement is equal to 1 and if the bit in position n of the Advertisement Elements Bitmap tracked for the sender of the received MCCAOP advertisement is equal to 0, the mesh STA shall add the reservations of the received MCCAOP Advertisement element with the MCCAOP Advertisement Element Index set to n to its tracked reservations. If the mesh STA does not receive this MCCAOP Advertisement element of the sender of the MCCAOP Advertisement before a frame exchange sequence on the wireless medium causes the mesh STA to set its NAV, the mesh STA shall perform the MCCAOP Advertisement request procedure as described in 9.20.3.7.8.

NOTE—If the bit in position n of the received Advertisement Elements Bitmap contained in the received MCCAOP Advertisement is equal to the bit in position n of the Advertisement Elements Bitmap tracked for the sender of the received MCCAOP advertisement, then the Advertisement element with the MCCAOP Advertisement Element Index equal to n is current, and no update of this Advertisement element is needed.

9.20.3.7.8 MCCAOP advertisement request procedure

To request all MCCAOP Advertisement elements from a neighbor peer mesh STA, the mesh STA transmits an MCCA Advertisement Request frame without an MCCAOP Advertisement Overview element.

To request a subset of the MCCAOP Advertisement elements of a neighbor peer mesh STA, the mesh STA transmits an MCCA Advertisement Request frame including an MCCAOP Advertisement Overview element. The mesh STA shall set the contents of the MCCAOP Advertisement Overview element as follows. The mesh STA sets

a) The Advertisement Set Sequence Number field to the Advertisement Sequence Number that it tracks for the recipient of this frame
b) In the Advertisement Element Bitmap, the bit to 1 for each MCCAOP Advertisement Element that the mesh STA requests from the recipient of this frame
c) The Flags field, the MCCA Access Fraction field, and the MAF Limit field to zero

The mesh STA shall discard the MCCA Advertisement Request frame from its frame queue if it receives all of the MCCAOP Advertisement elements that it requests in the MCCAOP Advertisement Request.

9.20.3.8 MCCAOP teardown

9.20.3.8.1 Conditions that trigger an MCCAOP teardown

The MCCAOP owner and the MCCAOP responder may initiate a teardown of an MCCAOP reservation, e.g., when the reservation is no longer needed. A mesh STA shall act as follows to resolve conflicts between MCCAOP reservations in its neighborhood MCCAOP periods. If the conflict is caused by overlapping reservations from its TX-RX periods and broadcast periods, it shall select one of these reservations and initiate a teardown for it. If the conflict is caused by an overlap between a reservation from its TX-RX periods or broadcast periods, and another reservation from its interference periods, it shall act as follows. It creates a first unsigned integer by inverting the bit order of its MAC address and a second unsigned integer by inverting the bit order of the lowest of the known MAC addresses of the owner and responder(s) of the reservation in the interference periods. If the first unsigned integer is smaller than the second unsigned integer, it shall initiate a teardown of the reservation in its TX-RX or broadcast periods. Otherwise, it may initiate a teardown of the reservation in its TX-RX or broadcast periods.

There are also other conditions that trigger the MCCAOP owner and responder to delete a reservation, without an explicit tear down. An MCCAOP owner shall delete a reservation for an individually addressed transmission when it has not received an acknowledgement for any frame transmission in the MCCAOPs corresponding to the reservation for greater than dot11MCCAOPtimeout time. An MCCAOP responder shall delete a reservation for individually addressed transmission or group addressed transmissions when it has not received a frame transmission in any of the MCCAOPs corresponding to the reservation for greater than dot11MCCAOPtimeout time.

9.20.3.8.2 MCCAOP teardown procedure

The teardown is initiated by transmitting an MCCA Teardown frame. The MCCAOP Reservation ID field in the MCCAOP Teardown element is set to the MCCAOP Reservation ID of the reservation that is to be torn down. In case the tear down is initiated by an MCCAOP responder, the MCCAOP Owner field of the MCCAOP Teardown element is set to the MAC address of the MCCAOP owner.

The transmitter of the MCCA Teardown frame deletes the reservation after the MCCA Teardown frame has been successfully transmitted. The receiver of the MCCA Teardown frame acts as follows. In case the MCCAOP Reservation ID field corresponds to a reservation for individually addressed transmissions, it deletes the reservation. If the reservation is for group addressed transmissions for which it is the MCCAOP owner, it deletes the reservation if there are no other MCCAOP responders for this reservation.

The MCCAOP owner acts as follows when deleting a reservation:

- It stops executing the access procedure described in 9.20.3.9.1 at the start of the MCCAOPs corresponding to the reservation that was deleted.
- In case the reservation was for individually addressed frames, it stops advertising the MCCAOP reservation in its TX-RX Periods Report.
- In case the reservation was for group addressed frames, it stops advertising the MCCAOP reservation in its Broadcast Periods Report.

The MCCAOP responder acts as follows when deleting a reservation:

- It stops executing the procedure described in 9.20.3.9.2 during the MCCAOPs corresponding to the reservation that was deleted.
- In case the reservation was for individually addressed frames, it stops advertising the MCCAOP reservation in its TX-RX Periods Report.
- In case the reservation was for group addressed frames, it stops advertising the MCCAOP reservation in its Broadcast Periods Report.

9.20.3.9 Access during MCCAOPs

9.20.3.9.1 Access by MCCAOP owners

At the start of the MCCAOP, the EDCAF of the MCCAOP owner shall set AIFSN[AC] equal to dot11MCCAAIFSN, CWmax[AC] equal to dot11MCCACWmax, CW[AC] equal to dot11MCCACWmin, QSRC[AC] to 0, and QLRC[AC] to 0 for all ACs. The TXOP limit shall specify a duration value no larger than the MCCAOP Duration.

During the MCCAOP, the EDCAFs of the ACs operates as specified in 9.19.2, with the following modifications.

- During the MCCAOP, the EDCAF of each AC shall consider only those frame whose RA matches the MAC address of the MCCAOP responder.
- In cases where the access to the medium is delayed, the TXOPlimit value shall specify a duration to end no later than the MCCAOP start time plus the MCCAOP Duration.
- As specified in 9.20.3.9.2, neighboring STAs shall not access the WM during an MCCAOP, until they receive a frame from either the MCCAOP owner or the MCCAOP responder. With the exception of truncation of an MCCA TXOP by means of a CF-End, standard EDCA TXOP rules apply for the remainder of the MCCAOP. For HT mesh STAs, these include the reverse direction protocol as specified in 9.15.
- At the end of the MCCAOP, the parameters used by the EDCAF of the MCCAOP owner shall be set to the MIB attribute table dot11EDCATable, and QSRC[AC] and QLRC[AC] shall be set to 0 for all ACs.

The MCCAOP owner may adjust the duration of an MCCAOP by setting the Duration/ID field in the frames it transmits. In particular, if an MCCAOP owner has no data to transmit in an MCCAOP corresponding to an MCCAOP reservation that is intended for individually addressed frames, it may transmit an individually addressed QoS Null frame during the MCCAOP to end the MCCAOP.

NOTE—It is recommended to send a QoS Null frame to end the MCCAOP although there might be situations in which the transmission of a QoS Null is not needed or undesirable.

If an MCCAOP owner has no data to transmit in an MCCAOP reservation that is intended for group addressed frames, it may transmit a group addressed QoS Null frame during the MCCAOP to end the MCCAOP.

9.20.3.9.2 Access during an MCCAOP by mesh STAs that are not the MCCAOP owner

The MAC of a mesh STA with dot11MCCAActivated is true shall provide a Reservation Allocation Vector (RAV) mechanism to indicate a busy medium from the start of an MCCAOP corresponding to a reservation in its interference periods until the receipt of a frame transmitted by either the MCCAOP owner or the MCCAOP responder. The RAV mechanism is provided in addition to the PHY and virtual CS mechanisms described in 9.3.2.1. It is different from the virtual CS mechanism in two aspects. Firstly, a mesh STA might be neighbor to multiple ongoing MCCAOPs corresponding to different reservations and the regular NAV setting and updating rules do not suffice to prevent interference during these reservations. Secondly, the virtual CS mechanism is set immediately upon receipt of a frame, whereas the RAV mechanism is based on reservation frames received at some earlier time instant. When either the CS function provided by the PHY, the virtual CS function provided by the MAC via the NAV, or the RAV mechanism indicate a busy medium during an MCCAOP for which the mesh STA is neither the MCCAOP owner nor the MCCAOP responder, the medium shall be considered busy; otherwise, it shall be considered idle.

The RAV mechanism maintains an index of future MCCAOPs based on the reservation information that is available in the interference periods of a mesh STA. At the start of each MCCAOP corresponding to a reservation in the interference periods, a RAV is set to indicate a busy medium for the duration of the MCCAOP given in the MCCAOP Duration subfield of the MCCAOP reservation. At the start of each MCCAOP corresponding to a reservation in the TX-RX or broadcast periods for which the mesh STA is an MCCAOP responder, a RAV is set to indicate a busy medium for the duration of the MCCAOP given in the MCCAOP Duration subfield of the MCCAOP reservation. The RAV may be thought of as a counter, corresponding to an MCCAOP corresponding to a reservation in the interference periods. The RAV counts down to zero at a uniform rate. When the counter is zero, the RAV indication is that the medium is idle; when nonzero, the indication is busy.

The mesh STA clears the RAV timer, i.e., sets it to 0, upon receipt of a frame from either the MCCAOP owner or responder. If a mesh STA receives an RTS frame during an MCCAOP for which it is a MCCAOP responder, with the RA address matching its MAC address and with the MAC address in the TA field in the RTS frame matching the MAC address of the MCCAOP owner, then the STA shall send the CTS frame after SIFS, without regard for the NAV and the RAV, and without resetting its NAV. The RAV for an MCCAOP is not cleared upon receipt of a frame originating from stations that are not the MCCAOP owner or responder. Since the NAV is set upon receipt of frames with a Duration/ID field, the MCCAOP owner and responder adjust the reservation period of an MCCAOP to their actual traffic needs by the Duration/ID field in the transmitted frame and obtain protection of the frame transmission via the NAV setting.

The RAV mechanism might be represented by a number of counters, where each counter corresponds to one MCCAOP. The number of counters needed at any instant is equal to the number of MCCAOPs at this instant corresponding to reservations in the interference periods of the mesh STA.

9.20.3.10 Interaction with time synchronization

If a mesh STA adjusts its TBTT, e.g., in response to a TBTT Adjustment Request, it shall adjust the reservations by modifying the MCCAOP Offset of each of the tracked MCCAOP reservations. If a mesh STA adjusts its timing offset value with respect to a neighbor mesh STA, as specified in 13.13.2.2, it shall adjust the reservations by modifying the MCCAOP Offset of each of the tracked MCCAOP reservations for which this neighbor mesh STA is the owner. In either case, an MCCAOP advertisement of a mesh STA shall always contain the most recent MCCAOP Offsets.

9.21 Block Acknowledgment (Block Ack)

9.21.1 Introduction

The Block Ack mechanism improves channel efficiency by aggregating several acknowledgments into one frame. There are two types of Block Ack mechanisms: immediate and delayed. Immediate Block Ack is suitable for high-bandwidth, low-latency traffic while the delayed Block Ack is suitable for applications that tolerate moderate latency.[38] In this subclause, the STA with data to send using the Block Ack mechanism is referred to as the *originator*, and the receiver of that data as the *recipient*.

The Block Ack mechanism is initialized by an exchange of ADDBA Request/Response frames. After initialization, blocks of QoS data frames may be transmitted from the originator to the recipient. A block may be started within a polled TXOP or by winning EDCA contention. The number of frames in the block is limited, and the amount of state that is to be kept by the recipient is bounded. The MPDUs within the block of frames are acknowledged by a BlockAck frame, which is requested by a BlockAckReq frame.

The Block Ack mechanism does not require the setting up of a TS; however, QoS STAs using the TS facility may choose to signal their intention to use Block Ack mechanism for the scheduler's consideration in assigning TXOPs. Acknowledgments of frames belonging to the same TID, but transmitted during multiple TXOPs, may also be combined into a single BlockAck frame. This mechanism allows the originator to have flexibility regarding the transmission of data MPDUs. The originator may split the block of frames across TXOPs, separate the data transfer and the Block Ack exchange, and interleave blocks of MPDUs carrying all or part of MSDUs or A-MSDUs for different TIDs or RAs.

Figure 9-25 illustrates the message sequence chart for the setup, data and Block Ack transfer, and the teardown of the Block Ack mechanism, which are discussed in detail in 9.21.2 to 9.21.5.

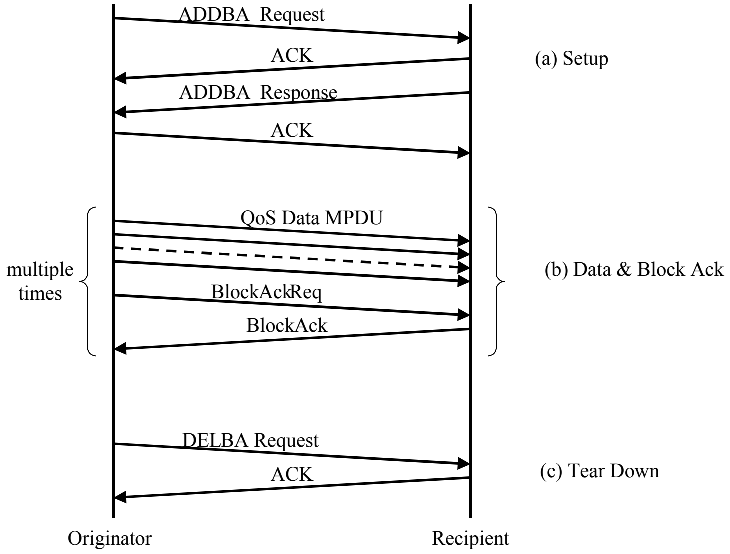


Figure 9-25—Message sequence chart for Block Ack mechanism: (a) setup, (b) data and acknowledgment transfer and (c) tear down

All operations on sequence numbers are performed modulo 2^{12}. Comparisons between sequence numbers are circular modulo 2^{12}, i.e., the sequence number space is considered divided into two parts, one of which is "old" and one of which is "new," by means of a boundary created by adding half the sequence number range to the current start of receive window (modulo 2^{12}).

[38] The delayed Block Ack mechanism is primarily intended to allow existing implementations to use this feature with minimal hardware changes and also to allow inexpensive implementations that would use the processing power on the host.

9.21.2 Setup and modification of the Block Ack parameters

An originator that intends to use the Block Ack mechanism for the transmission of QoS data frames to an intended recipient should first check whether the intended recipient STA is capable of participating in Block Ack mechanism by discovering and examining its Delayed Block Ack and Immediate Block Ack capability bits. If the intended recipient STA is capable of participating, the originator sends an ADDBA Request frame indicating the TID for which the Block Ack is being set up. For an ADDBA set up between STAs where one is a non-HT STA, the Block Ack Policy and Buffer Size fields in the ADDBA Request frame are advisory and may be changed by the recipient. The Buffer Size field in the ADDBA Request frame is advisory and may be changed by the recipient for an ADDBA set up between HT STAs.

The recipient STA shall respond by an ADDBA Response frame. The recipient STA has the option of accepting or rejecting the request. When the recipient STA accepts, then a Block Ack agreement exists between the originator and recipient.

When the recipient STA accepts, it indicates the type of Block Ack and the number of buffers that it shall allocate for the support of this Block Ack agreement within the ADDBA Response frame. Each Block Ack agreement that is established by a STA may have a different buffer allocation. If the intended recipient STA rejects the request, then the originator shall not use the Block Ack mechanism.

When the Block Ack Policy subfield value is set to 1 by the originator of an ADDBA Request frame between HT STAs, then the ADDBA Response frame accepting the ADDBA Request frame shall contain 1 in the Block Ack Policy subfield.

When a Block Ack agreement is established between two HT STAs, the originator may change the size of its transmission window if the value in the Buffer Size field of the ADDBA Response frame is larger than the value in the ADDBA Request frame. If the value in the Buffer Size field of the ADDBA Response frame is smaller than the value in the ADDBA Request frame, the originator shall change the size of its transmission window (WinSizeO) so that it is not greater than the value in the Buffer Size field of the ADDBA Response frame and is not greater than the value 64.

The A-MSDU Supported field indicates whether an A-MSDU may be sent under the particular Block Ack agreement. The originator sets this field to 1 to indicate that it might transmit A-MSDUs with this TID. The recipient sets this field to 1 to indicate that it is capable of receiving an A-MSDU with this TID.

NOTE—The recipient is free to respond with any setting of the A-MSDU supported field. If the value in the ADDBA Response frame is not acceptable to the originator, it can delete the Block Ack agreement and transmit data using normal acknowledgment.

If the Block Ack mechanism is being set up for a TS, bandwidth negotiation (using ADDTS Request and Response frames) should precede the setup of the Block Ack mechanism.

Once the Block Ack exchange has been set up, data and ACK frames are transferred using the procedure described in 9.21.3.

9.21.3 Data and acknowledgment transfer using immediate Block Ack policy and delayed Block Ack policy

After setting up either an immediate Block Ack agreement or a Delayed Block Ack agreement following the procedure in 9.21.2, and having gained access to the medium and established protection, if necessary, the originator may transmit a block of QoS data frames separated by SIFS period, with the total number of frames not exceeding the Buffer Size subfield value in the associated ADDBA Response frame and subject to any additional duration limitations based on the channel access mechanism. Each of the frames shall have the Ack Policy subfield in the QoS Control field set to Block Ack. The RA field of the frames shall be the recipient's individual address. The originator requests acknowledgment of outstanding QoS data frames by sending a Basic BlockAckReq frame. The recipient shall maintain a Block Ack record for the block.

Subject to any constraints in this subclause about permitted use of TXOP according to the channel access mechanism used, the originator may

- Separate the Block and Basic BlockAckReq frames into separate TXOPs
- Split a Block frame across multiple TXOPs
- Split transmission of data MPDUs sent under Block Ack policy across multiple TXOPs
- Interleave MPDUs with different TIDs within the same TXOP
- Sequence or interleave MPDUs for different RAs within a TXOP

A protective mechanism (such as transmitting using HCCA, RTS/CTS, or the mechanism described in 9.23) should be used to reduce the probability of other STAs transmitting during the TXOP. If no protective mechanism is used, then the first frame that is sent as a block shall have a response frame and shall have the Duration field set so that the NAVs are set to appropriate values at all STAs in the BSS.

The originator shall use the Block Ack starting sequence control to signal the first MPDU in the block for which an acknowledgment is expected. MPDUs in the recipient's buffer with a sequence control value that precedes the starting sequence control value are called *preceding MPDUs*. The recipient shall reassemble any complete MSDUs from buffered preceding MPDUs and indicate these to its higher layer. The recipient shall then release any buffers held by preceding MPDUs. The range of the outstanding MPDUs (i.e., the reorder buffer) shall begin on an MSDU boundary. The total number of frames that can be sent depends on the total number of MPDUs in all the outstanding MSDUs. The total number of MPDUs in these MSDUs may not exceed the reorder buffer size in the receiver.

The recipient shall maintain a Block Ack record consisting of originator address, TID, and a record of reordering buffer size indexed by the received MPDU sequence control value. This record holds the acknowledgment state of the data frames received from the originator.

If the immediate Block Ack policy is used, the recipient shall respond to a Basic BlockAckReq frame with a Basic BlockAck frame. If the recipient sends the Basic BlockAck frame, the originator updates its own record and retries any frames that are not acknowledged in the Basic BlockAck frame, either in another block or individually.

If the delayed Block Ack policy is used, the recipient shall respond to a Basic BlockAckReq frame with an ACK frame. The recipient shall then send its Basic BlockAck response in a subsequently obtained TXOP. Once the contents of the Basic BlockAck frame have been prepared, the recipient shall send this frame in the earliest possible TXOP using the highest priority AC. The originator shall respond with an ACK frame upon receipt of the Basic BlockAck frame. If delayed Block Ack policy is used and if the HC is the recipient, then the HC may respond with a +CF-Ack frame if the Basic BlockAckReq frame is the final frame of the polled TXOP's frame exchange. If delayed Block Ack policy is used and if the HC is the originator, then the HC may respond with a +CF-Ack frame if the Basic BlockAck frame is the final frame of the TXOP's frame exchange.

The Basic BlockAck frame contains acknowledgments for the MPDUs of up to 64 previous MSDUs. In the Basic BlockAck frame, the STA acknowledges only the MPDUs starting from the starting sequence control until the MPDU with the highest sequence number that has been received, and the STA shall set bits in the Block Ack bitmap corresponding to all other MPDUs to 0. The status of MPDUs that are considered "old" and prior to the sequence number range for which the receiver maintains status shall be reported as successfully received (i.e., the corresponding bit in the bitmap shall be set to 1). If the Basic BlockAck frame indicates that an MPDU was not received correctly, the originator shall retry that MPDU subject to that MPDU's appropriate lifetime limit.

A typical Block Ack frame exchange sequence using the immediate Block Ack for a single TID is shown in Figure 9-26.

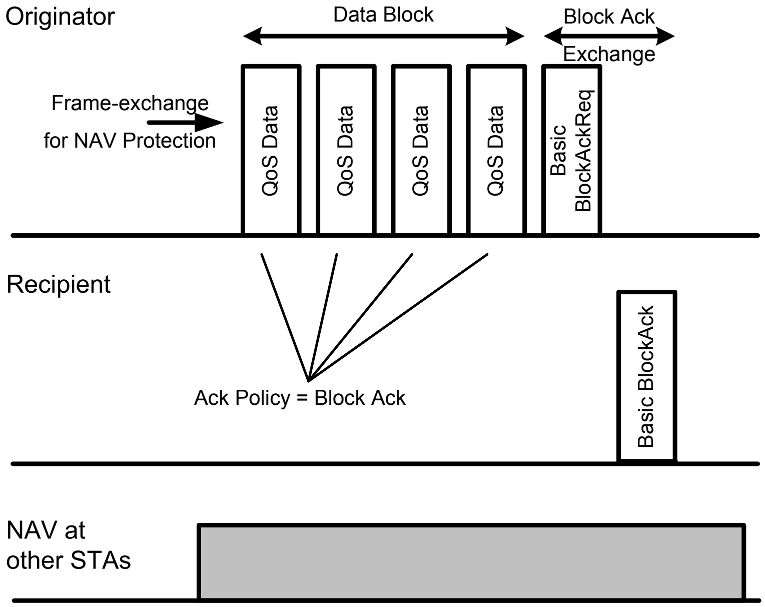


Figure 9-26—A typical Block Ack sequence when immediate policy is used

A typical Block Ack sequence using the delayed Block Ack is shown in Figure 9-27.

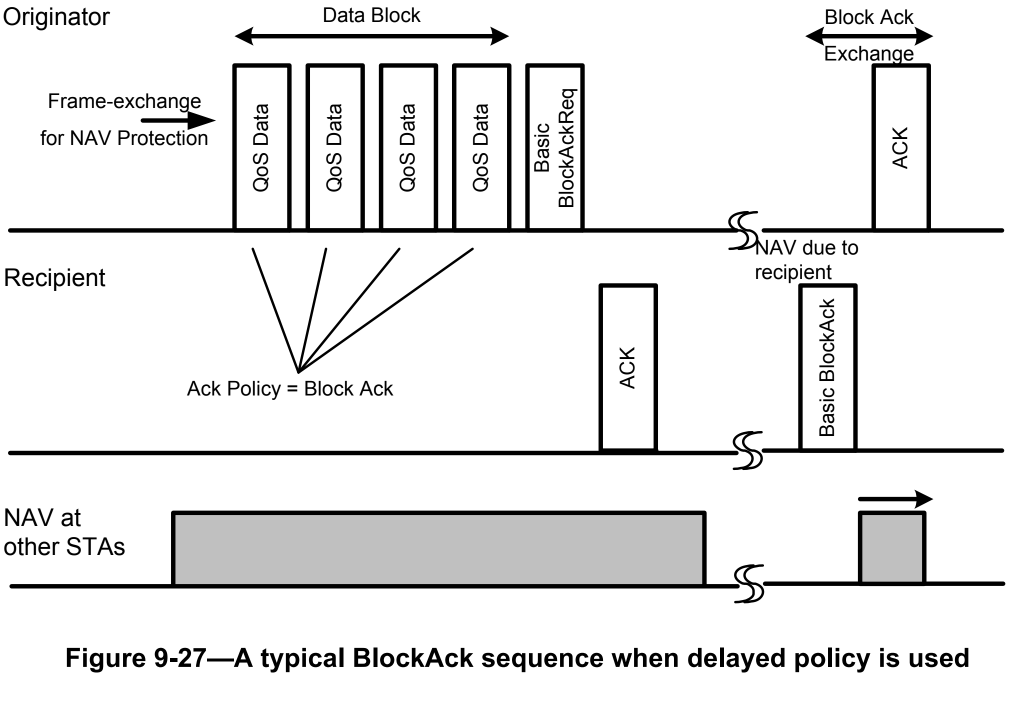


Figure 9-27—A typical BlockAck sequence when delayed policy is used

The subsequent Basic BlockAckReq frame's starting sequence number shall be higher than or equal to the starting sequence number of the immediately preceding Basic BlockAckReq frame for the same TID.

The originator may continue to transmit MPDUs (subject to the negotiated buffer size constraint) to the recipient after transmitting the Basic BlockAckReq frame, but before receiving the Basic BlockAck frame (applicable only to delayed Block Ack). The bitmap in the Basic BlockAck frame shall include the status of frames received between the start sequence number and the transmission of the Basic BlockAckReq frame. A recipient sending a delayed Basic BlockAck frame may update the bitmap with information on QoS data frames received between the receipt of the Basic BlockAckReq frame and the transmission of the Basic BlockAck frame.

If there is no response (i.e., neither a Basic BlockAck nor an ACK frame) to the Basic BlockAckReq frame, the originator may retransmit the Basic BlockAckReq frame within the current TXOP (if time permits) or within a subsequent TXOP. MSDUs that are sent using the Block Ack mechanism are not subject to retry limits but only to MSDU lifetime. The originator need not set the retry bit to 1 for any possible retransmissions of the MPDUs.

The Basic BlockAckReq frame shall be discarded if all MSDUs referenced by this frame have been discarded from the transmit buffer due to expiry of their lifetime limit.

In order to improve efficiency, originators using the Block Ack facility may send MPDU frames with the Ack Policy subfield of the QoS control field set to Normal Ack if only a few MPDUs are available for transmission. The Block Ack record shall be updated irrespective of the Ack Policy subfield in the QoS data frame for the TID with an active Block Ack. When there are sufficient number of MPDUs, the originator may switch back to the use of Block Ack. The reception of QoS data frames using Normal Ack policy shall not be used by the recipient to reset the timer to detect Block Ack timeout (see 10.5.4). This allows the recipient to delete the Block Ack if the originator does not switch back to using Block Ack.

The frame exchange sequences are provided in Annex G.

9.21.4 Receive buffer operation

For each Block Ack agreement, the recipient maintains a MAC variable NextExpectedSequenceNumber. The NextExpectedSequenceNumber is initialized to 0 when a Block Ack agreement is accepted.

Upon the receipt of a QoS data frame from the originator for which a Block Ack agreement exists, the recipient buffers the MSDU regardless of the value of the Ack Policy subfield within the QoS Control field of the QoS data frame, unless the sequence number of the frame is older than the NextExpectedSequenceNumber for that Block Ack agreement, in which case the frame is discarded because it is either old or a duplicate.

The recipient flushes received MSDUs from its receive buffer as described in this subclause.

If a BlockAckReq frame is received, all complete MSDUs and A-MSDUs with lower sequence numbers than the starting sequence number contained in the BlockAckReq frame shall be passed up to the next MAC process as shown in Figure 5-1. Upon arrival of a BlockAckReq frame, the recipient shall pass up the MSDUs and A-MSDUs starting with the starting sequence number sequentially until there is an incomplete or missing MSDU or A-MSDU in the buffer. If no MSDUs or A-MSDUs are passed up to the next MAC process after the receipt of the BlockAckReq frame and the starting sequence number of the BlockAckReq frame is newer than the NextExpectedSequenceNumber for that Block Ack agreement, then the NextExpectedSequenceNumber for that Block Ack agreement is set to the sequence number of the BlockAckReq frame.

If, after an MPDU is received, the receive buffer is full, the complete MSDU or A-MSDU with the earliest sequence number shall be passed up to the next MAC process.

If, after an MPDU is received, the receive buffer is not full, but the sequence number of the complete MSDU or A-MSDU in the buffer with the lowest sequence number is equal to the NextExpectedSequenceNumber for that Block Ack agreement, then the MPDU shall be passed up to the next MAC process.

Each time that the recipient passes an MSDU or A-MSDU for a Block Ack agreement up to the next MAC process, the NextExpectedSequenceNumber for that Block Ack agreement is set to the sequence number of the MSDU or A-MSDU that was passed up to the next MAC process plus one.

The recipient shall pass MSDUs and A-MSDUs up to the next MAC process in order of increasing sequence number.

9.21.5 Teardown of the Block Ack mechanism

When the originator has no data to send and the final Block Ack exchange has completed, it shall signal the end of its use of the Block Ack mechanism by sending the DELBA frame to its recipient. There is no management response frame from the recipient.[39] The recipient of the DELBA frame shall release all resources allocated for the Block Ack transfer.

The Block Ack agreement may be torn down if there are no BlockAck, BlockAckReq, or QoS data frames (sent under Block Ack policy) for the Block Ack's TID received from the peer within a duration of Block Ack timeout value (see 10.5.4).

9.21.6 Selection of BlockAck and BlockAckReq variants

The Compressed Bitmap subfield of the BA Control field or BAR Control field shall be set to 1 in all BlockAck and BlockAckReq frames sent from one HT STA to another HT STA and shall be set to 0 otherwise.

The Multi-TID subfield of the BA Control field shall be set to 1 in all BlockAck frames related to an HT-immediate agreement transmitted inside a PSMP sequence and shall be set to 0 otherwise. The Multi-TID subfield of the BAR Control field shall be set to 1 in all BlockAckReq frames related to an HT-immediate agreement transmitted inside a PSMP sequence and shall be set to 0 otherwise.

Where the terms BlockAck and BlockAckReq are used within 9.21.7 and 9.21.8, the appropriate variant according to this subclause (e.g., Compressed, Multi-TID) is referenced by the generic term.

9.21.7 HT-immediate Block Ack extensions

9.21.7.1 Introduction to HT-immediate Block Ack extensions

An HT extension to the Block Ack feature (defined in 9.21.1 through 9.21.5), called *HT-immediate Block Ack*, is defined in 9.21.7.2 through 9.21.7.9.

The HT-immediate extensions simplify immediate Block Ack use with A-MPDUs and reduce recipient resource requirements.

An HT STA shall support HT-immediate Block Ack in the role of recipient.

[39] Normal Ack rules apply.

9.21.7.2 HT-immediate Block Ack architecture

The HT-immediate Block Ack rules are explained in terms of the architecture shown in Figure 9-28 and explained in this subclause.

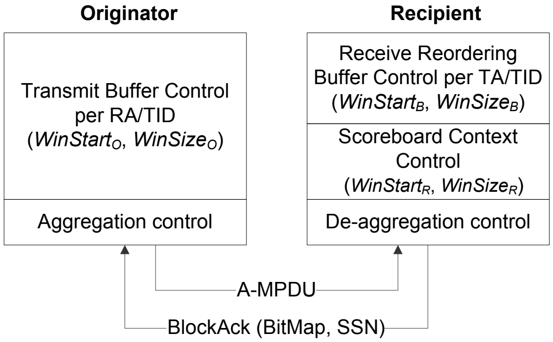

Originator
Recipient
Transmit Buffer Control per RA/TID ($WinStart_O$, $WinSize_O$)
Receive Reordering Buffer Control per TA/TID ($WinStart_B$, $WinSize_B$)
Scoreboard Context Control ($WinStart_R$, $WinSize_R$)
Aggregation control
De-aggregation control
A-MPDU
BlockAck (BitMap, SSN)

Figure 9-28—HT-immediate Block Ack architecture

The originator contains a transmit buffer control that uses $WinStart_O$ and $WinSize_O$ to submit MPDUs for transmission and releases transmit buffers upon receiving BlockAck frames from the recipient.

$WinStart_O$ is the starting sequence number of the transmit window, and $WinSize_O$ is the number of buffers negotiated in the Block Ack agreement.

The Aggregation control creates A-MPDUs. It may adjust the Ack Policy field of transmitted QoS data frames according to the rules defined in 9.21.7.7 in order to solicit BlockAck responses.

The recipient contains a receive reordering buffer control per TA/TID, which contains a related control state. The receive reordering buffer is responsible for reordering MSDUs or A-MSDUs so that MSDUs or A-MSDUs are eventually passed up to the next MAC process in order of received sequence number. It is also responsible for identifying and discarding duplicate frames (i.e., frames that have the same sequence number as a currently buffered frame) that are part of this Block Ack agreement. It maintains its own state independent of the scoreboard context control to perform this reordering as specified in 9.21.7.6.

For each HT-immediate Block Ack agreement, the recipient chooses either full-state or partial-state operation (this choice is known only to the recipient). A STA may simultaneously use full-state operation for some agreements and partial-state operation for other agreements. The scoreboard context control stores an acknowledgment bitmap containing the current reception status of MSDUs or A-MSDUs for HT-immediate Block Ack agreements. Under full-state operation, status is maintained in statically assigned memory. Under partial-state operation, status is maintained in a cache memory; therefore, the status information is subject to cache replacement. This entity provides the bitmap and the value for the Starting Sequence Number subfield to be sent in BlockAck responses to the originator.

The deaggregation control entity separates frames contained in an A-MPDU.

Each received MPDU is analyzed by the scoreboard context control as well as by the receive reordering buffer control.

Each HT-immediate Block Ack agreement is uniquely identified by a tuple of Address 1, Address 2, and TID from the ADDBA Response frame that successfully established the HT-immediate Block Ack agreement. The STA that corresponds to Address 1 of the ADDBA Response frame is the originator. The STA that corresponds to Address 2 of the ADDBA Response frame is the recipient. Data MPDUs that contain the same values for Address 1, Address 2, and TID as a successful ADDBA Response frame are related with the HT-immediate

Block Ack agreement that was established by the successful receipt of that ADDBA Response frame provided that the HT-immediate Block Ack agreement is still active.

9.21.7.3 Scoreboard context control during full-state operation

For each HT-immediate Block Ack agreement that uses full-state operation, a recipient shall maintain a block acknowledgment record as defined in 9.21.3. This record includes a bitmap, indexed by sequence number; a 12-bit unsigned integer starting sequence number, $WinStart_R$, representing the lowest sequence number position in the bitmap; a variable $WinEnd_R$; and the maximum transmission window size, $WinSize_R$, which is set to the smaller of 64 and the value of the Buffer Size field of the associated ADDBA Response frame that established the Block Ack agreement. $WinEnd_R$ is defined as the highest sequence number in the current transmission window. A STA implementing full-state operation for an HT-immediate Block Ack agreement shall maintain the block acknowledgment record for that agreement according to the following rules:

a) At HT-immediate Block Ack agreement establishment:
   1) $WinStart_R = SSN$ from the ADDBA Request frame that elicited the ADDBA Response frame that established the HT-immediate Block Ack agreement.
   2) $WinEnd_R = WinStart_R + WinSize_R – 1$.

b) For each received data MPDU that is related with a specific full-state operation HT-immediate Block Ack agreement, the block acknowledgment record for that agreement is modified as follows, where *SN* is the value of the Sequence Number subfield of the received data MPDU:
   1) If $WinStart_R \leq SN \leq WinEnd_R$, set to 1 the bit in position *SN* within the bitmap.
   2) If $WinEnd_R < SN < WinStart_R+2^{11}$,
      i) Set to 0 the bits corresponding to MPDUs with Sequence Number subfield values from $WinEnd_R$+1 to $SN – 1$.
      ii) Set $WinStart_R = SN – WinSize_R + 1$.
      iii) Set $WinEnd_R = SN$.
      iv) Set to 1 the bit at position *SN* in the bitmap.
   3) If $WinStart_R+2^{11} \leq SN < WinStart_R$, make no changes to the record.

   NOTE—A later-arriving data MPDU might validly contain a sequence number that is lower than an earlier-arriving one. This might happen because the transmitter may choose to send data MPDUs in a nonsequential sequence number order or because a previous data MPDU transmission with lower sequence number is not successful and is being retransmitted.

c) For each received BlockAckReq frame that is related with a specific full-state operation HT-immediate non-Protected Block Ack agreement, the block acknowledgment record for that agreement is modified as follows, where *SSN* is the value from the Starting Sequence Number subfield of the received BlockAckReq frame:
   1) If $WinStart_R < SSN \leq WinEnd_R$,
      i) Set $WinStart_R = SSN$.
      ii) Set to 0 the bits corresponding to MPDUs with Sequence Number subfield values from $WinEnd_R + 1$ through $WinStart_R + WinSize_R – 1$.
      iii) Set $WinEnd_R = WinStart_R + WinSize_R – 1$.
   2) If $WinEnd_R < SSN < WinStart_R+2^{11}$,
      i) Set $WinStart_R = SSN$.
      ii) Set $WinEnd_R = WinStart_R + WinSize_R – 1$.
      iii) Set to 0 bits the corresponding to MPDU with Sequence Number subfield values from $WinStart_R$ to $WinEnd_R$.

3) If $WinStart_R + 2^{11} \leq SSN \leq WinStart_R$, make no changes to the record.

9.21.7.4 Scoreboard context control during partial-state operation

For an HT-immediate Block Ack agreement that uses partial-state operation, a recipient shall maintain a temporary block acknowledgment record as defined in 9.21.3. This temporary record includes a bitmap, indexed by sequence number; a 12-bit unsigned integer $WinStart_R$ (the lowest sequence number represented in the bitmap); a 12-bit unsigned integer $WinEnd_R$ (the highest sequence number in the bitmap); the originator address; TID; and the maximum transmission window size, $WinSize_R$, which is set to the smaller of 64 and the value of the Buffer Size field of the associated ADDBA Response frame that established the Block Ack agreement.

During partial-state operation of scoreboard context control, the recipient retains the current record for an HT-immediate Block Ack agreement at least as long as it receives data from the same originator. If a frame for an HT-immediate Block Ack agreement from a different originator is received, the temporary record may be discarded if the resources it uses are needed to store the temporary record corresponding to the newly arriving frame.

A STA implementing partial-state operation for an HT-immediate Block Ack agreement shall maintain the temporary block acknowledgment record for that agreement according to the following rules:

a) During partial-state operation, $WinStart_R$ is determined by the Sequence Number subfield value of received data MPDUs and by the Starting Sequence Number subfield value of received BlockAckReq frames as described below.

b) For each received data MPDU that is related with a specific partial-state operation HT-immediate Block Ack agreement, when no temporary record for the agreement related with the received data MPDU exists at the time of receipt of the data MPDU, a temporary block acknowledgment record is created as follows, where *SN* is the value of the Sequence Number subfield of the received data MPDU:

   1) $WinEnd_R = SN$.
   2) $WinStart_R = WinEnd_R - WinSize_R + 1$.
   3) Create a bitmap of size $WinSize_R$, with the first bit corresponding to sequence number $WinStart_R$ and the last bit corresponding to sequence number $WinEnd_R$, and set all bits in the bitmap to 0.
   4) Set to 1 the bit in the position in the bitmap that corresponds to *SN*.

c) For each received data MPDU that is related with a specific partial-state operation HT-immediate Block Ack agreement, when a temporary record for the agreement related with the received data MPDU exists at the time of receipt of the data MPDU, the temporary block acknowledgment record for that agreement is modified in the same manner as the acknowledgment record for a full-state agreement described in 9.21.7.3.

d) For each received BlockAckReq frame that is related with a specific partial-state operation HT-immediate non-Protected Block Ack agreement, when no temporary record for the agreement related with the received frame exists at the time of receipt of the frame, a temporary block acknowledgment record is created as follows, where *SSN* is the starting value of the Sequence Number subfield of the received BlockAckReq frame:

   1) $WinStart_R = SSN$.
   2) $WinEnd_R = WinStart_R + WinSize_R – 1$.
   3) Create a bitmap of size $WinSize_R$, and set all bits in the bitmap to 0.

e) For each received BlockAckReq frame that is related with a specific partial-state operation HT-immediate non-Protected Block Ack agreement, when a temporary record for the agreement related with the received frame exists at the time of receipt of the frame, the temporary block acknowledgment record for that agreement is modified in the same manner as the acknowledgment record for a full-state agreement described in 9.21.7.3.

9.21.7.5 Generation and transmission of BlockAck by an HT STA

Except when operating within a PSMP exchange, a STA that receives a PPDU that contains a BlockAckReq in which the Address 1 field matches its MAC address during either full-state operation or partial-state operation shall transmit a PPDU containing a BlockAck frame that is separated on the air by a SIFS interval from the PPDU that elicited the BlockAck as a response. A STA that receives an A-MPDU that contains one or more MPDUs in which the Address 1 field matches its MAC address with the ACK Policy field equal to Normal Ack (i.e., implicit Block Ack request) during either full-state operation or partial-state operation shall transmit a PPDU containing a BlockAck frame that is separated on the air by a SIFS interval from the PPDU that elicited the BlockAck as a response.

When responding with a BlockAck frame to either a received BlockAckReq frame or a received A-MPDU with ACK Policy equal to Normal Ack (i.e., implicit Block Ack request) during either full-state operation or partial-state operation, any adjustment to the value of $WinStart_R$ according to the procedures defined within 9.21.7.3 and 9.21.7.4 shall be performed before the generation and transmission of the response BlockAck frame. The Starting Sequence Number subfield of the Block Ack Starting Sequence Control subfield of the BlockAck frame shall be set to any value in the range from ($WinEnd_R$ – 63) to $WinStart_R$. The values in the recipient's record of status of MPDUs beginning with the MPDU for which the Sequence Number subfield value is equal to $WinStart_R$ and ending with the MPDU for which the Sequence Number subfield value is equal to $WinEnd_R$ shall be included in the bitmap of the BlockAck frame.

When responding with a BlockAck frame to either a received BlockAckReq frame or a received A-MPDU with ACK Policy equal to Normal Ack (i.e., implicit Block Ack request) during either full-state or partial-state operation, if the adjusted value of $WinStart_R$ is greater than the value of the starting sequence number of the BlockAck frame, within the bitmap of the BlockAck frame, the status of MPDUs with sequence numbers that are less than the adjusted value of $WinStart_R$ may be set to any value.

When responding with a BlockAck frame to either a received BlockAckReq frame or a received A-MPDU with ACK Policy equal to Normal Ack (i.e., implicit Block Ack request) during either full-state or partial-state operation, if the adjusted value of $WinEnd_R$ is less than the value of the starting sequence number of the BlockAck frame plus 63, within the bitmap of the BlockAck frame, the status of MPDUs with sequence numbers that are greater than the adjusted value of $WinEnd_R$ shall be reported as unsuccessfully received (i.e., the corresponding bit in the bitmap shall be set to 0).

If a BlockAckReq is received and no matching partial state is available, the recipient shall send a null BlockAck in which the bitmap is set to all zeros.

9.21.7.6 Receive reordering buffer control operation

9.21.7.6.1 General

The behavior described in this subclause, 9.21.7.6.2, and 9.21.7.6.3 applies to a STA that uses either partial-state operation or full-state operation for an HT-immediate Block Ack agreement.

A receive reordering buffer shall be maintained for each HT-immediate Block Ack agreement. Each receive reordering buffer includes a record comprising the following:

- Buffered MSDUs or A-MSDUs that have been received, but not yet passed up to the next MAC process
- A $WinStart_B$ parameter, indicating the value of the Sequence Number subfield of the first (in order of ascending sequence number) MSDU or A-MSDU that has not yet been received

- A $WinEnd_B$ parameter, indicating the highest sequence number expected to be received in the current reception window
- A $WinSize_B$ parameter, indicating the size of the reception window

$WinStart_B$ is initialized to the Starting Sequence Number subfield value of the ADDBA Request frame that elicited the ADDBA Response frame that established the HT-immediate Block Ack agreement.

$WinEnd_B$ is initialized to $WinStart_B + WinSize_B - 1$, where $WinSize_B$ is set to the smaller of 64 and the value of the Buffer Size field of the ADDBA Response frame that established the Block Ack agreement.

Any MSDU or A-MSDU that has been passed up to the next MAC process shall be deleted from the receive reordering buffer.

The recipient shall always pass MSDUs or A-MSDUs up to the next MAC process in order of increasing Sequence Number subfield value.

9.21.7.6.2 Operation for each received data MPDU

For each received data MPDU that is related to a specific HT-immediate Block Ack agreement, the receive reordering buffer record shall be modified as follows, where SN is the value of the Sequence Number subfield of the received MPDU:

a) If $WinStart_B \leq SN \leq WinEnd_B$,
   1) Store the received MPDU in the buffer, if no MSDU with the same sequence number is already present; otherwise discard the MPDU.
   2) Pass MSDUs or A-MSDUs up to the next MAC process that are stored in the buffer in order of increasing value of the Sequence Number subfield starting with the MSDU or A-MSDU that has SN=$WinStart_B$ and proceeding sequentially until there is no buffered MSDU or A-MSDU for the next sequential value of the Sequence Number subfield.
   3) Set $WinStart_B$ to the value of the Sequence Number subfield of the last MSDU or A-MSDU that was passed up to the next MAC process plus one.
   4) Set $WinEnd_B = WinStart_B + WinSize_B - 1$.

b) If $WinEnd_B < SN < WinStart_B + 2^{11}$,
   1) Store the received MPDU in the buffer, if no MSDU with the same sequence number is already present; otherwise discard the MPDU.
   2) Set $WinEnd_B = SN$.
   3) Set $WinStart_B = WinEnd_B - WinSize_B + 1$.
   4) Pass any complete MSDUs or A-MSDUs stored in the buffer with Sequence Number subfield values that are lower than the new value of $WinStart_B$ up to the next MAC process in order of increasing Sequence Number subfield value. Gaps may exist in the Sequence Number subfield values of the MSDUs or A-MSDUs that are passed up to the next MAC process.
   5) Pass MSDUs or A-MSDUs stored in the buffer up to the next MAC process in order of increasing value of the Sequence Number subfield starting with $WinStart_B$ and proceeding sequentially until there is no buffered MSDU or A-MSDU for the next sequential Sequence Number subfield value.
   6) Set $WinStart_B$ to the Sequence Number subfield value of the last MSDU or A-MSDU that was passed up to the next MAC process plus one.
   7) Set $WinEnd_B = WinStart_B + WinSize_B - 1$.

c) If $WinStart_B + 2^{11} \leq SN < WinStart_B$, discard the MPDU (do not store the MPDU in the buffer, and do not pass the MSDU or A-MSDU up to the next MAC process).

9.21.7.6.3 Operation for each received BlockAckReq

For each received BlockAckReq frame that is related with a specific HT-immediate Block Ack agreement, the receive reordering buffer record is modified as follows, where *SSN* is the Starting Sequence Number subfield value of the received BlockAckReq frame:

a) If $WinStart_B < SSN < WinStart_B + 2^{11}$,
   1) For a non-Protected Block Ack agreement, set $WinStart_B = SSN$. See 9.21.9 for a Protected Block Ack agreement.
   2) Set $WinEnd_B = WinStart_B + WinSize_B - 1$.
   3) Pass any complete MSDUs or A-MSDUs stored in the buffer with Sequence Number subfield values that are lower than the new value of $WinStart_B$ up to the next MAC process in order of increasing Sequence Number subfield value. Gaps may exist in the Sequence Number subfield values of the MSDUs or A-MSDUs that are passed up to the next MAC process.
   4) Pass MSDUs or A-MSDUs stored in the buffer up to the next MAC process in order of increasing Sequence Number subfield value starting with $SN = WinStart_B$ and proceeding sequentially until there is no buffered MSDU or A-MSDU for the next sequential Sequence Number subfield value.
   5) Set $WinStart_B$ to the Sequence Number subfield value of the last MSDU or A-MSDU that was passed up to the next MAC process plus one.
   6) Set $WinEnd_B = WinStart_B + WinSize_B - 1$.

b) If $WinStart_B + 2^{11} \le SSN < WinStart_B$, do not make any changes to the receive reordering buffer record.

9.21.7.7 Originator's behavior

A STA may send a block of data in a single A-MPDU where each data MPDU has its Ack Policy field set to Normal Ack. The originator expects to receive a BlockAck response immediately following the A-MPDU if at least one data frame is received without error.

Alternatively, the originator may send an A-MPDU where each data MPDU has its Ack Policy field set to Block Ack under an HT-immediate Block Ack agreement if it does not require a BlockAck response immediately following the A-MPDU.

If the BlockAck is lost, the originator may transmit a BlockAckReq to solicit an immediate BlockAck or it may retransmit the data frames.

A BlockAckReq sent using HT-delayed operation may be transmitted within an A-MPDU provided that its BAR Ack Policy subfield is set to No Acknowledgment.

The originator may transmit QoS data MPDUs with a TID matching an established Block Ack agreement in any order provided that their sequence numbers lie within the current transmission window. The originator may transmit an MPDU with a sequence number that is beyond the current transmission window ($SN > WinStart_O + WinSize_O - 1$), in which case the originator's transmission window (and the recipient's window) is moved forward. The originator should not transmit MPDUs that are lower than (i.e., $SN < WinStart_O$) the current transmission window.

The originator shall not retransmit an MPDU after that MPDU's appropriate lifetime limit.

The originator may send a BlockAckReq for non-Protected Block Ack agreement or a robust management ADDBA frame for Protected Block Ack agreement when a data MPDU that was previously transmitted within an A-MPDU that had the Ack Policy field equal to Normal Ack is discarded due to exhausted MSDU lifetime. The purpose of this BlockAckReq is to shift the recipient's $WinStart_B$ value past the hole in the sequence number space that is created by the discarded data MPDU and thereby to allow the earliest possible passing of buffered frames up to the next MAC process.

9.21.7.8 Maintaining BlockAck state at the originator

If an originator successfully receives a BlockAck in response to a BlockAckReq, the originator shall maintain BlockAck state as defined in 9.21.3.

If the originator receives a BlockAck in response to HT-immediate BlockAckReq, it shall, in addition,

— Not update the status of MPDUs with Sequence Number subfield values between $WinStart_O$ and *SSN* of the received BlockAck, and

  NOTE—It is possible for the Starting Sequence Number subfield value (*SSN*) of the received BlockAck to be greater than $WinStart_O$ because of the failed reception of a nonzero number of MPDUs beginning with the MPDU with Sequence Number subfield value equal to $WinStart_O$ at a recipient that is using partial-state operation.

— Not update the status of MPDUs that have been already positively acknowledged.

  NOTE—This second rule means that if an originator successfully delivered an MPDU and received the BlockAck in one TXOP and then receives a BlockAck in a later TXOP in which the MPDU is not indicated as successfully received (because the partial state has been reset), the originator knows not to retry the MPDU.

9.21.7.9 Originator's support of recipient's partial state

A recipient may choose to employ either full-state operation or partial-state operation for each individual Block Ack agreement. An originator is unaware of the recipient's choice of full-state or partial-state operation.

NOTE—The originator might solicit a BlockAck as the last activity associated with that Block Ack agreement in the current TXOP to reduce the probability that data are unnecessarily retransmitted due to loss of partial state.

9.21.8 HT-delayed Block Ack extensions

9.21.8.1 Introduction

Subclauses 9.21.8.2 and 9.21.8.3 define an HT extension to the Block Ack feature to support operation on delayed Block Ack agreements established between HT STAs. Other than the exceptions noted in 9.21.8.1 through 9.21.8.3, the operation of HT Delayed Block Ack is the same as is described in 9.21.7.

The HT-delayed extensions simplify the use of delayed Block Ack in an A-MPDU and reduce resource requirements.

9.21.8.2 HT-delayed Block Ack negotiation

HT-delayed Block Ack is an optional feature. An HT STA declares support for HT-delayed Block Ack in the HT Capabilities element.

An HT STA shall not attempt to create a BlockAck agreement under HT-delayed Block Ack Policy unless the recipient HT STA declares support for this feature.

9.21.8.3 Operation of HT-delayed Block Ack

The BlockAck response to an HT-delayed BlockAckReq is transmitted after an unspecified delay and when the recipient of the BlockAckReq next has the opportunity to transmit. This response may be transmitted in a later TXOP owned by the recipient of the BlockAckReq or in the current or a later TXOP owned by the sender of the BlockAckReq using the RD feature (see 9.25).

The No Ack feature of the BlockAckReq and BlockAck frame may be used under an HT-delayed Block Ack agreement.

A BlockAckReq or BlockAck frame containing a BAR Ack Policy or BA Ack Policy subfield equal to 1 indicates that no acknowledgment is expected to these control frames. Otherwise, an Ack MPDU response is expected after a SIFS.

Setting of the BAR Ack Policy and BA Ack Policy subfields may be performed independently for BlockAckReq and BlockAck frames associated with the same HT-delayed Block Ack agreement. All four combinations of the values of these fields are valid.

Setting of the BAR Ack Policy and BA Ack Policy subfields is dynamic and may change from PPDU to PPDU.

9.21.9 Protected Block Ack Agreement

A STA indicates support for Protected Block Ack by setting the MFPC, MFPR, and PBAC RSN Capabilities subfields to 1. Such a STA is a PBAC STA; otherwise, the STA is a non-PBAC STA. A Block Ack agreement that is successfully negotiated between two PBAC STAs is a Protected Block Ack agreement. A Block Ack agreement that is successfully negotiated between two STAs when either or both of the STAs is not a PBAC STA is a non-Protected Block Ack agreement.

A PBAC STA may choose to negotiate a Block Ack agreement with a non-PBAC STA if dot11RSNAPBACRequired is 0; otherwise, it shall negotiate a Block Ack agreement only with other PBAC STAs. If a PBAC STA is communicating with a non-PBAC STA, it shall follow the rules for an non-Protected Block Ack agreement.

A STA that has successfully negotiated a Protected Block Ack agreement shall obey the following rule as a Block Ack originator in addition to rules specified in 9.21.7.7 and 9.21.7.8:

— To change the value of $WinStart_B$ at the receiver, the STA shall use a robust management ADDBA Request frame.

A STA that has successfully negotiated a Protected Block Ack agreement shall obey the following rules as a Block Ack recipient in addition to rules specified in 9.21.7.3 to 9.21.7.6:

— The recipient STA shall respond to a BlockAckReq from a PBAC enabled originator with an immediate BlockAck. The Block Ack Starting Sequence Control subfield value shall be ignored for the purposes of updating the value of $WinStart_B$. The Block Ack Starting Sequence Control subfield value may be utilized for the purposes of updating the value of $WinStart_R$. If the Block Ack Starting Sequence Control subfield value is greater than $WinEnd_B$ or less than $WinStart_B$, dot11PBACErrors shall be incremented by 1.

— Upon receipt of a valid robust management ADDBA Request frame for an established Protected Block Ack agreement whose TID and transmitter address are the same as those of the Block Ack agreement, the STA shall update its $WinStart_R$ and $WinStart_B$ values based on the starting sequence number in the robust management ADDBA Request frame according to the procedures outlined for reception of BlockAckReq frames in 9.21.7.3, 9.21.7.4, 9.21.7.6.1, and 9.21.7.6.3, while treating the starting sequence number as though it were the *SSN* of a received BlockAckReq frame. Values in other fields of the ADDBA frame shall be ignored.

9.22 No Acknowledgment (No Ack)

The usage of No Ack is determined by the policy at the QoS STA. When No Ack policy is used, there is no MAC-level recovery, and the reliability of this traffic is reduced, relative to the reliability of traffic with other acknowledgment policies, due to the increased probability of lost frames from interference, collisions, or time-varying channel parameters. A protective mechanism (such as transmitting using HCCA, RTS/CTS, or the mechanism described in 9.23) should be used to reduce the probability of other STAs transmitting during the TXOP.

9.23 Protection mechanisms

9.23.1 Introduction

These protection mechanisms cause a STA that is a potential interferer to defer any transmission for a known period of time. When these mechanisms are used, non-ERP STAs do not interfere with frame exchanges using ERP PPDUs between ERP STAs and non-HT STAs do not interfere with frame exchanges using HT PPDUs between HT STAs. As a result, non-ERP and/or non-HT STAs are allowed to coexist with ERP and/or HT STAs.

9.23.2 Protection mechanism for non-ERP receivers

The intent of a protection mechanism is to cause a STA to not transmit an MPDU of type Data or an MMPDU with an ERP-OFDM preamble and header unless it has attempted to update the NAV of receiving NonERP STAs. The updated NAV period shall be longer than or equal to the total time required to send the data and any required response frames. ERP STAs shall use protection mechanisms (such as RTS/CTS or CTS-to-self) for ERP-OFDM MPDUs of type Data or an MMPDU when the Use_Protection field of the ERP element is equal to 1 (see the requirements of 9.7). Protection mechanisms frames shall be sent using one of the mandatory Clause 16 or Clause 17 rates and using one of the mandatory Clause 16 or Clause 17 waveforms, so all STAs in the BSA are able to learn the duration of the exchange even if they cannot detect the ERP-OFDM signals using their CCA function.

Note that when using the Clause 19 options, ERP-PBCC or DSSS-OFDM, there is no need to use protection mechanisms, as these frames start with a DSSS header.

In the case of a BSS composed of only ERP STAs, but with knowledge of a neighboring co-channel BSS having NonERP traffic, the AP may require protection mechanisms to protect the BSS's traffic from interference. This provides propagation of NAV to all attached STAs and all STAs in a neighboring co-channel BSS within range by messages sent using rates contained in the BSSBasicRateSet parameter. The frames that propagate the NAV throughout the BSS include RTS/CTS/ACK frames, all data frames with the "more fragments" field equal to 1, all data frames sent in response to PS-Poll that are not proceeded in the frame sequence by a data frame with the "more fragments" field equal to 1, Beacon frames with nonzero CFDurRemaining, CF-End frames, and CF-End+ACK frames.

When RTS/CTS is used as the protection mechanism, cases exist such as NAV resetting (discretionary, as indicated in 9.3.2.4), where a hidden STA may reset its NAV and this may cause a collision. The likelihood of occurrence is low, and it is not considered to represent a significant impairment to overall system operation. A mechanism to address this possible situation would be to use alternative protection mechanisms or to revert to alternative modulation methods.

If a protection mechanism is being used, a fragment sequence shall use ERP-OFDM modulation for the final fragment and control response.

The rules for calculating RTS/CTS NAV fields are unchanged when using RTS/CTS as a protection mechanism.

Additionally, if any of the rates in the BSSBasicRateSet parameter of the protection mechanism frame transmitting STA's BSS are Clause 16 or Clause 17 rates, then the protection mechanism frames shall be sent at one of those Clause 16 or Clause 17 basic rates.

The NonERP_Present bit shall be set to 1 when a NonERP STA is associated with the BSS. In an IBSS if a member of an IBSS detects one or more NonERP STAs that are members of the same IBSS the NonERP_Present bit should be set to 1. Examples of when the NonERP present bit may additionally be set to 1 include, but are not limited to, when:

a) A NonERP infrastructure or independent BSS is overlapping (a NonERP BSS may be detected by the reception of a Beacon where the supported rates contain only Clause 16 or Clause 17 rates).
b) In an IBSS if a Beacon frame is received from one of the IBSS participants where the supported rate set contains only Clause 16 or Clause 17 rates.
c) A management frame (excluding a Probe Request) is received where the supported rate set includes only Clause 16 or Clause 17 rates.

A NonERP mesh STA shall set the NonERP_Present and Use_Protection bits to 1, when establishing a mesh peering with a mesh STA.

When a mesh STA establishes a mesh peering with a NonERP mesh STA, the mesh STA shall set the NonERP_Present bit to 1 and the mesh STA should set the Use_Protection bit to 1. In addition, a mesh STA should set the NonERP_Present bit and the Use_Protection bit to 1 when

— A mesh STA detects the overlapped presence of either a NonERP BSS, a NonERP IBSS, or a NonERP MBSS, or
— A Beacon frame is received from a neighbor STA where the supported rate set contains only Clause 16 or Clause 17 rates, or
— A management frame (excluding Probe Request) is received where the supported rate set includes only Clause 16 or Clause 17 rates.

A mesh STA may set the NonERP_Present and the Use_Protection bits to 1 based on its internal policies, which is beyond the scope of the standard.

The Use_Protection bit may be set to 1 when the NonERP_Present bit is 1.

If one or more NonERP STAs are associated in the BSS, the Use_Protection bit shall be set to 1 in transmitted ERP elements.

In an IBSS the setting of the Use_Protection bit is left to the STA. In an IBSS there is no uniform concept of association; therefore, a typical algorithm for setting the Use_Protection bit takes into account the traffic pattern and history on the network. If a member of an IBSS detects one or more NonERP STAs that are members of the same IBSS or receives a Beacon from a member of the same IBSS with the Use_Protection bit equal to 1, then the Use_Protection bit should be set to 1 in the ERP element of transmitted Beacon and Probe Response frames.

ERP APs and ERP STAs shall invoke the use of a protection mechanism after transmission or reception of the Use_Protection bit with a value of 1 in an MMPDU to or from the BSS that the ERP AP or ERP STA has joined or started. ERP APs and ERP STAs may additionally invoke protection mechanism use at other times. ERP APs and ERP STAs may disable protection mechanism use after transmission or reception of the Use_Protection bit with a value of 0 in an MMPDU to or from the BSS that the ERP AP or ERP STA has joined or started.

ERP mesh STAs shall invoke the use of a protection mechanism after the transmission of the Use_Protection bit with a value of 1 in an MMPDU. In addition, ERP mesh STAs may invoke protection mechanism at other times. ERP mesh STAs may disable protection mechanism use after transmission of the Use_Protection bit with a value of 0 in an MMPDU.

When there are no NonERP STAs associated with the BSS and the ERP element sender's dot11ShortPreambleOptionImplemented is true, then the Barker_Preamble_Mode bit may be set to 0. The Barker_Preamble_Mode bit shall be set to 1 by the ERP element sender if one or more associated NonERP STAs are not short preamble capable as indicated in their Capability Information field, or if the ERP element senders dot11ShortPreambleOptionImplemented is false.

When dot11ShortPreambleOptionImplemented is true and all peer mesh STAs support the short preamble, the mesh STA may set the Baker_Preamble_Mode bit to 0. When dot11ShortPreambleOptionImplemented is false or any of its peer mesh STAs do not support the short preamble, the mesh STA shall set the Barker_Preamble_Mode field to 1.

If a member of an IBSS detects one or more nonshort-preamble-capable STAs that are members of the same IBSS, then the Barker_Preamble_Mode bit should be set to 1 in the transmitted ERP element.

ERP APs and ERP STAs shall use long preambles when transmitting Clause 16, Clause 17, and Clause 19 frames after transmission or reception of an ERP element with a Barker_Preamble_Mode value of 1 in an MMPDU to or from the BSS that the ERP AP or ERP STA has joined or started, regardless of the value of the short preamble capability bit from the same received or transmitted MMPDU that contained the ERP element. ERP APs and ERP STAs may additionally use long preambles when transmitting Clause 16, Clause 17, and Clause 19 frames at other times. ERP APs and ERP STAs may use short preambles when transmitting Clause 16, Clause 17, and Clause 19 frames after transmission or reception of an ERP element with a Barker_Preamble_Mode value of 0 in an MMPDU to or from the BSS that the ERP AP or ERP STA has joined or started, regardless of the value of the short preamble capability bit from the same received or transmitted MMPDU. NonERP STAs and NonERP APs may also follow the rules given in this paragraph.

ERP mesh STAs shall use long preambles when transmitting Clause 16, Clause 17, and Clause 19 frames after transmission or reception of an ERP element with a Barker_Preamble_Mode value of 1 in an MMPDU to or from the MBSS to which the ERP mesh STA belongs. Mesh STAs may additionally use long preambles when transmitting Clause 16, Clause 17, and Clause 19 frames at other times.

9.23.3 Protection mechanisms for transmissions of HT PPDUs

9.23.3.1 General

Transmissions of HT PPDUs, referred to as *HT transmissions*, are protected if there are other STAs present that cannot interpret HT transmissions correctly. The HT Protection and Nongreenfield HT STAs Present fields in the HT Operation element within Beacon and Probe Response frames are used to indicate the protection requirements for HT transmissions.

The HT Protection field may be set to no protection mode only if the following are true:

- All STAs detected (by any means) in the primary or the secondary channel are HT STAs, and
- All STAs that are known by the transmitting STA to be a member of this BSS are either
  - 20/40 MHz HT STAs in a 20/40 MHz BSS, or
  - 20 MHz HT STAs in a 20 MHz BSS.

The HT Protection field may be set to nonmember protection mode only if the following are true:

- A non-HT STA is detected (by any means) in either the primary or the secondary channel or in both the primary and secondary channels, that is not known by the transmitting STA to be a member of this BSS, and
- All STAs that are known by the transmitting STA to be a member of this BSS are HT STAs.

The HT Protection field may be set to 20 MHz protection mode only if the following are true:

— All STAs detected (by any means) in the primary channel and all STAs detected (by any means) in the secondary channel are HT STAs and all STAs that are members of this BSS are HT STAs, and
— This BSS is a 20/40 MHz BSS, and
— There is at least one 20 MHz HT STA associated with this BSS.

The HT Protection field is set to non-HT mixed mode otherwise.

NOTE—The rules stated above allow an HT AP to select non-HT mixed mode at any time.

In an IBSS, the HT Protection field is reserved, but an HT STA shall protect HT transmissions as though the HT Protection field were equal to non-HT mixed mode. A STA that is a member of an IBSS shall protect HT-greenfield format PPDUs and RIFS sequences, adhering to the same requirements as described in the line of Table 9-9 labeled "Use_Protection = 0 or ERP element is not present (HT Protection field equal to non-HT mixed mode)."

Table 9-9—Protection requirements for HT Protection field values nonmember protection mode and non-HT mixed mode

Condition	Requirements
Use_Protection = 0 or ERP element is not present (HT Protection field equal to non-HT mixed mode)	The protection requirements for HT transmissions using HT-greenfield format are specified in 9.23.3.1. The protection requirements for HT transmissions using RIFS within the HT transmission burst are specified in 9.23.3.3. The protection mechanism for other transmissions not already described above is based on one of the sequences defined in Table 9-10.
Use_Protection = 1 (HT Protection field equal to nonmember protection mode or non-HT mixed mode)	All HT transmissions shall be protected using mechanisms as described in 9.23.2. The frames that are used for providing the protection shall be sent at a Clause 16 or Clause 17 rate.

In an MBSS, the HT Protection field and the Nongreenfield HT STAs Present field are determined as described in 9.23.3.5.

In an IBSS and an MBSS, the RIFS Mode field of the HT Operation element is reserved, but an HT STA shall operate as though this field were equal to 1.

During the 40 MHz phase of PCO operation, a PCO active STA may act as though the HT Protection field were equal to no protection mode, regardless of the actual value of the HT Protection field transmitted by the AP.

When the HT Protection field is equal to no protection mode or 20 MHz protection mode and the Nongreenfield HT STAs Present field is equal to 0, no protection is required since all HT STAs in the BSS are capable of decoding HT-mixed format and HT-greenfield format transmissions.

When the HT Protection field is equal to no protection mode or 20 MHz protection mode and the Nongreenfield HT STAs Present field is equal to 1, HT transmissions that use the HT-greenfield format shall be protected. This protection may be established by transmitting a PPDU with the TXVECTOR FORMAT parameter set to HT_MF or any of the methods described in Table 9-10.

Table 9-10—Applicable HT protection mechanisms

HT protection mechanism
Control frames such as RTS/CTS or CTS-to-self prior to the HT transmissions: — 20 MHz transmissions use the rates defined in Clause 18 or Clause 19 — 40 MHz transmissions use non-HT duplicate frames defined in Clause 20
Transmit an initial frame within a non-HT PPDU that requires a response frame. The remaining TXOP following the first PPDU exchange may contain PPDUs using HT-greenfield format and/or separated by RIFS.
L-SIG TXOP protection
Using a PPDU with the TXVECTOR FORMAT parameter set to HT_MF, transmit first a PPDU that requires a response that is sent using a non-HT PPDU. The remaining TXOP following the first PPDU exchange may contain HT-greenfield format and/or RIFS sequences.

When the HT Protection field is equal to nonmember protection mode and the Use_Protection field in the ERP element is equal to 0, HT transmissions should be protected. When the HT Protection field is equal to nonmember protection mode, the Use_Protection field in the ERP element is equal to 0, and the Nongreenfield HT STAs Present field is equal to 1, HT transmissions using HT-greenfield format shall be protected. When the HT Protection field is equal to nonmember protection mode and the Use_Protection field in the ERP element is equal to 1, HT transmissions shall be protected according to the requirements described in Table 9-9.

When the HT Protection field is equal to non-HT mixed mode, HT transmissions shall be protected. The type of protection required depends on the type of transmission as well as the type of the non-HT STAs that are present in the BSS. Protection requirements that apply when the HT Protection field is equal to non-HT mixed mode are described in Table 9-9.

If the transmission requires protection and the Use_Protection field within the ERP element is equal to 0 or the ERP element is not present in the Beacon, HT transmissions shall be protected using one of the mechanisms identified in Table 9-10.

NOTE—Rules for rate selection for the HT protection mechanisms listed in Table 9-10 are described in 9.7.

If the HT Protection field is equal to no protection mode and the Secondary Channel Offset field is equal to SCA or SCB, a STA may transmit a 40 MHz HT PPDU (TXVECTOR parameter CH_BANDWIDTH set to HT_CBW40) to initiate a TXOP provided that the restrictions specified in 9.7 are obeyed. When the HT Protection field is not equal to no protection mode or the Secondary Channel Offset field is equal to SCN, a STA shall not transmit a 40 MHz HT PPDU (TXVECTOR parameter CH_BANDWIDTH set to HT_CBW40) to initiate a TXOP.

9.23.3.2 Protection rules for HT STA operating a direct link

An HT STA operating a direct link with another HT STA in a non-HT BSS shall operate according to the rules found in 9.23 as though the following fields have the settings indicated:

a) The RIFS Mode field of the HT Operation element equal to 1
b) The HT Protection field equal to non-HT mixed mode
c) The Nongreenfield HT STAs Present field equal to 1
d) The OBSS Non-HT STAs Present field equal to 1
e) The L-SIG TXOP Full Support field equal to 0
f) The PCO Active field equal to 0

g) The Basic MCS Set field equal to all zeros

9.23.3.3 RIFS protection

If the HT Protection field is equal to nonmember protection mode or non-HT mixed mode, the AP may set the RIFS Mode field to 0 according to implementation-specific criteria (i.e., such as to protect overlapping non-HT BSSs in the primary or secondary channels).

If the HT Protection field is not equal to nonmember protection mode and it is not equal to non-HT mixed mode, the RIFS Mode field shall be set to 1.

If the RIFS Mode field of an AP's HT Operation element is equal to 1:

a) A STA that is associated with the AP may protect RIFS sequences when the HT Protection field of the HT Operation element transmitted by the AP is equal to nonmember protection mode.
b) A STA that is associated with the AP shall protect RIFS sequences when the HT Protection field of the HT Operation element transmitted by the AP is equal to non-HT mixed mode.

A STA shall not transmit PPDUs separated by a RIFS unless the RIFS Mode field of the HT Operation element is equal to 1.

9.23.3.4 Use of OBSS Non-HT STAs Present field

The OBSS Non-HT STAs Present field allows HT APs to report the presence of non-HT STAs that are not members of its BSS in the primary channel, the secondary channel, or in both primary and secondary channels.

A second HT AP that detects a first HT AP's Beacon frame with the OBSS Non-HT STAs Present field equal to 1 may cause HT-greenfield format and RIFS sequence transmissions of the second AP's BSS to be protected by setting the HT Protection field of its HT Operation element to non-HT mixed mode. If the NonERP_Present field is equal to 1 in the first AP's Beacon frame, the Use_Protection field may also be set to 1 by the second AP.

An HT STA may also scan for the presence of non-HT devices either autonomously or, for example, after the STA's AP transmits an HT Operation element with the HT Protection field equal to nonmember protection mode. Non-HT devices can be detected as follows:

— Reception of a management frame that does not carry an HT Capabilities element and the frame is required to carry this element when transmitted by an HT STA, or
— Reception of a Beacon containing an HT Operation element with the OBSS Non-HT STAs Present field equal to 1.

When non-HT devices are detected, the STA may enable protection of its HT-greenfield format and RIFS sequence transmissions.

NOTE—If a non-HT device is detected and the STA determines that its HT-greenfield format or RIFS sequence transmissions are affecting the operation of the non-HT device, then the STA might enable protection of its HT-greenfield format and RIFS sequence transmissions.

See also 10.9.8.5, which defines rules for the OBSS Non-HT STAs Present field related to HT-greenfield transmissions in certain operating classes.

9.23.3.5 Protection rules for an HT mesh STA in an MBSS

A mesh STA determines the HT Protection and Nongreenfield HT STAs Present fields in the HT Operation element in the transmitting frame as follows:

The HT Protection field in a mesh STA may be set to no protection mode only if

- All STAs detected in the primary or the secondary channel are HT STAs, and
- All mesh STA members of this MBSS that are one-hop neighbors of the transmitting mesh STA are either:
  - 20/40 MHz HT mesh STAs in a 20/40 MHz MBSS, or
  - 20 MHz HT mesh STAs in a 20 MHz MBSS.

The HT Protection field in a mesh STA may be set to nonmember protection mode only if

- A non-HT STA is detected in either the primary or the secondary channel or in both the primary and secondary channels, that is not known by the transmitting mesh STA to be a member of this MBSS, and
- All mesh STA members of this MBSS that are one-hop neighbors of the transmitting mesh STA are HT mesh STAs.

The HT Protection field in a mesh STA may be set to 20 MHz protection mode only if

- All STAs detected in the primary and all STAs detected in the secondary channel are HT STAs, and
- All mesh STA members of this MBSS that are one-hop neighbors of the transmitting mesh STA are HT mesh STAs, and
- The MBSS is a 20/40 MHz MBSS, and
- There is at least one 20 MHz HT mesh STA that is one-hop neighbor of the transmitting mesh STA.

The HT Protection field in a mesh STA is set to non-HT mixed mode otherwise.

If two peer HT mesh STAs report the same protection mode in HT Protection field, the protection mechanisms of the related mode shall be used to protect the transmission between the peer HT mesh STAs.

If an HT mesh STA and its peer HT mesh STA report different protection modes in HT Protection field, the following rules shall be used:

a) If an HT mesh STA or its peer HT mesh STA reports non-HT mixed mode, the protection mechanisms of non-HT mixed mode shall be used to protect the transmission between the peer HT mesh STAs.
b) If an HT mesh STA or its peer HT mesh STA reports nonmember protection mode and non-HT mixed mode is not reported by any of these HT mesh STAs, the protection mechanisms of nonmember protection mode shall be used to protect the transmission between the peer HT mesh STAs.
c) If an HT mesh STA or its peer HT mesh STA reports 20 MHz protection mode and neither non-HT mixed mode nor nonmember protection mode is reported by any of these HT mesh STAs, the protection mechanisms of 20 MHz protection mode shall be used to protect the transmission between the peer HT mesh STA.

If at least one HT peer mesh STA in its mesh neighborhood indicates the Nongreenfield HT STAs Present equal to 1, the protection rules related to Nongreenfield HT STAs Present should also be applied to the communication between HT peer mesh STAs.

9.23.4 L_LENGTH and L_DATARATE parameter values for HT-mixed format PPDUs

L_LENGTH and L_DATARATE determine the duration that non-HT STAs do not transmit, equal to the remaining duration of the HT PPDU or the L-SIG duration when L-SIG TXOP protection is used as defined in 9.23.5, following the non-HT portion of the preamble of the HT-mixed format PPDU.

The L_DATARATE parameter of the TXVECTOR shall be set to the value 6 Mb/s.

A STA that is transmitting a PPDU with the FORMAT parameter of the TXVECTOR equal to HT_MF and that is not operating by the L-SIG TXOP protection rules described in 9.23.5 shall set the value of the L_LENGTH parameter to the value (in octets) given by Equation (9-12):

$$\text{L\_LENGTH} = \left\lceil \frac{((\text{TXTIME} - \text{Signal Extension}) - (\text{aPreambleLength} + \text{aPLCPHeaderLength}))}{\text{aSymbolLength}} \right\rceil \times N_{OPS} - \left\lceil \frac{\text{aPLCPServiceLength} + \text{aPLCPConvolutionalTailLength}}{8} \right\rceil \tag{9-12}$$

where

$\lceil x \rceil$ denotes the smallest integer greater than or equal to x

TXTIME is the duration (in microseconds) of the HT PPDU defined in 6.5.7

Signal Extension is 0 μs when TXVECTOR parameter NO_SIG_EXTN is true and is the duration of signal extension as defined by aSignalExtension in Table 20-25 of 20.4.4 when TXVECTOR parameter NO_SIG_EXTN is false

aSymbolLength is the duration of a symbol (in microseconds), defined in 6.5.4

(aPreambleLength + aPLCPHeaderLength) is the duration (in microseconds) of the non-HT PLCP preamble and L-SIG, defined in 6.5.4

N_{OPS} is the number of octets transmitted during a period of aSymbolLength at the rate specified by L_DATARATE

aPLCPServiceLength is the number of bits in the PLCP SERVICE field, defined in 6.5.4 (PLME-CHARACTERISTICS.confirm)

aPLCPConvolutionalTailLength is the number of bits in the convolutional code tail bit sequence, defined in 6.5.4

NOTE 1—The last term of the L_LENGTH definition corrects for the fact that non-HT STAs add the length of the SERVICE field and tail bits (assuming a single convolutional encoder) to the value communicated by the L_LENGTH field.

NOTE 2—For a Clause 20 PHY, this equation simplifies to $\text{L\_LENGTH} = \left\lceil \frac{((\text{TXTIME} - \text{Signal Extension}) - 20)}{4} \right\rceil \times 3 - 3$.

A STA that is operating under L-SIG TXOP protection shall set the L_LENGTH parameter according to rules described in 9.23.5.

A STA shall not transmit a PPDU with the FORMAT parameter set to HT_MF in TXVECTOR if the corresponding L_LENGTH value calculated with Equation (9-12) exceeds 4095 octets.

NOTE—The transmission of frames with L_LENGTH above 2332 octets (i.e., a data MPDU containing an unencrypted 2304 octet MSDU) might be accompanied by a protection mechanism (e.g., RTS/CTS or CTS-to-self protection) if it is determined that the use of L_LENGTH fails to effectively suppress non-HT transmissions. How this is determined is outside the scope of this standard.

9.23.5 L-SIG TXOP protection

9.23.5.1 General rules

Figure 9-29 illustrates the basic concept of L-SIG TXOP protection. The terms used in this figure are defined below and in 20.3.2.

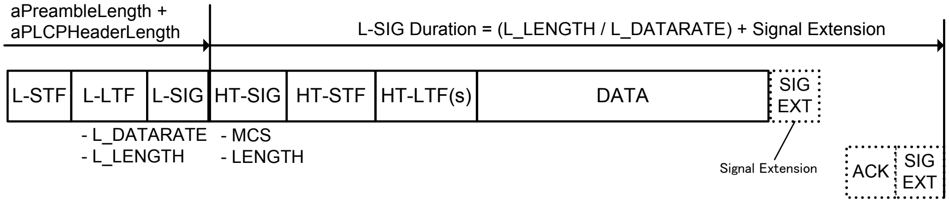


Figure 9-29—Basic concept of L-SIG TXOP protection

The AP determines whether all HT STAs associated with its BSS support L-SIG TXOP protection and indicates this determination in the L-SIG TXOP Protection Full Support field of its HT Operation element. This field shall not be set to 1 unless the L-SIG TXOP Protection field is equal to 1 for all HT STAs in the BSS.

Support for L-SIG TXOP protection at an intended recipient can be determined through examination of its HT Capability element.

In an IBSS and an MBSS, the L-SIG TXOP Protection Full Support field of the HT Operation element is reserved, but HT STAs shall operate as though the field were equal to 0.

A STA shall not transmit a frame using L-SIG TXOP protection directed to a recipient that does not support L-SIG TXOP protection.

A STA that transmits an L-SIG TXOP Protected frame should use an MCS from the BSSBasicMCSSet for the transmission of that frame if

- — The frame initiates a TXOP in an IBSS or in an MBSS, or
- — The L-SIG TXOP Protection Full Support field is equal to 0 by its AP.

Under L-SIG TXOP protection operation, the L-SIG field with an HT-mixed format PHY preamble represents a duration value equivalent (except in the case of the initial frame that establishes the TXOP, as described below) to the sum of:

a) The value of Duration/ID field contained in the MAC header and

b) The duration remaining in the current packet after the L-SIG, which is equal to the duration of the current packet less (aPreambleLength + aPLCPHeaderLength)

A duration value determined from the L_DATARATE and L_LENGTH parameters of the TXVECTOR or RXVECTOR rounded up to a multiple of aSymbolLength that is not equal to the remaining duration of the frame is called an *L-SIG duration*. The TXVECTOR L_LENGTH (defined in 20.2.2), when L-SIG TXOP protection is used, shall contain the value (in octets) given by Equation (9-13).

$$\begin{aligned} \text{L\_LENGTH} = & \left\lceil \frac{\text{L-SIG Duration} - \text{Signal Extension}}{\text{aSymbolLength}} \right\rceil \times N_{OPS} \\ & - \left\lceil \frac{\text{aPLCPServiceLength} + \text{aPLCPConvolutionalTailLength}}{8} \right\rceil \end{aligned} \tag{9-13}$$

where

$\lceil x \rceil$ denotes the lowest integer greater than or equal to x
Signal Extension is defined in 9.23.4
aSymbolLength is the duration of symbol, defined in 6.5.4
N_{OPS} is the number of octets transmitted during a period of aSymbolLength at the rate specified by L_DATARATE
aPLCPServiceLength is the number of bits in the PLCP SERVICE field, defined in 6.5.4
aPLCPConvolutionalTailLength is the number of bits in the convolutional code tail bit sequence, defined in 6.5.4
durations are expressed in microseconds

NOTE—For a Clause 20 PHY, this equation simplifies to $\text{L\_LENGTH} = \left\lceil \frac{(\text{L-SIG Duration} - \text{Signal Extension})}{4} \right\rceil \times 3 - 3$.

Non-HT STAs are not able to receive any PPDUs that start during the L-SIG duration. Therefore, no frame shall be transmitted to a non-HT STA during an L-SIG protected TXOP.

See also 9.3.2.4, which describes a rule for resetting a NAV value that was set by an L-SIG TXOP protected frame.

9.23.5.2 L-SIG TXOP protection rules at the TXOP holder

Figure 9-30 illustrates an example of how L-SIG durations are set when using L-SIG TXOP protection.

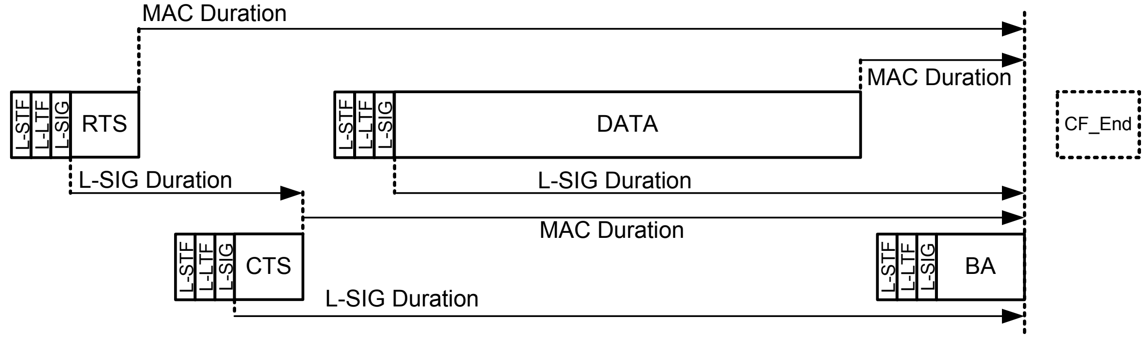


Figure 9-30—Example of L-SIG duration setting

An L-SIG TXOP protected sequence shall start with one of the following:

— an initial handshake, which is the exchange of two frames (each inside an HT-mixed format PPDU) that establish protection (e.g., RTS/CTS) or
— an initial frame that establishes protection but generates no response (e.g., a CTS to self)

provided that this initial sequence is also valid for the start of a TXOP. The term *L-SIG TXOP protected sequence* includes these initial frames and any subsequent frames transmitted within the protected duration.

Under L-SIG TXOP protection operation, when the initial PPDU that establishes protection requires a response, the L-SIG duration of the initial PPDU shall be as follows:

$$\text{L-SIG Duration} = (T_{Init\_PPDU} - (\text{aPreambleLength} + \text{aPLCPHeaderLength})) + \text{SIFS} + T_{Res\_PPDU}$$

where

T_{Init_PPDU} is the length in time (in microseconds) of the entire initial PPDU

T_{Res_PPDU} is the length in time (in microseconds) of the expected response PPDU

$(\text{aPreambleLength} + \text{aPLCPHeaderLength})$ is the length in time (in microseconds) of the non-HT PCLP header, defined in 6.5.4

When the initial PPDU that establishes protection requires no response, the L-SIG duration shall contain the following value:

$$\text{L-SIG Duration} = (T_{Init\_MACDur} + T_{Init\_PPDU} - (\text{aPreambleLength} + \text{aPLCPHeaderLength}))$$

where

T_{Init_MACDur} is the Duration/ID field value carried in the MAC header of the initial PPDU

An HT STA using L-SIG TXOP protection should use an accurate prediction of the TXOP duration inside the Duration/ID field of the MAC header to avoid inefficient use of the channel capability.

The L-SIG duration of the initial frame shall allow for the longest possible duration of the response frame (i.e., taking into account wrapped +HTC in the case of control response frames). If the actual duration of the response frame is less than this allowed duration, the TXOP holder shall delay transmission of the third PPDU in the L-SIG TXOP protected sequence until a SIFS after this L-SIG duration expires.

NOTE—A result of this step is that a non-HT STA sees a SIFS interval between the end of the first PPDU and the start of the third PPDU.

If the initial frame handshake succeeds (i.e., upon reception of a response frame with L-SIG TXOP protection addressed to the TXOP holder), all HT-mixed format PPDUs transmitted inside an L-SIG TXOP protection protected TXOP shall contain an L-SIG duration that extends to the endpoint indicated by the MAC Duration/ID field. The first PPDU transmitted after a successful initial handshake (i.e., upon reception of a response frame with L-SIG TXOP protection addressed to the TXOP holder) shall have the TXVECTOR FORMAT parameter set to HT_MF.

NOTE—The requirement to use HT_MF for the third PPDU arises as follows. A third-party STA receives the first PPDU, but does not receive any MPDU correctly from it. It sets its NAV based on the L-SIG duration. The STA does not receive the second PPDU. It is necessary for the STA to be able to determine either an L-SIG duration or MAC duration value from the third PPDU in order to protect the remaining time in the TXOP. The ability of the STA to make this determination is enabled by sending the third PPDU using HT-mixed format, containing an L-SIG duration as shown in Figure 9-30.

The TXOP holder should transmit a CF_End frame starting a SIFS after the L-SIG TXOP protected period. This step enables STAs to terminate the EIFS procedure to avoid potential unfairness or a capture effect.

NOTE—This case is not an instance of TXOP truncation, because it is not transmitted to reset the NAV.

9.23.5.3 L-SIG TXOP protection rules at the TXOP responder

On receiving a PPDU containing an L-SIG duration addressed to itself, a TXOP responder that set the L-SIG TXOP Protection Support field to 1 on association shall generate an L-SIG TXOP protection response frame with the L-SIG duration equivalent to the following:

$$\text{L-SIG Duration} = (T_{MACDur} - \text{SIFS} - (\text{aPreambleLength} + \text{aPLCPHeaderLength}))$$

where

T_{MACDur} is the Duration/ID field value carried in the MAC header of frame(s) received in the PPDU that generated the response

A STA shall not transmit a response frame containing an L-SIG duration unless it is in response to a frame that also contained an L-SIG duration.

9.23.5.4 L-SIG TXOP protection NAV update rule

An HT STA that set the L-SIG TXOP Protection Support field to 1 on association that receives a PHY-RXSTART.indication primitive with RXVECTOR parameter FORMAT equal to HT_MF and LSIGVALID equal to true and that receives no valid MPDU from which a Duration/ID field value can be determined shall, when the PHY-RXEND.indication primitive is received, update its NAV to a value equal to the following:

$$\text{L-SIG duration} - (\text{TXTIME} - (\text{aPreambleLength} + \text{aPLCPHeaderLength}))$$

where
TXTIME is the time required to send the entire PPDU

9.24 MAC frame processing

9.24.1 Introduction

Subclauses 9.24.2 to 9.24.9 describe MAC frame and element processing requirements to provide frame, Action frame, element, and subelement extensibility.

9.24.2 Revision level field processing

A MAC entity that receives a frame with a higher revision level than it supports shall discard the frame without indication to the sending STA or to LLC.

9.24.3 Duration/ID field processing

When the contents of a received Duration/ID field, treated as an unsigned integer and without regard for address values, type, and subtype (even when type or subtype contain reserved values), are less than 32 768, the duration value is used to update the network allocation vector (NAV) according to the procedures defined in 9.3.2.4 or 9.19.3.4, as appropriate.

When the contents of a received Duration/ID field, treated as an unsigned integer, are greater than 32 768, the contents are interpreted as appropriate for the frame type and subtype or ignored if the receiving MAC entity does not have a defined interpretation for that type and subtype.

9.24.4 Response to an invalid Action frame

If a STA receives an individually addressed Action frame with an unrecognized Category field or some other syntactic error and the MSB of the Category field equal to 0, then the STA shall return the Action frame to the source without change except that the MSB of the Category field is set to 1.

9.24.5 Operation of the Dialog Token field

A Dialog Token is an integer value that assists a STA in grouping management frames sent or received at different times as part of the same dialog. The algorithm by which the integer value for the dialog is selected is implementation specific, but should be selected in a manner that minimizes the probability of a frame associated with one dialog being incorrectly associated with another dialog.

9.24.6 Element parsing

A STA that encounters an unknown or reserved element ID value in a management frame received without error shall ignore that element and shall parse any remaining management frame body for additional elements with recognizable element ID values.

The MME of a Vendor-Specific Protected Action frame is located at the end of the frame body.

NOTE—It is not necessary to be able to parse the Vendor-Specific Content to locate the MME.

9.24.7 Vendor specific element parsing

A STA receiving a vendor-specific element that it does not support shall ignore the vendor-specific element.

9.24.8 Extensible element parsing

Table 8-54 indicates which elements are considered extensible in future revisions of the standard, by placing a Yes in the Extensible column. A STA that receives an extensible element in which the Length field plus two exceeds the value indicated in Table 8-54 shall discard any part of the element beyond the maximum length indicated in this table and shall otherwise process the element as though this truncated element had been received.

9.24.9 Extensible subelement parsing

A subelement has the structure defined in 8.4.3 and is contained within an element or subelement.

A STA that encounters an unknown, unsupported, or reserved subelement ID value contained in an element or subelement shall ignore the subelement with that subelement ID value and shall continue to parse any remaining element or subelement body for additional subelements with recognizable subelement ID values.

Subelement information is listed in Table 8-60, Table 8-62, Table 8-65, Table 8-68, Table 8-70, Table 8-72, Table 8-73, Table 8-75, Table 8-83, Table 8-85, Table 8-86, Table 8-87, Table 8-89, Table 8-90, Table 8-92, Table 8-115, Table 8-117, Table 8-120, Table 8-207, Table 8-208, Table 8-209, and Table 8-212. These subelement tables indicate which subelements are considered extensible in future revisions of the standard, by placing a Yes in the Extensible column. A STA that receives an extensible subelement in which the Length field exceeds the value indicated in the subelement tables shall discard any part of the subelement beyond the maximum length indicated in the subelement tables and shall otherwise process the subelement as though this truncated subelement had been received.

9.25 Reverse Direction Protocol

9.25.1 Reverse direction (RD) exchange sequence

An RD exchange sequence comprises the following:

a) The transmission of a PPDU by a TXOP holder containing an RD grant (the *RDG PPDU*), which is indicated by the PPDU containing one or more +HTC MPDUs in which the RDG/More PPDU subfield is equal to 1. The STA that transmits this PPDU is known as the *RD initiator*. The rules for an RD initiator apply only during a single RD exchange sequence, i.e., after the transmission of an RDG PPDU and up to the end of the last PPDU in the RD exchange sequence.

b) The transmission of one or more PPDUs (the *RD response burst*) by the STA addressed in the MPDUs of the RDG PPDU. The first (or only) PPDU of the RD response burst contains at most one immediate BlockAck or ACK response frame. The last (or only) PPDU of the RD response burst contains any MPDUs requiring an immediate BlockAck or ACK response. The STA that transmits the RD response burst is known as the *RD responder*. The rules for an RD responder apply only during a single RD exchange sequence, i.e., following the reception of an RDG PPDU and up to the transmission of a PPDU by the RD responder in which the RDG/More PPDU subfield is equal to 0.

c) The transmission of a PPDU by the RD initiator containing an immediate BlockAck or ACK MPDU (the *RD initiator final PPDU*), if so required by the last PPDU of the RD response burst.

NOTE—An RD initiator might include multiple RD exchange sequences within a single TXOP. Each RD exchange sequence within a single TXOP might be addressed to a different recipient, and any single recipient might be given more than one RDG within a single TXOP.

An example of an RD exchange sequence is given in S.3.

9.25.2 Support for RD

Support of the RD feature is an option for an HT STA. It is optional in the sense that a TXOP holder is never required to generate an RDG, and a STA receiving an RDG is never required to use the grant.

Support of the RD feature as an RD responder is indicated using the RD Responder subfield of the HT Extended Capabilities field of the HT Capabilities element. A STA shall set the RD Responder subfield to 1 in frames that it transmits containing the HT Capabilities element if dot11RDResponderOptionImplemented is true. Otherwise, the STA shall set the RD Responder subfield to 0.

9.25.3 Rules for RD initiator

An RDG shall not be present unless the MPDU carrying the grant, or every MPDU carrying the grant in an A-MPDU, matches one of the following conditions:

- A QoS data MPDU with the Ack Policy field equal to any value except PSMP Ack (i.e., including Implicit Block Ack Request), or
- A BlockAckReq related to an HT-immediate Block Ack agreement, or
- An MPDU not needing an immediate response (e.g., BlockAck under an HT-immediate Block Ack agreement, or Action No Ack).

An RDG shall not be present within a PSMP sequence.

NOTE 1—These rules together with the rules in 8.6.3 cause an RDG to be delivered in a PPDU that either requires no immediate response or requires an immediate BlockAck or ACK response.

NOTE 2—An RD initiator is not required to examine the RD Responder field of a potential responder before deciding whether to send a PPDU to that STA in which the RDG/More PPDU subfield is set to 1.

NOTE 3—An RD initiator is required according to 9.9 to examine the +HTC Support field of a potential responder before deciding whether to send a PPDU to that STA in which the RDG/More PPDU subfield is set to 1.

Transmission of a +HTC frame by an RD initiator with the RDG/More PPDU subfield equal to 1 (either transmitted as a non-A-MPDU frame or within an A-MPDU) indicates that the duration indicated by the Duration/ID field is available for the RD response burst and RD initiator final PPDU (if present).

An RD initiator that sets the RDG/More PPDU field to 1 in a +HTC frame shall set the AC Constraint subfield to 1 in that frame if the TXOP was gained through the EDCA channel access mechanism and shall otherwise set it to 0.

An RD initiator shall not transmit a +HTC frame with the RDG/More PPDU subfield set to 1 that requires a response MPDU that is not one of the following:

- Ack
- Compressed BlockAck

Subject to TXOP constraints, after transmitting an RDG PPDU, an RD initiator may transmit its next PPDU as follows:

a) *Normal continuation:* The RD initiator may transmit its next PPDU a minimum of a SIFS after receiving a response PPDU that meets one of the following conditions:
   1) Contains one or more correctly received +HTC frames with the RDG/More PPDU subfield equal to 0, or
   2) Contains one or more correctly received frames that are capable of carrying the HT Control field but did not contain an HT Control field, or
   3) Contains a correctly received frame that requires an immediate response
b) *Error recovery:* The RD initiator may transmit its next PPDU when the CS mechanism (see 9.3.2.1) indicates that the medium is idle at the TxPIFS slot boundary (defined in 9.3.7) (this transmission is a continuation of the current TXOP).

NOTE 1—Error recovery of the RDG mechanism is the responsibility of the RD initiator.

NOTE 2—After transmitting a PPDU containing an RDG, if the response is corrupted so that the state of the RDG/More PPDU subfield is unknown, the RD initiator of the RD exchange is not allowed to transmit after a SIFS interval. Transmission can occur a PIFS interval after deassertion of CS.

NOTE 3—After transmitting a PPDU requiring a response but not containing an RDG, the state of the RDG/More PPDU subfield in the response does not affect the behavior of the RD initiator.

A STA that transmits a QoS +CF-ACK data frame according to the rules in 9.19.3.5 may also include an RDG in that frame provided that

- It is a non-A-MPDU frame, and
- The target of the +CF-ACK is equal to the Address 1 field of the frame.

NOTE—The RD initiator can transmit a CF-End frame according to the rules for TXOP truncation in 9.19.2.7 following a RD transmit sequence. An RD responder never transmits a CF-End.

9.25.4 Rules for RD responder

An RD responder shall transmit the initial PPDU of the RD response burst a SIFS after the reception of the RDG PPDU. PPDUs in a response burst are separated by SIFS or RIFS. The RIFS rules in the RD are the same as in the forward direction; the use of RIFS is constrained as defined in 9.3.2.3.2 and 9.23.3.3.

NOTE—The transmission of a response by the RD responder does not constitute a new channel access but a continuation of the RD initiator's TXOP. An RD responder ignores the NAV when responding to an RDG.

The recipient of an RDG may decline the RDG by

- Not transmitting any frames following the RDG PPDU when no response is otherwise required, or
- Transmitting a control response frame with the RDG/More PPDU subfield set to 0, or
- Transmitting a control response frame that contains no HT Control field

An RD responder may transmit a +CF-ACK non-A-MPDU frame in response to a non-A-MPDU QoS Data +HTC MPDU that has the Ack Policy field equal to Normal Ack and the RDG/More PPDU subfield equal to 1.

The RD responder shall verify that its PPDU transmission(s) and any expected responses fit entirely within the remaining TXOP duration, as indicated in the Duration/ID field of MPDUs within the RDG PPDU.

An RD responder shall not transmit an MPDU (either individually or aggregated within an A-MPDU) that is not one of the following:

- ACK
- Compressed BlockAck
- Compressed BlockAckReq
- QoS data
- Management

If the AC Constraint subfield is equal to 1, the RD responder shall transmit data frames of only the same AC as the last frame received from the RD initiator. For a BlockAckReq or BlockAck frame, the AC is determined by examining the TID field. For a management frame, the AC is AC_VO. The RD initiator shall not transmit a +HTC MPDU with the RDG/More PPDU subfield set to 1 from which the AC cannot be determined. If the AC Constraint subfield is equal to 0, the RD responder may transmit data frames of any TID.

During an RDG, the RD responder shall not transmit any frames with an Address 1 field that does not match the MAC address of the RD initiator.

If an RDG PPDU also requires an immediate BlockAck response, the BlockAck response frame shall be included in the first PPDU of the response.

When a PPDU is not the final PPDU of a response burst, an HT Control field carrying the RDG/More PPDU subfield set to 1 shall be present in every MPDU within the PPDU capable of carrying the HT Control field. The last PPDU of a response burst shall have the RDG/More PPDU subfield set to 0 in all +HTC MPDUs contained in that PPDU.

The RD responder shall not set the RDG/More PPDU subfield to 1 in any MPDU in a PPDU that contains an MPDU that requires an immediate response.

NOTE— If the RD responder transmits a PPDU that expects a transmission by the RD initiator after SIFS and no such transmission is detected, the RD responder has to wait for either another RDG or its own TXOP before it can retry the exchange.

After transmitting a PPDU containing one or more +HTC MPDUs in which the RDG/More PPDU subfield is equal to 0, the RD responder shall not transmit any more PPDUs within the current response burst.

NOTE— If an RD-capable STA that is not the TXOP holder receives a PPDU that does not indicate an RDG, there is no difference in its response compared to a STA that is not RD-capable.

9.26 PSMP Operation

9.26.1 Frame transmission mechanism during PSMP

9.26.1.1 PSMP frame transmission (PSMP-DTT and PSMP-UTT)

The attribute aDTT2UTTTime is the minimum time between the end of the PSMP-DTT and the start of a PSMP-UTT addressed to the same STA. This value represents the minimum time the STA is provided to react to Multi-TID BlockAck, BlockAck, Multi-TID BlockAckReq, BlockAckReq, and data frames received during the PSMP-DTT with data, BlockAck, BlockAckReq, Multi-TID BlockAckReq, and Multi-TID BlockAck frames transmitted in the PSMP-UTT. In a PSMP sequence, if the traffic conditions are such that the time between the PSMP-DTT and PSMP-UTT of a STA would otherwise be less than the value of aDTT2UTTTime, the AP shall delay the start of entire PSMP-UTT phase to meet this requirement.

A PSMP sequence may be used to transmit group addressed frames along with individually addressed frames. Individually addressed frames shall be scheduled after group addressed frames.

In a PSMP frame, the STA_ID subfields of all its STA Info fields with STA_INFO Type equal to 2 (individually addressed) shall be unique, i.e., each STA identified in the PSMP frame is identified exactly once.

Individually addressed entries in the PSMP frame should have their PSMP-DTT and PSMP-UTT start offsets scheduled to minimize the number of on/off transitions or to maximize the delay between their PSMP-DTT and PSMP-UTT periods. Entries that have only PSMP-DTT should be scheduled closer to the start of the PSMP-DTTs. Entries that have only PSMP-UTT should be scheduled toward the end of PSMP-UTTs. Entries that have both PSMP-DTT and PSMP-UTT should be scheduled closer to the transition point from downlink to uplink transmissions.

NOTE—For effective resource allocation, the AP needs to precisely estimate the PSMP-UTT duration for each STA using the information indicated in a TSPEC, such as Minimum Data Rate, Mean Data Rate, Peak Data Rate, Burst Size, and Delay Bound fields. However, when the traffic characteristic is quite bursty (e.g., a real-time video application), precise estimation of PSMP-UTT duration is difficult without timely and frequent feedback of the current traffic statistics. In order to avoid wasting the available bandwidth by overestimating the PSMP-UTT duration, the AP can allocate the minimum amount of time to each STA using the PSMP-UTT Duration subfield in the PSMP frame, based on the value of the Minimum Data Rate field specified in the TSPEC. When the STA receives the PSMP frame, it decides whether the allocated resource indicated by the PSMP-UTT duration is sufficient for its queued data. If the allocated resource is sufficient, the STA can transmit all the queued data at the allocated time.

Frames of different TIDs may be transmitted within a PSMP-DTT or PSMP-UTT allocation of a PSMP sequence without regard to user priority.

Within a PSMP-DTT or PSMP-UTT between HT STAs, BlockAckReq and BlockAck frames for which an HT-immediate Block Ack agreement exists shall be the multi-TID variants, i.e., Multi-TID BlockAckReq and Multi-TID BlockAck, respectively. Within a PSMP-DTT or PSMP-UTT between STAs where one is not an HT STA, BlockAckReq and BlockAck frames shall be exchanged through the use of an immediate Block Ack agreement and shall be the basic variants, i.e. Basic BlockAck Req and Basic BlockAck, respectively.

9.26.1.2 PSMP downlink transmission (PSMP-DTT)

During a PSMP sequence, a STA shall be able to receive frames during its scheduled PSMP-DTT and is not required to be able to receive frames at other times.

The AP shall verify that any transmissions within a PSMP sequence to a STA participating in the PSMP sequence occur wholly within the STA’s PSMP-DTT.

The PSMP-DTT may contain one or more PPDUs, each of which may contain either an A-MPDU or a single (non-A-MPDU) MPDU. Data may be transmitted using either format, provided that the format is supported by both the transmitter and the receiver.

PPDUs within a PSMP-DTT may be separated using RIFS or SIFS. The use of RIFS is limited as defined in 9.3.2.3.2 and 9.23.3.3.

Each PSMP-DTT shall contain only frames addressed to the RA signaled by the corresponding STA_INFO field. PPDUs from adjacent PSMP-DTTs shall be separated by at least SIFS. In other words, PPDUs to a different RA are separated by at least SIFS.

9.26.1.3 PSMP uplink transmission (PSMP-UTT)

A STA that has frames to send that are valid for transmission within the PSMP-UTT shall start transmission without performing CCA and regardless of NAV at the start of its PSMP-UTT.

The STA shall complete its transmission within the allocated PSMP-UTT, even if it has more data queued than can be transmitted during its allocated PSMP-UTT.

NOTE—PSMP-UTT is a scheduled transmission period for the STA, and transmission within a PSMP-UTT does not imply that the STA is a TXOP holder. This lack of being a TXOP holder disallows a STA from using TXOP truncation during PSMP-UTT.

The uplink schedule in a PSMP frame shall include an interval between the end of one PSMP-UTT and the start of the following PSMP-UTT within the same PSMP sequence. This interval shall be either aIUStime or SIFS. The aIUStime value shall not be used unless the use of RIFS is permitted, as defined in 9.23.3.3. The PSMP-UTT Duration subfield in the PSMP frame does not include this interval.

PPDUs transmitted within a PSMP-UTT may be separated using RIFS or SIFS. The use of RIFS is limited as defined in 9.3.2.3.2 and 9.23.3.3.

An AP may transmit a PSMP frame (called a *PSMP recovery frame*) during a PSMP-UTT when both of the following conditions are met:

— The CS mechanism (see 9.3.2.1) indicates that the medium is idle at the TxPIFS slot boundary (defined in 9.3.7) after the start of the PSMP-UTT, and
— The PSMP-UTT duration is longer than the total time of the PSMP recovery frame plus PIFS.

The PSMP recovery frame shall not modify the schedule of a STA that is not scheduled to use this PSMP-UTT. The schedules of other STAs shall remain unchanged. The PSMP recovery frame may include:

a) A modified PSMP-UTT (and/or PSMP-DTT) for the currently scheduled STA by adjusting the time remaining by a PIFS interval plus the duration of the PSMP recovery frame, and
b) PSMP-UTTs for other STAs that were originally scheduled after this PSMP-UTT in the PSMP sequence in which the PSMP-UTT start offset values are reduced by the time difference between the end of the original PSMP frame and the end of the PSMP recovery frame.

If the currently scheduled PSMP-UTT duration is shorter than the total time of PSMP recovery frame plus PIFS, no PSMP recovery frame is transmitted.

Figure 9-31 illustrates a PSMP sequence with and without PSMP recovery.

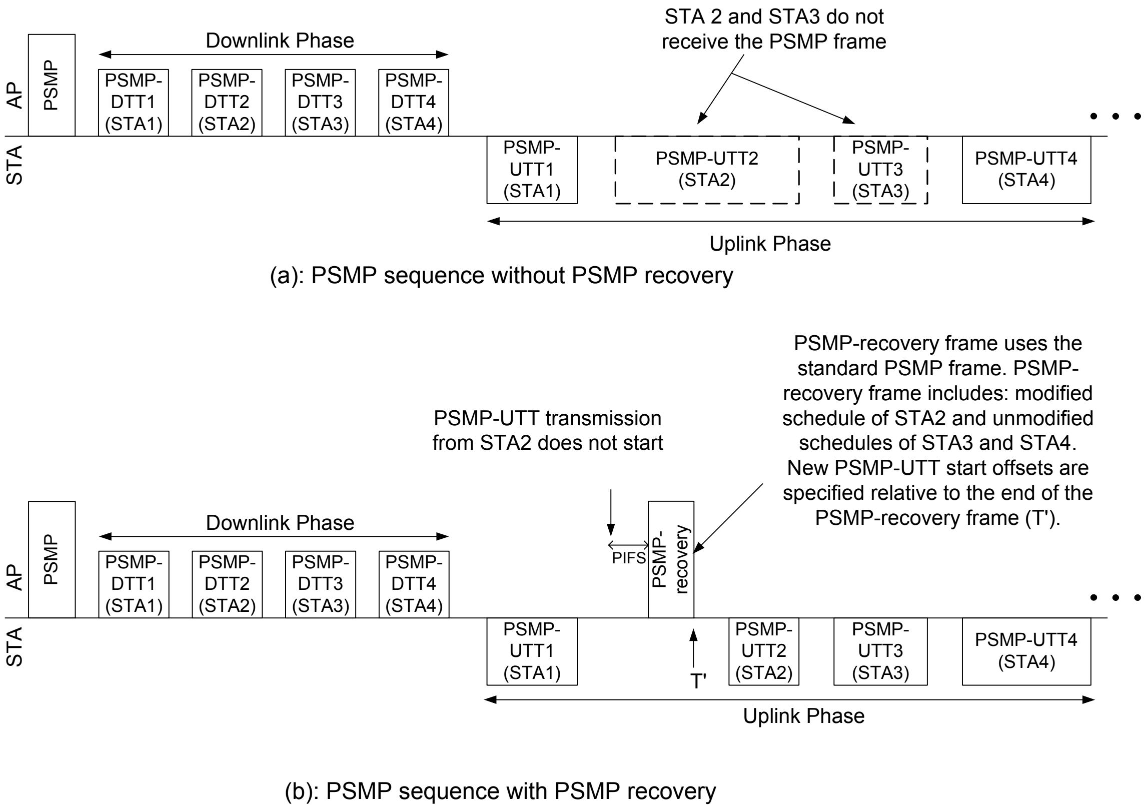

AP
STA
PSMP
Downlink Phase
PSMP-DTT1 (STA1)
PSMP-DTT2 (STA2)
PSMP-DTT3 (STA3)
PSMP-DTT4 (STA4)
STA 2 and STA3 do not receive the PSMP frame
PSMP-UTT1 (STA1)
PSMP-UTT2 (STA2)
PSMP-UTT3 (STA3)
PSMP-UTT4 (STA4)
Uplink Phase
• • •
(a): PSMP sequence without PSMP recovery
PSMP-UTT transmission from STA2 does not start
PSMP-recovery frame uses the standard PSMP frame. PSMP-recovery frame includes: modified schedule of STA2 and unmodified schedules of STA3 and STA4. New PSMP-UTT start offsets are specified relative to the end of the PSMP-recovery frame (T').
PIFS
PSMP-recovery
T'
(b): PSMP sequence with PSMP recovery

Figure 9-31—Illustration of PSMP sequence with and without PSMP recovery

9.26.1.4 PSMP burst

After transmission of an initial PSMP sequence, additional PSMP sequences may be transmitted by the AP in order to support resource allocation and error recovery. An initial PSMP sequence followed by one or more PSMP sequences is termed a *PSMP burst* and is shown in Figure 9-32.

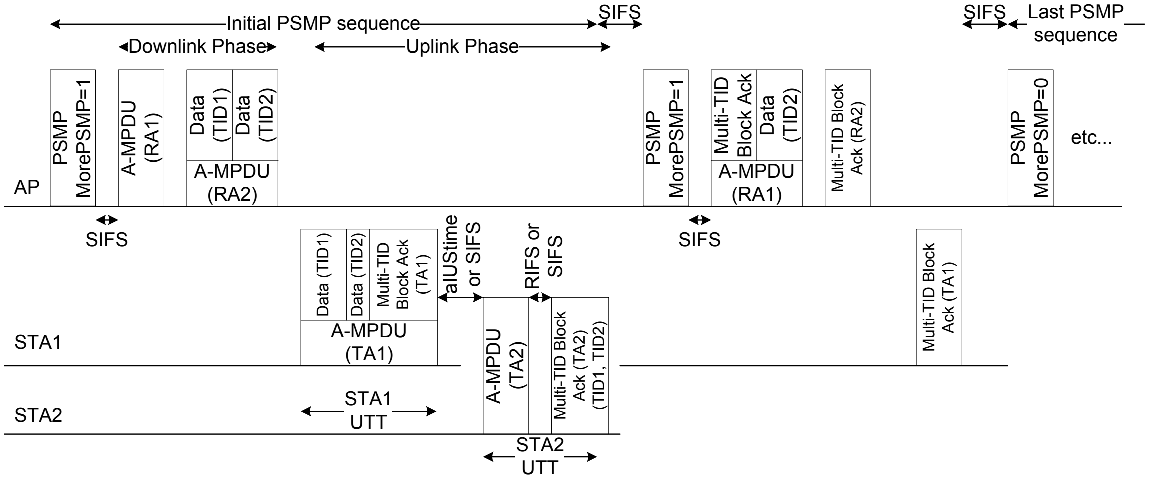


Figure 9-32—PSMP burst

A STA shall not transmit a +HTC MPDU in which the RDG/More PPDU subfield is set to 1 during a PSMP burst.

An AP may transmit a CF-End frame a SIFS period after the end of a PSMP sequence to end the PSMP burst.

NOTE—A non-AP STA does not transmit a CF-End frame during the PSMP burst because it is not a TXOP holder during its PSMP-UTT.

During the PSMP-DTT or PSMP-UTT, a STA shall not transmit a frame unless it is one of the following:

- Multi-TID BlockAck under HT-immediate policy
- Multi-TID BlockAckReq under HT-immediate policy
- BlockAck under an immediate policy with the BA Ack Policy subfield set to 1 (representing No Acknowledgment)
- BlockAckReq under an immediate policy with the BAR Ack Policy subfield set to 1 (representing No Acknowledgment)
- QoS data
- PSMP (a PSMP recovery frame as described in 9.26.1.3)
- BlockAckReq under HT-delayed policy with the BAR Ack Policy subfield set to 1 (representing No Acknowledgment)
- BlockAck under HT-delayed policy with the BA Ack Policy subfield set to 1 (representing No Acknowledgment)
- An MPDU that does not require an immediate response (e.g., Management Action No Ack)

NOTE—An AP can gain access to the channel after a PIFS in order to start transmission of a PSMP sequence.

9.26.1.5 Resource allocation within a PSMP burst

If the allocated PSMP-UTT duration is not long enough for its queued data, the STA transmits only the part of the queued data that fits within the allocated PSMP-UTT duration and may transmit a resource request to the AP within that PSMP-UTT. The resource request is communicated by setting either the Queue Size field or the TXOP Duration Request field of the QoS Control field that is carried in a QoS data frame (see Figure 9-33).

If a STA receives an PSMP-UTT that is not long enough to transmit data from its queues, it may transmit within the PSMP-UTT a QoS Null frame containing information about the state of its transmit queues.

NOTE 1—An HT AP might use this information to schedule a PSMP-UTT either in the current PSMP burst or a later PSMP burst.

NOTE 2—An HT AP might allocate a PSMP-UTT duration in the next PSMP sequence based on the resource request from the STA sufficient to allow transmission of the remaining queued data.

NOTE 3—The PSMP burst supports retransmission as well as additional resource allocation (see Figure 9-34). Frames transmitted under an HT-immediate Block Ack agreement during the PSMP-DTT are acknowledged by a Multi-TID BlockAck frame during the PSMP-UTT period of the current PSMP sequence. Frames transmitted under an immediate Block Ack agreement during the PSMP-DTT are acknowledged by a Basic BlockAck during the PSMP-UTT period of the current PSMP sequence. Frames transmitted under an HT-immediate Block Ack agreement during the PSMP UTT can be acknowledged using a Multi-TID BlockAck frame during the PSMP-DTT period of the next PSMP sequence. Frames transmitted under an immediate Block Ack agreement during the PSMP-UTT can be acknowledged using a Basic BlockAck during the PSMP-DTT period of the next PSMP sequence. Any failed transmissions during the PSMP-DTT or PSMP-UTT periods can be respectively retransmitted during the PSMP-DTT or PSMP-UTT period of the next PSMP sequence.

Figure 9-33 and Figure 9-34 illustrate the operation of resource allocation. STA1 requests the AP to provide additional resources in its transmission to the AP. The box labeled "Queued data" represents the duration that would be required to transmit data queued for transmission at the STA. In Figure 9-34, since the AP does not receive an acknowledgment from STA2, the AP retransmits the data addressed to STA2 and also allocates resources to STA2 so that STA2 can transmit in the next PSMP sequence.

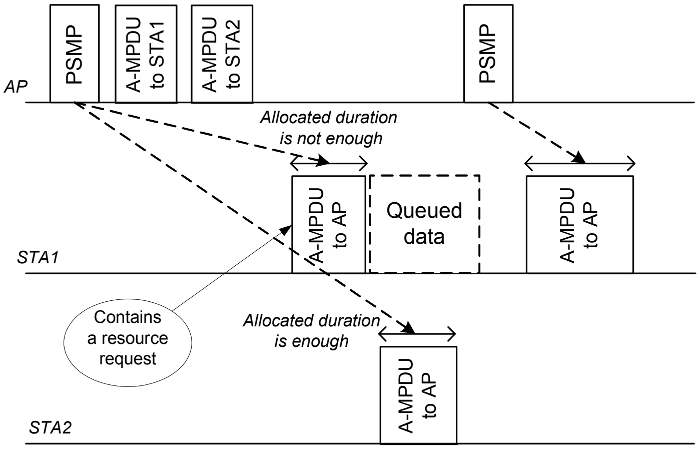


Figure 9-33—PSMP burst showing resource allocation

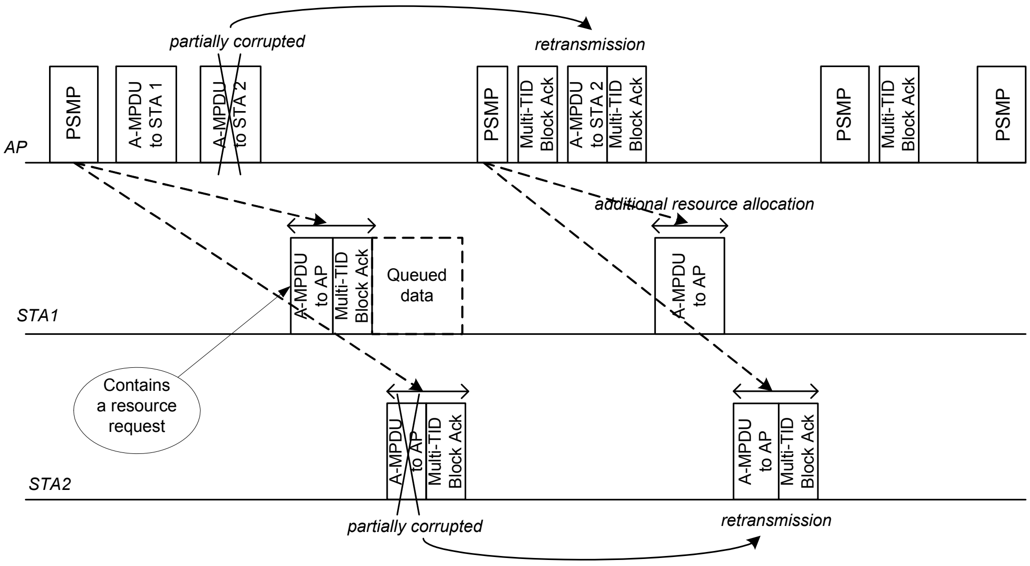


Figure 9-34—PSMP burst showing retransmission and resource allocation

9.26.1.6 PSMP-UTT retransmission

An AP transmits BlockAck or Multi-TID BlockAck responses, if any, to a STA's PSMP-UTT data transmissions under an immediate or HT-immediate Block Ack agreement, respectively, in the PSMP-DTT of a subsequent PSMP sequence.

NOTE—An AP might reserve a PSMP-UTT in a subsequent PSMP sequence to allow a STA to retransmit failed frames. The STA can retransmit failed frames in a PSMP sequence of the current PSMP burst if a PSMP-UTT reservation is present or in a subsequent SP.

A STA that cannot complete its retransmissions in the last PSMP sequence of the PSMP burst because not enough time is allocated in its PSMP-UTT may transmit the data outside any PSMP sequence.

NOTE 1—In the case of uplink frames transmitted outside the scheduled SP, the Multi-TID BlockAck frame that acknowledges these frames is delivered in the PSMP-DTT within the next SP.

NOTE 2—A non-AP STA might transmit data outside the PSMP sequence. The acknowledgment of such frames is based on their Ack Policy field value and whether a Block Ack agreement has been established, as follows:

— An Ack Policy of Block Ack, Normal Ack, or Implicit Block Ack Request results in the behavior defined in 8.2.4.5.4.
— An Ack Policy of PSMP Ack causes the AP to record the received data frame and results in the transmission of a Multi-TID BlockAck frame in the next PSMP-DTT allocated to the STA.

9.26.1.7 PSMP acknowledgment rules

A non-AP STA shall transmit a Multi-TID BlockAck frame during its PSMP-UTT for data received with the ACK Policy field set to PSMP Ack or for TIDs in a received Multi-TID BlockAckReq frame for which a BlockAck (Compressed BlockAck or Multi-TID BlockAck) has not yet been transmitted. An AP shall transmit a Multi-TID BlockAck frame during a PSMP-DTT addressed to the STA for the data received from that STA with the ACK Policy field set to PSMP Ack or for TIDs in a Multi-TID BlockAckReq frame received from that STA for which a BlockAck (Compressed BlockAck or Multi-TID BlockAck) has not yet been transmitted.

Data sent and received by a non-AP STA within a PSMP sequence may be contained in an A-MPDU that contains MPDUs of multiple TIDs. Frames of differing TIDs may be transmitted in the same PSMP-DTT or PSMP-UTT and are not subject to AC prioritization.

The subtype subfield of data frames and the Ack Policy subfield of QoS data frames transmitted during either PSMP-DTT or PSMP-UTT periods are limited by the following rules:

- A QoS data frame transmitted under an immediate or HT-immediate Block Ack agreement during either a PSMP-DTT or a PSMP-UTT shall have one of the following Ack Policy values: PSMP Ack or Block Ack.
- A QoS data frame transmitted under an HT-delayed Block Ack agreement during either a PSMP-DTT or a PSMP-UTT shall have the Ack Policy field set to Block Ack.
- A data frame with the RA field containing an individual address transmitted during either a PSMP-DTT or a PSMP-UTT and for which no Block Ack agreement exists shall be a QoS data subtype and shall have the Ack Policy field set to No Ack.
- The Ack Policy field of a QoS data frame transmitted during a PSMP sequence shall not be set to either Normal ACK or Implicit Block ACK.

All TID values within a Multi-TID BlockAck frame or Multi-TID BlockAckReq frame shall identify a Block Ack agreement that is HT-immediate. QoS data frames transmitted with Ack Policy field equal to PSMP Ack shall have a TID value that identifies a Block Ack agreement that is immediate or HT-immediate BlockAck.

NOTE—In this case, HT-immediate relates to the keeping of acknowledgment state for timely generation of a Multi-TID BlockAck frame. It does not imply that there is any response mechanism for sending a Multi-TID BlockAck frame after a SIFS interval. The timing of any response is determined by the PSMP schedule.

Acknowledgment for data transmitted under an immediate or HT-immediate Block Ack agreement may be requested implicitly using PSMP Ack setting of the Ack Policy field in data frames or explicitly with a Basic BlockAckReq or Multi-TID BlockAckReq frame. An AP that transmits data frames with the Ack Policy field equal to PSMP Ack or that transmits a Basic BlockAckReq or Multi-TID BlockAckReq frame addressed to a STA in a PSMP-DTT shall allocate sufficient time for a Basic BlockAck or Multi-TID BlockAck transmission, respectively, in a PSMP-UTT allocated to that STA within the same PSMP sequence. A STA that has correctly received a PSMP frame and that receives a QoS data MPDU with the Ack Policy field equal to PSMP Ack or that receives a Basic BlockAckReq or Multi-TID BlockAckReq frame shall transmit a Basic BlockAck frame or Multi-TID BlockAck frame, respectively, in the PSMP-UTT of the same PSMP sequence.

NOTE 1—If the STA does not receive the PSMP frame, it might still receive the downlink data, in which case it can record the status of the data in its Block Ack buffer, but it cannot transmit a Multi-TID BlockAck frame.

NOTE 2—A Multi-TID BlockAck frame or Multi-TID BlockAckReq frame might contain any TID related to an HT-Immediate Block ACK agreement regardless of the contents of any prior Multi-TID BlockAck or Multi-TID BlockAckReq or QoS data transmission.

An AP that receives a QoS data MPDU with the Ack Policy field equal to PSMP Ack during a PSMP-UTT shall transmit a Basic BlockAck or Multi-TID BlockAck response in the next PSMP-DTT that it schedules for that STA, except if it has transmitted a BlockAck for such TIDs to the STA outside the PSMP mechanism.

NOTE 1—The exception may occur if the non-AP STA transmits one or more BlockAckReq frames or QoS data frames with Ack Policy set to Implicit Block Ack outside the PSMP mechanism.

NOTE 2—An AP might receive a Multi-TID BlockAck frame in the PSMP-UTT of the current PSMP sequence. If the Multi-TID BlockAck frame indicates lost frames or if the AP does not receive an expected Multi-TID BlockAck frame, the AP might schedule and retransmit those frames in a PSMP sequence within the current PSMP burst or in the next SP.

A Multi-TID BlockAck frame shall include all the TIDs for which data were received with ACK Policy field equal to PSMP Ack and for the TIDs listed in any Multi-TID BlockAckReq frame received during the previous PSMP-DTT (STA) or PSMP-UTT (AP). The originator may ignore the bitmap for TIDs in the Multi-TID

BlockAck frame for which the originator has not requested a Multi-TID BlockAck frame to be present either implicitly (by the transmission of data MPDUs with the Ack Policy field set to PSMP Ack) or explicitly (by the transmission of a Multi-TID BlockAckReq frame).

If a BlockAckReq frame for an HT-delayed Block Ack agreement is transmitted during a PSMP sequence, the BAR Ack Policy subfield of the BlockAckReq frame shall be set to the value representing No Acknowledgment.

NOTE—Multi-TID BlockAck and Multi-TID BlockAckReq frames transmitted during PSMP use the Normal Acknowledgment setting of the BA Ack Policy or BAR Ack Policy subfield.

9.26.1.8 PSMP group addressed transmission rules

9.26.1.8.1 Rules at the AP

This subclause defines rules that shall be followed by a PSMP-capable AP for the transmission of group addressed frames (data and management) during a PSMP sequence.

Each separate group address for which frames are transmitted during a PSMP sequence shall have a single STA_INFO record with STA_INFO Type set to 1 (group addressed) present in the PSMP frame and transmit frames with the matching Address 1 field only during the PSMP-DTT indicated in this record.

The DA of the PSMP shall be set to the broadcast address, except if the PSMP contains only a single non-null PSMP-DTT and this PSMP-DTT contains frames for a group address, in which case the DA of the PSMP frame may be set to this group address.

NOTE—The transmission of a group addressed frame within a PSMP sequence does not change the rules regarding when that frame can be transmitted. In other words, if there is a power-saving STA in the BSS, the group addressed frame is transmitted following a DTIM beacon according to the rules in 10.2.1.

9.26.1.8.2 Rules at the STA

This subclause defines rules that shall be followed by a PSMP-capable STA for the reception of group addressed frames during a PSMP sequence.

The STA shall be awake to receive during all PSMP-DTTs identified by a group addressed STA_INFO record where the PSMP Group Address ID subfield matches the LSBs of any address within its dot11GroupAddressesTable.

9.26.2 Scheduled PSMP

A PSMP session exists while any periodic TS exists that was established by a TSPEC with the APSD field equal to 0 and the Schedule field equal to 1 (representing Scheduled PSMP). The creation of a PSMP session is described in 10.4.6.

While one or more PSMP sessions exist with the same SP, the AP shall periodically initiate a PSMP sequence by transmitting a PSMP frame using the SP indicated to the STA in response to the received TSPEC. Under S-PSMP rules, the AP shall not transmit a PSMP frame containing a STA_INFO record addressed to a STA unless the transmission occurs within a SP of that STA. The PSMP-DTT and PSMP-UTT allocated to a STA shall occur within a SP of that STA.

NOTE—An AP might simultaneously maintain multiple PSMP sessions with distinct SIs. The SIs of an AP's PSMP sessions are multiples of the SI granularity. An AP might combine the schedule of multiple PSMP sessions into a single PSMP frame if the start times of the PSMP sessions coincide. For example, the schedule carried by a PSMP frame related to a PSMP session at 20 ms and 30 ms SIs can be combined into a single PSMP frame once every 3 SIs of PSMP session at 20 ms or once every 2 SIs of the PSMP session at 30 ms.

The start time of a PSMP sequence should be aligned with the start time of the SP.

9.26.3 Unscheduled PSMP

An HT AP may start an unscheduled PSMP sequence that includes STAs that are PSMP-capable at any time that these STAs are awake.

NOTE—A STA in power save is awake as defined in 10.2.1.5 (U-APSD, S-APSD), as defined in 10.2.1.8 (PS-poll), or during a DTIM period.

U-APSD STAs can signal the queue size or TXOP duration required to transmit its queued data to the AP in the QoS Control field of the trigger frame. This information might be used by the AP to estimate the duration of the PSMP-UTT so that the STA can transmit the queued data.

All the behavior defined in 10.2.1.5, 10.2.1.6, and 10.2.1.10 applies to unscheduled PSMP with the following exceptions:

- PSMP allows the STA to sleep during PSMP-DTTs and PSMP-UTTs in which it has no interest.
- In addition to the EOSP mechanism, the AP may indicate the end of a SP through the transmission of a PSMP frame with the More PSMP field set to 0 or by transmission of a CF-End frame when a PSMP frame was expected.

9.27 Sounding PPDUs

A sounding PPDU is a PPDU for which the SOUNDING parameter of the corresponding RXVECTOR or TXVECTOR has the value SOUNDING. Sounding PPDUs are transmitted by STAs to enable the receiving STAs to estimate the channel between the transmitting STA and the receiving STA.

A STA transmits sounding PPDUs when it operates in the following roles:

- MFB requester (see 9.28.2)
- Beamformee responding to a training request, calibration initiator, or responder involved in implicit transmit beamforming (see 9.29.2.2, 9.29.2.3, and 9.29.2.4)
- Beamformer involved in explicit transmit beamforming (see 9.29.3)
- ASEL transmitter and ASEL sounding-capable transmitter involved in ASEL (see 9.30.2)

A STA receives sounding PPDUs when it operates in the following roles:

- MFB responder (see 9.28.2)
- Beamformer sending a training request, calibration initiator, or responder involved in implicit transmit beamforming (see 9.29.2.2, 9.29.2.3, and 9.29.2.4)
- Beamformee involved in explicit transmit beamforming (see 9.29.3)
- Transmit ASEL responder and ASEL receiver involved in ASEL (see 9.30.2)

When transmitting a sounding PPDU, the transmitting STA follows the rules stated below to determine the maximum number of space-time streams for which channel coefficients can be simultaneously estimated:

- When transmitting a sounding PPDU that
  - Contains a +HTC frame with the MRQ subfield equal to 1, or
  - Is sent as a response to a +HTC frame with the TRQ field equal to 1, or
  - Is sent during a calibration sounding exchange, or
  - Is sent by a beamformer involved in explicit transmit beamforming, or
  - Is sent in transmit or receive ASEL exchanges,
- Then,

- If the sounding PPDU is not an NDP sounding PPDU, the NUM_EXTEN_SS parameter in the TXVECTOR shall not be set to a value greater than the limit indicated by the Channel Estimation Capability subfield in the Transmit Beamforming Capabilities field transmitted by the STA that is the intended receiver of the sounding PPDU.

  NOTE—The maximum number of space-time streams for which channel coefficients can be simultaneously estimated using the HT-LTFs corresponding to the data portion of the packet is limited by the Rx MCS Bitmask subfield of the Supported MCS Set field and by the Rx STBC subfield of the HT Capabilities Info field. Both fields are part of the HT Capabilities element.

- If the sounding PPDU is an NDP, the number of spatial streams corresponding to the MCS parameter of the TXVECTOR shall not exceed the limit indicated by the Channel Estimation Capability subfield in the Transmit Beamforming Capabilities field transmitted from the STA that is the intended receiver of the NDP (see 9.31.2 for details on setting the MCS parameter).

If a STA sets the Receive Staggered Sounding Capable bit in the Transmit Beamforming Capabilities field to 1, the STA shall set the Channel Estimation Capability bit in the Transmit Beamforming Capabilities field to indicate a dimension that is greater than or equal to the dimension indicated by the Supported MCS Set field of the HT Capabilities element.

9.28 Link adaptation

9.28.1 Introduction

To fully exploit MIMO channel variations and transmit beamforming on a MIMO link, a STA can request that another STA provide MIMO channel sounding and MFB.

Link adaptation may be supported by immediate response or delayed response as described below. Unsolicited MFB is also possible.

- *Immediate:* An immediate response occurs when the MFB responder transmits the response in the TXOP obtained by the TXOP holder. This approach allows the MFB requester to obtain the benefit of link adaptation within the same TXOP.
- *Delayed:* A delayed response occurs when the MFB responder transmits the response in the role of a TXOP holder in response to an MRQ in a previous TXOP obtained by the MFB requester.
- *Unsolicited:* An unsolicited response occurs when a STA sends MFB independent of any preceding MRQ.

9.28.2 Link adaptation using the HT Control field

A STA that supports link adaptation using the HT Control field shall set the MCS Feedback field of the HT Extended Capabilities field to Unsolicited or Both, depending on its specific MFB capability, in HT Capabilities elements that it transmits. MRQs shall not be sent to STAs that have not advertised support for link adaptation. A STA whose most recently transmitted MCS Feedback field of the HT Extended capabilities field of the HT Capabilities element is equal to Unsolicited or Both may transmit unsolicited MFB in any frame that contains a +HTC field.

The MFB requester may set the MRQ subfield to 1 in the MAI subfield of the HT Control field of a +HTC frame to request a STA to provide MFB. In each MRQ, the MFB requester shall set the MSI subfield in the MAI subfield to a value in the range 0 to 6. How the MFB requester chooses the MSI value is implementation dependent.

NOTE—The MFB requester might use the MSI subfield as an MRQ sequence number, or it might implement any other encoding of the field.

The appearance of more than one instance of an HT Control field with the MRQ subfield equal to 1 within a single PPDU shall be interpreted by the receiver as a single request for MFB.

An MFB requester shall transmit +HTC frames with the MRQ subfield equal to 1 in one of the following ways:

- Within a sounding PPDU, or
- With the NDP Announcement subfield in the +HTC frame set to 1 and following the +HTC frame by an NDP transmission

The number of HT-LTFs sent in the sounding PPDU or in the NDP is determined by the total number of spatial dimensions to be sounded, including any extra spatial dimensions beyond those used by the data portion of the frame.

An MFB-capable STA (identified by the MCS Feedback field in Extended HT Capabilities Info field equal to 3) shall support the following:

- MFB estimate computation and feedback on the receipt of MRQ (MRQ=1 in +HTC) in a sounding PPDU for which the RXVECTOR NUM_EXTEN_SS parameter contains 0 in the PHYRXSTART.indication primitive.
- MFB estimate computation and feedback on the receipt of MRQ (MRQ=1 in +HTC) in a staggered sounding PPDU if this STA declares support for receive staggered sounding by setting the Receive Staggered Sounding Capable subfield of the Transmit Beamforming Capabilities field to 1.
- MFB estimate computation and feedback on the receipt of NDP (see 9.31) if this STA declares support for receiving NDP sounding by setting the Receive NDP Capable subfield of the Transmit Beamforming Capabilities field to 1. The MFB requester shall set the MRQ subfield to 1 in the frame where the NDP Announcement subfield is equal to 1.

On receipt of a +HTC frame with the MRQ subfield equal to 1, an MFB responder initiates computation of the MCS estimate based on the associated sounding PPDU and labels the result of this computation with the MSI value. The MFB responder includes the received MSI value in the MFSI field of the corresponding response frame. In the case of a delayed response, this use of MSI and MFSI fields allows the MFB requester to correlate the MFB with the related MRQ.

The responder may send a response frame with any of the following combinations of MFB and MFSI:

- MFB = 127, MFSI = 7: no information is provided for the immediately preceding request or for any other pending request. This combination is used when the responder is required to include an HT Control field due to other protocols that use this field (i.e., the Reverse Direction Protocol) and when no MFB is available. It has no effect on the status of any pending MRQ.
- MFB = 127, MFSI in the range 0 to 6: the responder is not now providing, and will never provide, feedback for the request that had the MSI value that matches the MFSI value.
- MFB in the range 0 to 126, MFSI in the range 0 to 6: the responder is providing feedback for the request that had the MSI value that matches the MFSI value.
- MFB in the range 0 to 126, MFSI = 7: the responder is providing unsolicited feedback.

Hardware and buffer capability may limit the number of MCS estimate computations that a MFB responder is capable of computing simultaneously. When a new MRQ is received either from a different MFB requester or from the same MFB requester with a different MSI value, and the MFB responder is not able to complete the computation for MRQ, the MFB responder may either discard the new request or may abandon an existing request and initiate an MCS estimate computation for the new MRQ.

An MFB responder that discards or abandons the computation for an MRQ should indicate this action to the MFB requester by setting the MFB to the value 127 in the next transmission of a frame addressed to the MFB requester that includes the HT Control field. The value of the MFSI is set to the MSI value of the sounding frame for which the computation was abandoned.

NOTE—The MFB requester can advertise the maximum number of spatial streams that it can transmit in its HT Capabilities element. In order to do so, the MFB requester sets the Tx MCS Set Defined bit of the Supported MCS Set field to 1 and indicates the maximum number of streams in the Tx Maximum Number Spatial Streams Supported subfield of the Supported MCS Set field. If the Tx Rx MCS Set Not Equal bit is equal to 0, the Tx MCS set is equal to the Rx MCS set, and the maximum number of transmit spatial streams is derived from the value of this field.

When computing the MCS estimate for an MFB requester whose Tx MCS Set Defined field is equal to 1, the number of spatial streams corresponding to the recommended MCS shall not exceed the limit indicated by the Tx Maximum Number Spatial Streams Supported field. The MFB responder shall not recommend an MCS corresponding to UEQM unless the MFB requester supports such modulation, as indicated by the Tx Unequal Modulation Supported bit in the Supported MCS Set field.

If the MFB is in the same PPDU as a Noncompressed Beamforming frame or a Compressed Beamforming frame, the MFB responder should estimate the recommended MCS under the assumption that the MFB requester uses the steering matrices contained therein.

After the MCS estimate computation is completed, the MFB responder should include the MFB in the MFB field in the next transmission of a frame addressed to the MFB requester that includes an HT Control field. When the MFB requester sets the MRQ subfield to 1 and sets the MSI subfield to a value that matches the MSI subfield value of a previous request for which the responder has not yet provided feedback, the responder shall discard or abandon the computation for the MRQ that corresponds to the previous use of that MSI subfield value.

A STA may respond immediately to a current request for MFB with a frame containing an MFSI field value and MFB field value that correspond to a request that precedes the current request.

NOTE 1—If an HT STA includes the HT Control field in the initial frame of an immediate response exchange and the responding HT STA includes the HT Control field in the immediate response frame, the immediate response exchange effectively permits the exchange of HT Control field elements.

NOTE 2—If an MRQ is included in the last PPDU in a TXOP and there is not enough time for a response, the recipient might transmit the response MFB in a subsequent TXOP.

NOTE 3—Bidirectional request/responses are supported. In this case, a STA acts as the MFB requester for one direction of a duplex link and a MFB responder for the other direction and transmits both MRQ and MFB in the same HT data frame.

NOTE 4—A STA that sets the MCS Feedback field to 0 in the HT Extended Capabilities field of the HT Capability elements that it transmits does not respond to an MRQ.

If a beamformer transmits a PPDU with the TXVECTOR EXPANSION_MAT_TYPE set to either COMPRESSED_SV or NON_COMPRESSED_SV, it should use the recommended MCS associated with those matrices reported in a Noncompressed Beamforming frame or a Compressed Beamforming frame.

9.29 Transmit beamforming

9.29.1 General

In order for a beamformer to calculate an appropriate steering matrix for transmit spatial processing when transmitting to a specific beamformee, the beamformer needs to have an accurate estimate of the channel over which it is transmitting. Two methods of calculation are defined as follows:

- *Implicit feedback*: When using implicit feedback, the beamformer receives long training symbols transmitted by the beamformee, which allow the MIMO channel between the beamformee and beamformer to be estimated. If the channel is reciprocal, the beamformer can use the training symbols that it receives from the beamformee to make a channel estimate suitable for computing the transmit steering matrix. Generally, calibrated radios in MIMO systems can improve reciprocity. See 9.29.2.

- *Explicit feedback*: When using explicit feedback, the beamformee makes a direct estimate of the channel from training symbols sent to the beamformee by the beamformer. The beamformee may prepare CSI or steering feedback based on an observation of these training symbols. The beamformee quantizes the feedback and sends it to the beamformer. The beamformer can use the feedback as the basis for determining transmit steering vectors. See 9.29.3.

An HT STA shall not transmit a PPDU with the TXVECTOR EXPANSION_MAT parameter present if dot11BeamFormingOptionActivated is false.

9.29.2 Transmit beamforming with implicit feedback

9.29.2.1 General

Transmit beamforming with implicit feedback can operate in a unidirectional or bidirectional manner. In unidirectional implicit transmit beamforming, only the beamformer sends beamformed transmissions. In bidirectional implicit transmit beamforming, both STAs send beamformed transmissions, i.e., a STA may act as both beamformer and beamformee.

Calibration of receive/transmit chains should be done to improve performance of transmit beamforming using implicit feedback. Over-the-air calibration is described in 9.29.2.4. For implicit transmit beamforming, only the beamformer, which is sending the beamformed transmissions, needs to be calibrated.

A STA that advertises itself as being capable of being a beamformer and/or beamformee using implicit feedback shall support the requirements in Table 9-11.

Table 9-11—STA type requirements for transmit beamforming with implicit feedback

STA capability	Required support
Beamformer	Shall set the Implicit Transmit Beamforming Capable subfield to 1 of the Transmit Beamforming Capability field of the HT Capabilities element in HT Capabilities elements that it transmits. Shall set the Implicit Transmit Beamforming Receiving Capable subfield to 1 of the Transmit Beamforming Capability field of the HT Capabilities element. Shall be capable of receiving a sounding PPDU for which the SOUNDING parameter is SOUNDING and the NUM_EXTEN_SS is equal to 0 in the RXVECTOR in the PHY-RXSTART.indication primitive, independently of the values of the Receive Staggered Sounding Capable and Receive NDP Capable subfields. Shall set the Calibration subfield to 3 of the Transmit Beamforming Capability field of the HT Capabilities element to advertise full calibration support.
Beamformee	Shall set the Implicit Transmit Beamforming Receiving Capable subfield to 1 of the Transmit Beamforming Capability field of the HT Capabilities element in HT Capabilities elements that it transmits. Shall be capable of setting the SOUNDING parameter to SOUNDING and the NUM_EXTEN_SS to 0 in the TXVECTOR in the PHY-TXSTART.request primitive when transmitting a sounding PPDU, as a response to TRQ=1, independently of the values of the Transmit Staggered Sounding Capable and Transmit NDP Capable subfields.

A STA that performs one of the roles related to transmit beamforming with implicit feedback shall support the associated capabilities shown in Table 9-12.

Table 9-12—Transmit beamforming support required with implicit feedback

Role	Required Support
Beamformee: A receiver of transmit beamformed PPDUs	Shall transmit sounding PPDUs as a response to TRQ=1.
Beamformer: A transmitter of beamformed PPDUs	Can receive sounding PPDUs. Can compute steering matrices from MIMO channel estimates obtained from long training symbols in sounding PPDUs received from the beamformee.
A responder in a calibration exchange	Can receive and transmit sounding PPDUs. Can respond with a CSI frame that contains channel measurement information obtained during reception of a sounding PPDU.
An initiator in a calibration exchange	Can receive and transmit sounding PPDUs. Can receive a CSI frame sent by a calibration responder.

When a beamformee transmits a sounding PPDU, the SOUNDING parameter in the TXVECTOR in the PHY-TXSTART.request primitive shall be set to SOUNDING. If the beamformee is capable of implicit transmit beamforming and the beamformer is capable of receiving implicit transmit beamforming, the sounding PPDU from the beamformee may be steered.

A PPDU containing one or more +HTC MPDUs in which the TRQ field is equal to 1 shall not be sent to a STA that sets the Implicit Transmit Beamforming Receiving Capable subfield of the Transmit Beamforming field of the HT Capabilities element to 0.

If a PPDU containing one or more +HTC MPDUs in which the TRQ field is equal to 1 requires an immediate response, either the response from the beamformee shall be included in a sounding PPDU, or the NDP Announcement subfield of the HT Control field shall be set to 1 and the PPDU shall be followed by an NDP. If the PPDU in which the TRQ field is equal to 1 does not require an immediate response, either the beamformee shall transmit a sounding PPDU in the next TXOP obtained by the beamformee, or the beamformee shall transmit a PPDU in the next TXOP obtained by the beamformee in which the NDP Announcement subfield of the HT Control field is set to 1 and that PPDU shall be followed by an NDP. The use of NDP as a sounding PPDU is described in 9.31.

NOTE—A STA that acts as a beamformer using implicit feedback expects to receive a sounding PPDU in response to a training request. The STA can compute steering matrices from the channel estimates obtained from the received sounding PPDU.

At the end of the TXOP, the final PPDU from the beamformer shall not have the TRQ field set to 1 in a frame that requests an immediate response if there is not enough time left in the TXOP for the beamformee to transmit the longest valid sounding PPDU with its response.

9.29.2.2 Unidirectional implicit transmit beamforming

Figure 9-35 shows an example of a PPDU exchange used in unidirectional implicit transmit beamforming using the Clause 20 PHY. In this example, sounding PPDUs are used that carry MPDUs (i.e., an example of implicit beamforming using NDPs is not shown here.) STA A is the beamformer that initiates the PPDU exchange, and STA B is the beamformee.

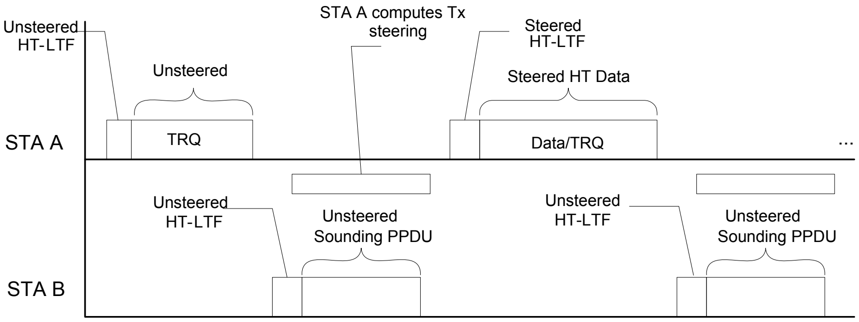


Figure 9-35—Example PPDU exchange for unidirectional implicit transmit beamforming

The PPDU exchange can be summarized as follows:

a) STA A initiates the frame exchange sequence by sending an unsteered PPDU to STA B. The PPDU includes a training request (TRQ= 1) in a +HTC MPDU.

b) STA B sends a sounding PPDU in response to the training request from STA A.

c) On receiving the sounding PPDU, STA A uses the resulting channel estimate to compute steering matrices and uses these matrices to send a steered PPDU back to STA B.

d) The steered PPDU transmitted in step c) and subsequent steered PPDUs transmitted by STA A may include training requests (TRQ=1) in a +HTC MPDU. In response to each training request, STA B returns a sounding PPDU to STA A, which enables STA A to update its steering vectors. If the steering vectors resulting from step c) or subsequent sounding PPDUs are deemed stale due to delay, the sequence may be restarted by returning to step a).

Step d) in the above PPDU exchange represents steady-state unidirectional transmit beamforming operation.

During the PPDU exchange, neither the receiving nor the transmitting STA should switch antennas.

9.29.2.3 Bidirectional implicit transmit beamforming

Figure 9-36 shows an example of a PPDU exchange used in bidirectional implicit transmit beamforming, using the Clause 20 PHY. In this example, sounding PPDUs are used that carry MPDUs. STA A initiates the frame exchange, and STA A and STA B alternate in the roles of beamformer and beamformee.

The PPDU exchange can be summarized as follows:

a) STA A initiates the frame exchange sequence by sending an unsteered PPDU to STA B. The PPDU includes a training request (TRQ= 1) in a +HTC MPDU.

b) STA B sends a sounding PPDU in response to the training request. In addition, this PPDU includes a training request in a +HTC MPDU to enable implicit transmit beamforming in the RD.

c) On receiving the sounding PPDU, STA A uses the resulting channel estimate to compute steering matrices and uses these matrices to send a steered PPDU back to STA B. This steered PPDU is also a sounding PPDU in response to the training request from STA B.

NOTE—Steering matrices with nonorthonormal columns should not be used in transmitting sounding PPDUs for implicit feedback. In general, bidirectional implicit beamforming will not function as described here when the steering matrices have nonorthonormal columns. See 20.3.12.2.

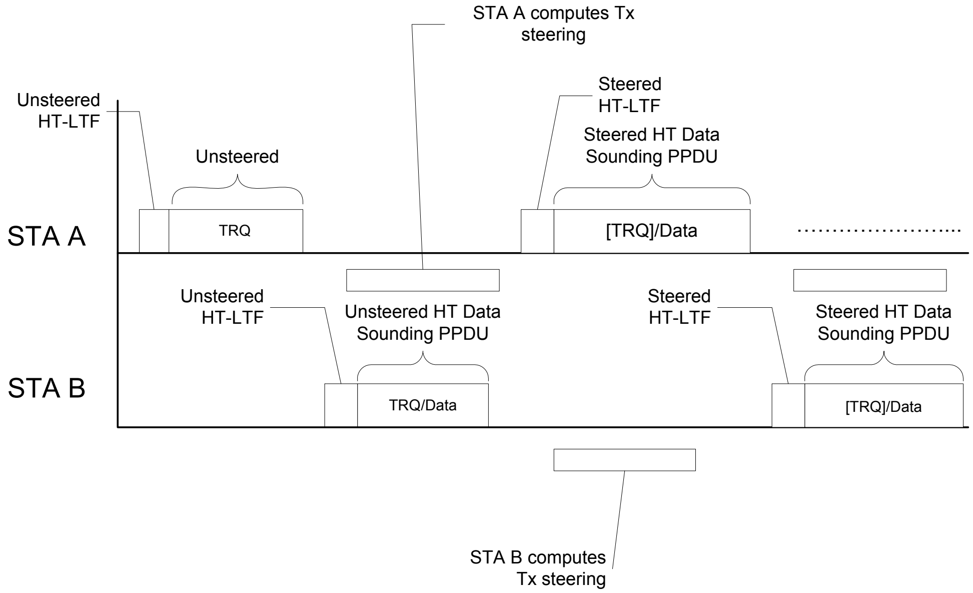

STA A computes Tx steering
Unsteered HT-LTF
Unsteered
Steered HT-LTF
Steered HT Data Sounding PPDU
STA A
TRQ
[TRQ]/Data
……………………..
Unsteered HT-LTF
Unsteered HT Data Sounding PPDU
Steered HT-LTF
Steered HT Data Sounding PPDU
STA B
TRQ/Data
[TRQ]/Data
STA B computes Tx steering

Figure 9-36—Example PPDU exchange for bidirectional implicit transmit beamforming

d) On receiving the sounding PPDU, STA B uses the resulting channel estimate to compute steering matrices and uses these matrices to send a steered PPDU back to STA A. The steered PPDU transmitted in step c) and subsequent steered PPDUs transmitted by STA A may include training requests in HTC. In response to each training request, STA B returns a sounding PPDU to STA A, which enables STA A to update its steering vectors. If the steering vectors resulting from step c) or subsequent sounding PPDUs are deemed stale due to delay, the sequence may be restarted by returning to step a).

e) The steered PPDU transmitted in step d) and subsequent steered PPDUs transmitted by STA B may include training requests in HTC. In response to each training request, STA A returns a sounding PPDU to STA B, which enables STA B to update its steering vectors. If the steering vectors resulting from step d) or subsequent sounding PPDUs are deemed stale due to delay, the sequence may be restarted by returning to step a).

Steps d) and e) in the above PPDU exchange represent steady-state bidirectional transmit beamforming operation.

During the PPDU exchange, neither the receiving nor the transmitting STA should switch antennas.

NOTE—The TRQ protocol used with the beamforming training process is not sufficient to permit STA B to transmit data frames in the RD. In the example shown in Figure 9-36, STA A would additionally have to follow the rules of the Reverse Direction Protocol (see 9.25).

9.29.2.4 Calibration

9.29.2.4.1 Introduction

Differences between transmit and receive chains in a STA degrade the inherent reciprocity of the over-the-air time division duplex channel and cause degradation of the performance of implicit beamforming. Calibration acts to remove or reduce differences between transmit and receive chains and enforce reciprocity in the observed baseband-to-baseband channels between two STAs.

A STA acting as a beamformer should be calibrated to maximize performance. A STA acting only as a beamformee does not need to be calibrated. If calibration is desired, it is performed using the over-the-air calibration procedure described below.

The calibration procedure involves the computation of correction matrices that cause the observed channel matrices in the two directions of the link to be transposes of each other and thus renders the resultant channel reciprocal. See 20.3.12.2 for a more detailed description. If it is able to do so, a STA should calibrate upon association.

NOTE—STAs with two or more transmit RF chains should be calibrated in order to engage in implicit transmit beamforming. STAs with any number of RF chains, including those with a single RF chain, can participate in a calibration exchange as a calibration responder.

9.29.2.4.2 Calibration capabilities

A STA that sets the Implicit Transmit Beamforming Capable subfield of the Transmit Beamforming Capabilities field to 1 shall support calibration and shall set the Calibration subfield of the Transmit Beamforming Capabilities field to 3 (indicating full support of calibration) in HT Capabilities elements that it transmits. A STA that does not set the Implicit Transmit Beamforming Capable subfield of the Transmit Beamforming Capabilities field to 1 may support calibration and shall set the Calibration subfield of the Transmit Beamforming Capabilities field to the value that indicates its calibration capability in the Transmit Beamforming Capabilities field in HT Capabilities elements that it transmits (see Table 8-128), when the Transmit Beamforming Capabilities field exists.

A STA that is capable of initiating calibration (the Calibration subfield of the Transmit Beamforming Capabilities field is equal to 3) shall set the CSI Max Number of Rows Beamformer Supported subfield to an appropriate value, even if the STA sets the Explicit Transmit Beamforming CSI Feedback subfield to the value 0.

In order to support calibration, a STA that advertises that it is capable of responding to a calibration request shall be capable of transmitting a CSI frame in which the value of the Grouping subfield of the MIMO Control field is 0 (no grouping) and the Coefficients Size subfield of the MIMO Control field is 3 ($Nb = 8$ bits) in response to a CSI feedback request indicated by the CSI/Steering subfield of the HT Control field equal to 1 and the Calibration Position subfield of the HT Control field equal to 3, independently of the advertised values of the Explicit Transmit Beamforming CSI Feedback subfield in the Transmit Beamforming Capabilities field in the HT Capabilities element. A STA that advertises that it is capable of initiating a calibration request shall be capable of receiving a CSI frame in which the value of the Grouping subfield of the MIMO Control field is equal to 0 (no grouping) and the Coefficients Size subfield of the MIMO Control field is equal to 3 ($Nb = 8$ bits) as a response to CSI feedback request indicated by the CSI/Steering subfield of the HT Control field equal to 1 with the Calibration Position subfield equal to 3, independently of the advertised values of the Explicit CSI Transmit Beamforming Capable subfield in the Transmit Beamforming Capabilities field in the HT Capabilities element.

A STA may initiate a calibration training frame exchange sequence with another STA if that STA supports calibration. A STA shall not initiate a calibration training frame exchange with another STA if that STA does not support calibration.

If the Receive NDP Capable field is equal to 1, the value of the Calibration field is equal to 1 or 3, and the device supports transmitting sounding PPDUs for which two or more channel dimensions can be estimated (two or more columns of the MIMO channel matrix), then transmission of NDPs shall be supported (and the Transmit NDP Capable bit shall be set to 1).

9.29.2.4.3 Sounding exchange for calibration

Figure 9-37 illustrates the calibration PPDU exchange using sounding PPDUs that contain an MPDU.

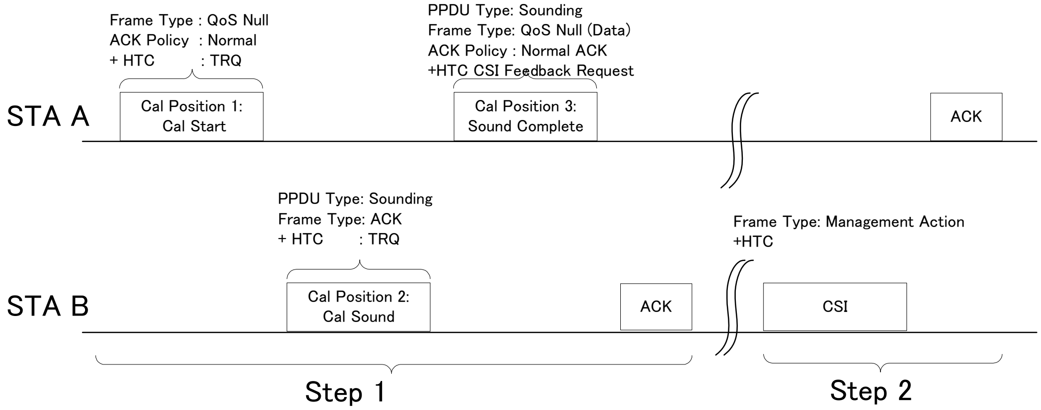


Figure 9-37—Calibration procedure with sounding PPDU containing an MPDU

The calibration procedure begins with a calibration sounding PPDU exchange sequence shown as Step 1 in Figure 9-37. The Calibration Sequence subfield in the HT Control field shall be incremented each time a new calibration procedure is started.

STA A (the calibration initiator) shall transmit a Calibration Start frame (Calibration Position subfield set to 1) with the TRQ field in the HT Control field set to 1. This frame initiates a calibration procedure. It shall be a QoS Null data frame with the ACK Policy field set to Normal ACK.

In response, STA B (the calibration responder) shall transmit a Calibration Sounding Response frame (Calibration Position subfield set to 2), a SIFS interval after the end of the Calibration Start frame, using a sounding PPDU. This step allows STA A to estimate the MIMO channel from STA B to STA A. In the Calibration Sounding Response frame, the Calibration Sequence subfield in HT Control field shall be set to the same value that is indicated in the Calibration Start frame. The Calibration Sounding Response frame shall have a frame type of ACK+HTC, and the TRQ field in the HT Control field in this frame shall be set to 1.

In response, STA A shall transmit a Calibration Sounding Complete frame (Calibration Position subfield set to 3) that contains the CSI/Steering subfield of the HT Control field set to 1, a SIFS interval after the end of the Calibration Sounding Response frame, using a sounding PPDU. This step allows STA B to estimate the MIMO channel from STA A to STA B. In this Calibration Sounding Complete frame, the Calibration Sequence subfield in the HT Control field shall be set to the same value that is indicated in the Calibration Sounding Response frame. The Calibration Sounding Complete frame shall be a QoS Null+HTC with the ACK Policy field set to Normal ACK.

A frame in which the Calibration Position subfield is equal to 2 or 3 shall be transmitted in a sounding PPDU (a PPDU for which the SOUNDING parameter is set to SOUNDING). The number of Long Training fields (LTFs) used to obtain MIMO channel estimation that are sent in the sounding PPDU shall be determined by the number of transmit chains (N_{TX}) used in sending these LTFs at the STA transmitting the sounding PPDU. The transmit chains used at the calibration initiator are those for which calibration is required.

The calibration responder may train up to maximum available transmit chains to maximize the quality of the resulting calibration, although the number of space-time streams for data symbols shall be determined by the rule described in 9.7.

When transmitting a sounding PPDU during Step 1 of a calibration procedure, if the Receive Staggered Capability subfield in the Transmit Beamforming Capability field of the HT Capabilities element transmitted by the intended receiver is 0, then,

— If the sounding PPDU is not an NDP, the number of antennas used by the sender shall be less than or equal to the maximum number of space-time streams indicated by the Rx MCS Bitmask subfield of the Supported MCS Set field and the Rx STBC subfield of the HT Capabilities element transmitted by the intended receiver.

— If the sounding PPDU is an NDP, the number of antennas used by the sender shall be less than or equal to the number indicated by the Channel Estimation Capability subfield in the Transmit Beamforming Capability field of the HT Capabilities element transmitted by the intended receiver.

Sounding packets in which the Calibration Position subfield is equal to 2 or 3 shall use the spatial mapping matrices defined in 20.3.12.3. The calibration responder shall not remove the spatial mapping from the CSI to be fed back to the initiator of the frame exchange.

NOTE—The calibration initiator of this frame exchange is responsible for accounting for the spatial mapping in both its local channel estimate as well as in the quantized CSI fed back to it.

The row order in the CSI feedback matrix transmitted from STA B shall correspond to the association of the rows of the spatial mapping matrix (see Equation (20-75)) to its transmit antennas. For example, the receive antenna at STA B associated with row *i* in the CSI feedback matrix in each subcarrier is the same as its transmit antenna associated with row *i* in the spatial mapping matrix used for transmitting the sounding response with Calibration Position equal to 2.

Figure 9-38 and Figure 9-39 illustrate the calibration PPDU exchange using NDPs.

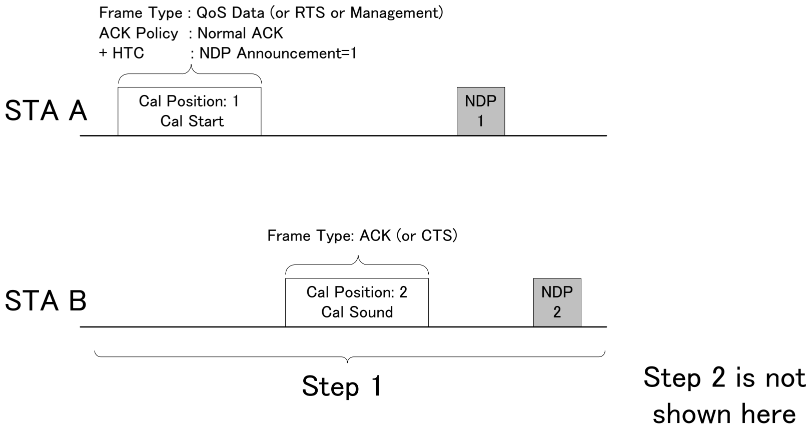


Figure 9-38—Calibration procedure with NDP

The calibration procedure begins with a calibration sounding PPDU exchange sequence, shown as Step 1 in Figure 9-38 and Figure 9-39, when STA B supports transmitting sounding PPDUs for which only one channel dimension can be estimated (i.e., a single column of the MIMO channel matrix). The Calibration Sequence subfield in the HT Control field shall be incremented each time a new calibration procedure is started.

NDP transmission within a calibration procedure follows the rules defined in 9.31.1. STA A transmits a Calibration Start frame (i.e., with the Calibration Position subfield set to 1) with the NDP Announcement subfield set to 1 and CSI/Steering subfield of the HT Control field set to 1. Only the current TXOP holder may set both the Calibration Position and NDP Announcement subfields to 1. This frame initiates a calibration procedure.

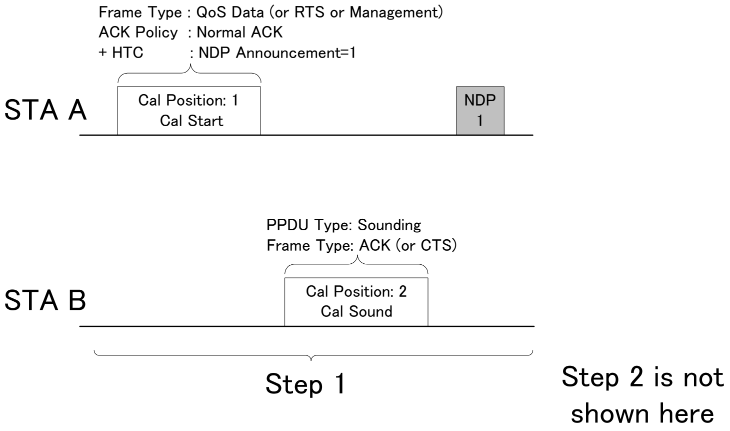


Figure 9-39—Calibration procedure with NDP when STA B supports transmitting sounding PPDUs for which only one channel dimension can be estimated (i.e., a single column of the MIMO channel matrix)

STA B shall transmit a Calibration Sounding Response frame (i.e., with the Calibration Position subfield set to 2) after a SIFS interval after the received Calibration Start frame. If STA B supports transmitting sounding PPDUs for which only one channel dimension can be estimated (i.e., a single column of the MIMO channel matrix), this Calibration Sounding Response frame is sent with the SOUNDING parameter of the TXVECTOR set to SOUNDING (see Figure 9-39).

As determined by NDP rules a) or b) in 9.31.1, STA A sends the first NDP as a sounding PPDU a SIFS after receiving the Calibration Sounding Response frame. This step allows STA B to estimate the MIMO channel from STA A to STA B. In the Calibration Sounding Response frame, the Calibration Sequence subfield in HT Control field shall be set to the same value that is contained in the Calibration Start frame. The Calibration Sounding Response frame shall contain an HT Control field, and the type and subtype of the frame are determined by the Calibration Start frame.

As determined by NDP rule d), STA B might transmit a second NDP as a sounding PPDU a SIFS interval after receiving the first NDP. This second NDP allows STA A to estimate the channel from STA B to STA A.

NOTE—STA B does not transmit an NDP when it supports transmitting sounding PPDUs for which only one channel dimension can be estimated (see Figure 9-39).

Otherwise (i.e., if STA B supports transmitting sounding PPDUs for which only one channel dimension can be estimated (single column of the MIMO channel matrix)), the transmission of the sounding PPDU in Calibration Position 2 allows STA A to estimate the channel from STA B to STA A.

NDP sounding PPDUs shall use the spatial mapping matrices defined in 20.3.13.3. The calibration responder shall not remove the spatial mapping from the CSI to be fed back to the initiator of the frame exchange.

NOTE—The calibration initiator of this frame exchange is responsible for accounting for the spatial mapping in both its local channel estimate as well as in the quantized CSI fed back to it.

9.29.2.4.4 CSI reporting for calibration

The remaining message exchange in the calibration procedure is not time critical.

The calibration initiator should not transmit any frames that are part of the calibration sequence shown in Step 1 in Figure 9-37 if either of the response frames from the calibration responder (the frames shown as Calibration Position 2 and ACK in Step 1) is not received correctly within an ACKTimeout interval (as defined in 9.3.2.8) after the PHY-TXEND.confirm primitive. If the calibration initiator aborts the calibration sequence, it may restart the calibration sequence with a value of the Calibration Sequence subfield in the Calibration Control subfield of the HT Control field that is different (i.e., incremented) from the value used in the aborted sequence. Within a non-NDP calibration sequence, the calibration responder should not transmit any further frames that are part of the calibration sequence shown in Step 1 if the frame having Calibration Position 3 is not received correctly within an ACKTimeout interval (as defined in 9.3.2.8) after the PHY-TXEND.confirm primitive.

When the MIMO channel measurements become available at STA B, STA B shall send one or more CSI frames that contain the CSI report (Step 2 in Figure 9-37). This CSI report shall have full precision, i.e, *Ng*=1 (no grouping) and *Nb*=3 (8 bits). In these CSI frames, the Calibration Sequence subfields in HT Control fields shall be set to the same value that is indicated in the Calibration Sounding Complete frame. These CSI frames shall have a frame type of Management Action +HTC.

STA B should finish transmission of the first CSI frame within aMaxCSIMatricesReportDelay (in milliseconds) after the reception of the frame containing the CSI feedback request or NDP announcement.

NOTE—If necessary, the CSI Report field can be split into up to 8 segments as specified in Table 8-42.

A STA that has started but not completed the calibration procedure and that receives some other request that requires the buffering of CSI (such as another calibration initiation frame, MFB request, CSI feedback request for link adaptation, or feedback request for explicit Transmit Beamforming) may ignore the other request.

From the beginning of Step 1 of the calibration procedure and continuing through the end of Step 2 of the calibration procedure, neither the receiving nor the transmitting STA should switch antennas.

9.29.3 Explicit feedback beamforming

In this subclause, the terms *beamformer* and *beamformee* refer to STAs that are involved in explicit feedback beamforming.

A beamformer uses the feedback response that it receives from the beamformee to calculate a beamforming feedback matrix for transmit beamforming. This feedback response may have one of three formats:

- *CSI*: The beamformee sends the MIMO channel coefficients to the beamformer.
- *Noncompressed beamforming*: The beamformee sends calculated beamforming feedback matrices to the beamformer.
- *Compressed beamforming*: The beamformee sends compressed beamforming feedback matrices to the beamformer.

The supported formats shall be advertised in the beamformee's HT Capabilities element.

NOTE—A beamformer might discard the feedback response if the TSF time when the PHY_CCA.indication(IDLE) primitive corresponding to the feedback response frame's arrival minus the value from the Sounding Timestamp field in the feedback response frame is greater than the coherence time interval of the propagation channel.

A beamformee's responding capabilities shall be advertised in HT Capabilities elements contained in Beacon, Probe Request, Probe Response, Association Request, Association Response, Action, and Action No Ack frames that are transmitted by the beamformee. Devices that are capable of acting as a beamformee shall advertise one of the following response capabilities in the Explicit Transmit Beamforming CSI Feedback subfield of the Transmit Beamforming Capability field:

- *Immediate*: The beamformee is capable of sending a feedback response a SIFS after receiving a sounding PPDU and/or is capable of sending a feedback response aggregated in a PPDU that contains a MAC response within the beamformer's TXOP.
- *Delayed*: The beamformee is not capable of sending the feedback response within the beamformer's TXOP, but it is capable of sending the feedback response in a TXOP that it obtains.

— *Immediate and Delayed*: The beamformee is capable of sending a feedback response a SIFS after receiving a sounding PPDU, sending a feedback response aggregated in a PPDU that contains a MAC response within the beamformer's TXOP, or sending the feedback response in a TXOP that it obtains.

The sounding frame types supported by the beamformee, staggered and/or NDP, are advertised in the HT Capabilities element in frames that are transmitted by the beamformee.

A STA that sets any of the Explicit Transmit Beamforming CSI Feedback Capable, Explicit Noncompressed Beamforming Feedback Capable, or Explicit Compressed Beamforming Feedback Capable fields to 1 shall transmit explicit feedback based on the receipt of a +HTC sounding PPDU in which the CSI/Steering subfield has a nonzero value and that does not contain Extension HT-LTFs (HT-ELTFs). This requirement is independent of the values of the Receive Staggered Sounding Capable and the Receive NDP Capable fields.

A beamformer shall set the SOUNDING parameter of the TXVECTOR to SOUNDING in the PHY-TXSTART.request primitive corresponding to each packet that is used for sounding.

A beamformer shall set the response type format indicated in the CSI/Steering subfield of the HT Control field of any sounding frame excluding the NDP and of any PPDU with the NDP Sounding Announcement field equal to 1 to one of the nonzero values (CSI, Compressed Beamforming, or Noncompressed Beamforming) that corresponds to a type that is supported by the beamformee.

The receipt of a PHY-RXSTART.indication primitive with the RXVECTOR SOUNDING parameter value equal to SOUNDING indicates a sounding packet. A non-NDP request for feedback is a sounding PPDU with a +HTC MPDU that contains a nonzero value of the CSI/Steering subfield and that has the NDP Announcement subfield equal to 0.

An NDP request for feedback is the combination of a +HTC MPDU that contains a nonzero value of the CSI/Steering subfield and that has the NDP Announcement subfield equal to 1 and the NDP that follows.

A beamformee that transmits a feedback frame in response to a sounding PPDU sent by a beamformer shall transmit a frame of the type (CSI, Compressed Beamforming, or Noncompressed Beamforming) indicated in the CSI/Steering subfield of the HT Control field transmitted by the beamformer.

The Beamformee decides on any tone grouping to be used in the explicit Beamforming feedback. The value selected shall be a value supported by the Beamformer as indicated in the Minimal Grouping subfield of the Beamformer's HT Capabilities element.

A beamformee that sets the Explicit Transmit Beamforming CSI Feedback field of its HT Capabilities element to either 2 or 3 shall transmit Explicit CSI feedback after SIFS or later in the beamformer's TXOP as a response to a non-NDP request for feedback in a frame that is appropriate for the current frame exchange sequence following Table 9-13. A beamformee that sets the Explicit Noncompressed Beamforming Feedback Capable field of its HT Capabilities element to either 2 or 3 shall transmit Explicit Noncompressed Beamforming feedback after SIFS or later in the beamformer's TXOP as a response to a non-NDP request for feedback in a frame that is appropriate for the current frame exchange sequence following Table 9-13.

A beamformee that sets the Explicit Compressed Feedback Capable field of its HT Capabilities element to either 2 or 3 shall transmit Explicit Compressed Beamforming feedback after SIFS or later in the beamformer's TXOP as a response to a non-NDP request for feedback in a frame that is appropriate for the current frame exchange sequence following Table 9-13.

Table 9-13—Rules for beamformee immediate feedback transmission responding to non-NDP sounding

Type of response	Rule
CTS response	If the transmission of a CTS is required in response to the non-NDP request for feedback, the transmission of the feedback response frame shall be delayed until the beamformee's next transmission within the TXOP. This feedback response frame may be aggregated in an A-MPDU with an ACK or BlockAck frame.
Acknowledgment response	If the transmission of an ACK or BlockAck control response frame is required in response to the non-NDP request for feedback, both the feedback response frame and the control response frame may be aggregated in an A-MPDU.
No control response	If the transmission of a control response frame is not required in response to the non-NDP request for feedback, the feedback response frame shall be sent a SIFS after the reception of the sounding PPDU containing the request for feedback.
Later aggregation of feedback and acknowledgment	If the immediate-feedback-capable beamformee cannot transmit an aggregated or immediate CSI/Steering response in a SIFS time after the end of the received sounding packet and the beamformee is subsequently required to transmit an ACK or BlockAck response in the same TXOP, it may transmit the feedback response in an A-MPDU with the ACK or BlockAck frame.

A beamformee that sets the Explicit Transmit Beamforming CSI Feedback field of its HT Capabilities element to either 2 or 3 shall transmit the Explicit CSI feedback after SIFS or later in the beamformer's TXOP as a response to an NDP request for feedback in a frame that is appropriate for the current frame exchange sequence following Table 9-14.

Table 9-14—Rules for beamformee immediate feedback transmission responding to NDP sounding

Type of response	Rule
Control response	If the transmission of a control response frame is required in response to the NDP request for feedback, the control response frame is transmitted a SIFS after reception of the PPDU that elicited the control response, and the feedback response frame may be transmitted a SIFS after the reception of the NDP. — If the feedback response frame is not transmitted a SIFS after the reception of the NDP and the beamformee is subsequently required to transmit an ACK or BlockAck response in the same TXOP, then the feedback response may be aggregated with the ACK or BlockAck frame. — If the feedback response frame is not transmitted a SIFS after the reception of the NDP and is not transmitted as part of an aggregated ACK or BlockAck response in the same TXOP, then the feedback response frame is delayed until the beamformee's next transmission within the TXOP.
No control response	If the transmission of a control response frame is not required in response to the NDP request for feedback, the feedback response frame may be sent a SIFS after the reception of the NDP. — If the feedback response frame is not transmitted a SIFS after the reception of the NDP and the beamformee is subsequently required to transmit an ACK or BlockAck response in the same TXOP, then the feedback response may be aggregated with the ACK or BlockAck frame. — If the feedback response frame is not transmitted a SIFS after the reception of the NDP and is not transmitted as part of an aggregated ACK or BlockAck response in the same TXOP, then the feedback response frame is delayed until the beamformee's next transmission within the TXOP.

A beamformee that sets the Explicit Noncompressed Beamforming Feedback Capable field of its HT Capabilities element to either 2 or 3 shall transmit the Explicit Noncompressed Beamforming feedback after SIFS or later in the beamformer's TXOP as a response to an NDP request for feedback in a frame that is appropriate for the current frame exchange sequence following Table 9-14.

A beamformee that sets the Explicit Compressed Beamforming Feedback Capable field of its HT Capabilities element to either 2 or 3 shall transmit the Explicit Compressed Beamforming feedback after SIFS or later in the beamformer's TXOP as a response to an NDP request for feedback in a frame that is appropriate for the current frame exchange sequence following Table 9-14.

When the beamformee sets the Explicit Transmit Beamforming CSI Feedback field of its HT Capabilities element to either 2 or 3 and the beamformer has transmitted an NDP or an non-NDP Explicit Beamforming CSI feedback request in a frame that does not require immediate control response, the beamformer shall not transmit the next packet to the beamformee until PIFS after transmitting the sounding request.

When the beamformee sets the Explicit Noncompressed Beamforming Feedback Capable field of its HT Capabilities element to either 2 or 3 and the beamformer has transmitted an NDP or an non-NDP Explicit Noncompressed Beamforming feedback request in a frame that does not require immediate control response, the beamformer shall not transmit the next packet to the beamformee until PIFS after the sounding request.

When the beamformee sets the Explicit Compressed Beamforming Feedback Capable field of its HT Capabilities element to either 2 or 3 and the beamformer has transmitted an NDP or an non-NDP Explicit Compressed Beamforming feedback request in a frame that does not require immediate control response, the beamformer shall not transmit the next packet to the beamformee until PIFS after transmitting the sounding request.

A beamformee shall not transmit a CSI, Compressed Beamforming, or Noncompressed Beamforming frame except in response to a request for feedback.

NOTE—Error recovery in a TXOP is not affected by sounding. A beamformer that is a TXOP holder and that fails to receive an expected response to a sounding PPDU can continue transmission as specified in 9.19.2.4.

A beamformee transmitting a feedback response after SIFS or later in the beamformer's TXOP shall use an Action No Ack frame or an Action No Ack +HTC frame (defined in 8.3.3.14).

A beamformee transmitting delayed feedback response shall use an Action frame or an Action +HTC frame to send this information within a separate TXOP.

If necessary, the CSI Report field, Noncompressed Beamforming Report field, or Compressed Beamforming Report field may be split into up to 8 frames. The length of each segment shall be equal number of octets for all segments except the last, which may be smaller.

NOTE—A STA that has been granted an RDG can act as a beamformer during the RDG time period, provided that the RD rules are obeyed.

A beamformee that advertises itself as delayed-feedback-capable shall not transmit an immediate feedback response unless it also advertises itself as immediate-feedback-capable.

A beamformer may use the following worst-case parameters to estimate the duration of the expected frame that contains the feedback response: Basic MCS, HT-Mixed Format, Supported Grouping.

An Explicit Feedback Request may be combined with an MRQ. If the response contains a beamforming feedback matrix, the returned MCS shall be derived from the same information that was used to generate this particular beamforming feedback matrix. If the response contains channel coefficients, the returned MCS shall be derived from an analysis of the sounding frame that was used to generate the channel coefficients. The MFB field set to 127 (meaning no feedback) may be used when the beamformee is unable to generate the MCS in time for inclusion in the response transmission frame. A CSI-capable STA may be incapable of generating MFB.

Explicit feedback shall be calculated only from a sounding PPDU.

9.30 Antenna selection (ASEL)

9.30.1 Introduction

ASEL is a time-variant mapping of the signals at the RF chains onto a set of antenna elements when the number of RF chains is smaller than the number of antenna elements at a STA and/or AP. The mapping might be chosen based on instantaneous or averaged CSI. ASEL requires the training of the full size channel associated with all antenna elements, which is obtained by transmitting or receiving sounding PPDUs over all antennas. These sounding PPDUs should be sent within a single TXOP. To avoid channel distortions, these sounding PPDUs shall be transmitted consecutively. The training information is exchanged using the HT Control field. When both transmitter and receiver have ASEL capabilities, training of transmit and receive antennas might be done one after another. ASEL supports up to eight antennas and up to four RF chains.

9.30.2 Procedure

A STA shall not initiate an ASEL training frame exchange sequence with another STA unless that STA supports ASEL, as determined by the ASEL Capability field (see 8.4.2.58.7).

A STA that is capable of supporting ASEL should set each subfield of the ASEL Capability field of the HT Capabilities element to 1 depending on its capabilities in HT Capabilities elements that it transmits (see 8.4.2.58.7).

A STA that sets the Explicit CSI Feedback Based Tx ASEL Capable subfield of the ASEL Capability field (see 8.4.2.58.7) to 1 shall set the CSI Max Number of Rows Beamformer Supported subfield of the Transmit Beamforming Capabilities field to an appropriate value, even if the STA sets the Explicit Transmit Beamforming CSI Feedback subfield to the value 0.

The frame exchange sequence for transmit ASEL is shown in Figure 9-40, where the term *ASEL transmitter* identifies the STA that is conducting transmit ASEL, and the term *transmit ASEL responder* identifies the STA that provides ASEL feedback. The frame exchange comprises the following steps:

a) (Optional) A transmit ASEL responder may initiate the transmit ASEL training by sending a +HTC frame with the ASEL Command subfield set to Transmit Antenna Selection Sounding Request (TXASSR).

b) The ASEL transmitter sends out consecutive sounding PPDUs separated by SIFS in a TXOP of which it is the TXOP holder with no ACK over different antenna sets, each PPDU containing a +HTC frame with the ASEL Command subfield set to Transmit Antenna Selection Sounding Indication (TXASSI or TXASSI-CSI). Each sounding PPDU is transmitted over one antenna set.

   If the ASEL transmitter allows antenna indices feedback (by setting the ASEL Command subfield to TXASSI), the antenna sets from which the sounding PPDUs are transmitted shall be disjoint. If the ASEL transmitter uses NDP sounding PPDUs for the ASEL sounding, the spatial mapping matrix Q_k shall be set equal to the identity matrix starting with the first NDP. If the ASEL transmitter uses non-NDP sounding PPDUs for the ASEL sounding, the spatial mapping matrix Q_k shall be an FFT matrix. An FFT matrix of size *N* is defined as a square matrix of dimension *NxN* with entries w_{im} ,

   i=*0,..N–1, m=0,..N–1* where $w_{im} = \frac{1}{\sqrt{N}} \times \exp\left(-j2\pi\frac{im}{N}\right)$.

The ASEL transmitter shall not include TXASSI-CSI in the command field of the sounding frame if the last received value of the Explicit CSI Feedback Capable subfield of the ASEL Capability field (see 8.4.2.58.7) from the receiver was 0.

NOTE—For example, in the case of sounding over all disjointed antenna sets, the number of consecutive sounding PPDUs equals the smallest integer that is greater than or equal to the number of antennas divided by the number of RF chains. These sounding PPDUs need to be sent within a single TXOP in order to minimize the change in the channel during this procedure.

c) The transmit ASEL responder estimates the subchannel corresponding to each sounding PPDU.

d) If the ASEL Command subfield in the sounding frames is equal to TXASSI-CSI, after receiving all the sounding PPDUs, the transmit ASEL responder shall respond with the full size CSI in a subsequent TXOP. If the ASEL Command subfield in the sounding frames is equal to TXASSI, after receiving all the sounding PPDUs, the transmit ASEL responder may either respond with the full size CSI in a subsequent TXOP, or conduct ASEL computation and provide the selected antenna indices in a subsequent TXOP.

   1) CSI is transported using the MIMO CSI Matrices frame defined in 8.5.12.6 contained within either an Action No Ack or Action frame. Multiple CSI frames may be required to provide the complete feedback, in which case the value of the Sounding Timestamp field within each of these CSI frames shall correspond to the arrival time of the sounding frame that was used to generate the feedback information contained in the frame.

   2) Antenna indices feedback is carried in the Antenna Selection Indices Feedback frame, defined in 8.5.12.9. One octet of the Antenna Selection Indices field is used to carry the selected antenna indices feedback.

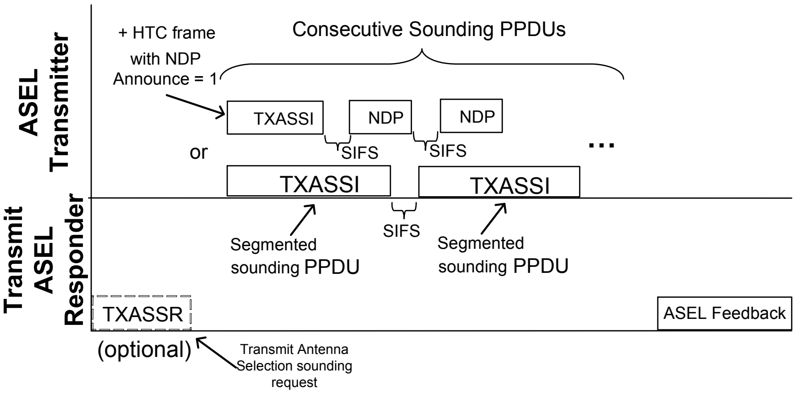


Figure 9-40—Transmit ASEL

If the transmit ASEL responder does not correctly receive all the sounding PPDUs but has correctly received at least one of the preceding sounding PPDUs, it shall send a +HTC frame with the MAI subfield set to the value ASELI (see Table 8-7), the ASEL Command subfield set to No Feedback Due to ASEL Training Failure or Stale Feedback to indicate the failure of the ASEL training process, and the ASEL Data subfield set to either of the following:

— The integer value corresponding to the number in sequence of the first sounding PPDU that was not properly received, where 0 corresponds to the first sounding PPDU in the ASEL training sequence or

— The value 0

A transmit ASEL responder that determines that the ASEL feedback is stale shall notify the ASEL transmitter by transmitting a +HTC MPDU with the MAI subfield set to ASELI, the ASEL Command subfield set to No Feedback Due to ASEL Training Failure or Stale Feedback, and the ASEL Data subfield set to 0.

If, in response to the transmission of a sequence of sounding PPDUs, the ASEL transmitter receives a +HTC MPDU with the MAI subfield equal to ASELI, the ASEL Command subfield equal to No Feedback Due to ASEL Training Failure or Stale Feedback, and the ASEL Data subfield equal to a nonzero value to indicate a failed ASEL training frame sequence, the ASEL transmitter may perform any of the following actions:

- Do nothing.
- Restart the failed ASEL training frame sequence from the point of failure by transmitting a +HTC MPDU with the MAI subfield set to ASELI, the ASEL Command subfield set to TXASSR, and the ASEL Data subfield set to a nonzero value to correspond to the command Transmit Antenna Selection Sounding Resumption (a Resumption MPDU), where the ASEL Data subfield value is the order number (from the original training frame sequence) of the first sounding PPDU transmitted in the restarted ASEL training frame sequence and where 0 corresponds to the first sounding PPDU in the original ASEL training sequence.
- Execute a new ASEL training frame sequence by transmitting a +HTC MPDU with the MAI subfield set to ASELI, the ASEL Command subfield set to TXASSI or TXASSI-CSI, and the ASEL Data subfield set to a nonzero value.

If a transmit ASEL responder receives a +HTC MPDU with the MAI subfield equal to ASELI, the ASEL Command subfield equal to TXASSR, and the ASEL Data subfield equal to a nonzero value to correspond to the command Transmit Antenna Selection Sounding Resumption (a Resumption MPDU), the transmit ASEL responder shall respond to the training sequence according to the original command value (i.e., TXASSI or TXASSI-CSI) and shall assume a total number of sounding PPDUs that corresponds to the number of sounding PPDUs from the original command frame. The number of sounding frames that follow the Resumption MPDU is equal to the number of sounding PPDUs from the original command frame minus the order number transmitted in the ASEL Data subfield of the Resumption MPDU.

The frame exchange sequence for receive ASEL is shown in Figure 9-41, where the term *ASEL receiver* identifies the STA that is conducting receive ASEL, and the term *ASEL sounding-capable transmitter* identifies the STA sending the consecutive sounding PPDUs used for receive ASEL calculations. The frame exchange comprises the following steps:

- The ASEL receiver transmits a +HTC frame with the MAI subfield set to ASELI, the ASEL Command subfield set to Receive Antenna Selection Sounding Request (RXASSR), and the ASEL Data subfield set to the number of sounding PPDUs required.

  NOTE— For example, in the case of sounding over all disjointed antenna sets, the number of total consecutive sounding PPDUs or NDPs equals the smallest integer that is greater than or equal to the number of antennas divided by the number of RF chains.

- The ASEL sounding-capable transmitter responds with the corresponding number of sounding PPDUs in its subsequent TXOP. These PPDUs are separated by SIFS. When using non-NDP sounding, each PPDU contains a +HTC frame in which the MAI subfield is set to ASELI, the ASEL Command subfield is set to Receive Antenna Selection Sounding Indication (RXASSI), and the ASEL Data subfield is set to the remaining number of sounding PPDUs to be transmitted. When using NDP sounding, the PPDU that precedes the first NDP contains a +HTC frame in which the NDP Announce field is set to 1, the MAI subfield is set to ASELI, the ASEL Command subfield is set to RXASSI, and the ASEL Data subfield is set to the remaining number of sounding PPDUs to be transmitted.

The ASEL receiver uses different antenna sets to receive these sounding PPDUs, estimates CSI after receiving all these sounding PPDUs, and conducts the ASEL.

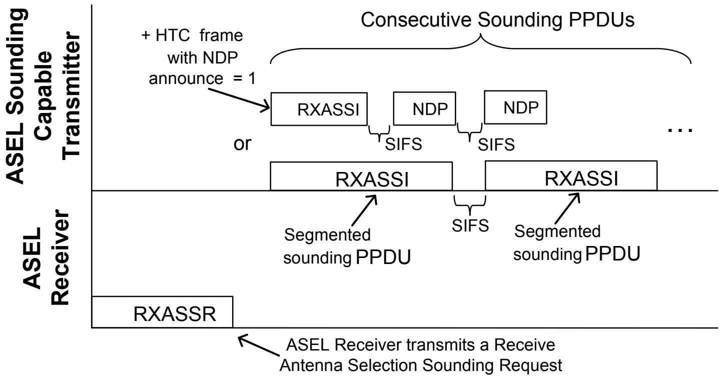

ASEL Sounding Capable Transmitter
ASEL Receiver
+ HTC frame with NDP announce = 1
Consecutive Sounding PPDUs
RXASSI
NDP
NDP
SIFS
SIFS
...
or
RXASSI
RXASSI
SIFS
Segmented sounding PPDU
Segmented sounding PPDU
RXASSR
ASEL Receiver transmits a Receive Antenna Selection Sounding Request

Figure 9-41—Receive ASEL

When transmitting the consecutive sounding PPDUs in transmit and receive ASEL exchanges (illustrated in Figure 9-40 and Figure 9-41), the transmitter shall not change the TXPWR_LEVEL parameter of the TXVECTOR.

When transmitting a sounding PPDU sent in transmit and receive ASEL exchanges (illustrated in Figure 9-40 and Figure 9-41), if the Receive Staggered Capability subfield of the Transmit Beamforming Capability field of the HT Capabilities element transmitted by the intended receiver is 0, then,

— If the sounding PPDU is not an NDP, the number of antennas used by the sender, as indicated by the ANTENNA_SET parameter in the TXVECTOR, shall be less than or equal to the maximum number of space-time streams indicated by the Rx MCS Bitmask subfield of the Supported MCS Set field and the Rx STBC subfield of the HT Capabilities element transmitted by the intended receiver.

— If the sounding PPDU is an NDP, the number of antennas used by the sender, as indicated by the ANTENNA_SET parameter in the TXVECTOR, shall be less than or equal to the number indicated by the Channel Estimation Capability subfield of the Transmit Beamforming Capability field of the HT Capabilities element transmitted by the intended receiver.

When both transmitter and receiver have ASEL capabilities, the following constraints apply:

— During a transmit ASEL frame exchange, the transmit ASEL responder shall use a subset of antennas that does not change during the reception of all of the sounding PPDUs of the ASEL sounding sequence.

— During a receive ASEL frame exchange, the ASEL sounding-capable transmitter shall use a subset of antennas that does not change during the transmission of all of the sounding PPDUs of the ASEL sounding sequence.

NOTE—When a receiver (either a transmit ASEL responder or an ASEL receiver) conducts ASEL computations (for either transmit or receive ASEL), if there is no transmit beamforming conducted at the same time, to achieve the best performance of ASEL, the receiver assumes that the first N_{STS} columns of the same spatial mapping matrix Q_k used for transmitting ASEL sounding PPDUs, where N_{STS} is the number of space-time streams, are applied for the spatial mapping at the ASEL transmitter after the ASEL exchange as in Figure 9-40 and Figure 9-41. To achieve the best performance of ASEL, the ASEL transmitter applies the first N_{STS} columns of the same Q_k for spatial mapping after the ASEL exchange as in Figure 9-40 and Figure 9-41.

9.31 Null data packet (NDP) sounding

9.31.1 NDP rules

Sounding may be accomplished using either staggered sounding PPDU or NDP, as described in 20.3.13. The MAC rules associated with sounding using NDP are described in 9.31.1 to 9.31.4.

An HT STA that has set the Receive NDP Capable field of its HT Capabilities element to 1 during association processes an NDP as a sounding packet if the destination of the sounding packet is determined to match itself as described in 9.31.3 and if the source of the sounding packet can be ascertained as described in 9.31.4.

An RXVECTOR LENGTH parameter equal to 0 indicates that the PPDU is an NDP.

A STA that is a TXOP holder or an RD responder shall not set both the NDP Announcement and RDG/More PPDU subfields to 1 simultaneously. The Calibration Position subfield shall not be set to any value except 0 and 1 in any +HTC frame in a PPDU that is also an NDP announcement. The Calibration Position subfield shall be set to 0 in any +HTC frame in a PPDU that is an NDP announcement that also contains any +HTC frame with the MAI subfield equal to ASELI. The Calibration Position subfield shall be set to 0 in all +HTC frames in a PPDU that is an NDP announcement and that contains any +HTC frame with the MRQ subfield equal to 1. The TRQ field shall be set to 0 in all +HTC frames in a PPDU that is an NDP announcement.

An NDP sequence contains at least one non-NDP PPDU and at least one NDP PPDU. Only one PPDU in the NDP sequence may contain an NDP announcement. An NDP sequence begins with an NDP announcement. The NDP sequence ends at the end of the transmission of the last NDP PPDU that is announced by the NDP announcement. A STA that transmits the first PPDU of an NDP sequence is the NDP sequence owner. In the NDP sequence, only PPDUs carrying NDP and PPDUs carrying non-A-MPDU control frames may follow the NDP sequence's starting PPDU.

A STA shall transmit only one NDP per NDP announcement, unless the NDP announcement includes a value in the ASEL Data subfield of the ASEL Command subfield of the HTC Control field that is greater than one. Each PPDU in an NDP sequence shall start a SIFS interval after end of the previous PPDU.

The +HTC field of a CTS frame shall not contain the NDP Announcement subfield set to 1.

NOTE—A CTS frame cannot be used for NDP announcement: if the CTS frame is a response to an RTS frame, the optional NAV reset timeout that starts at the end of the RTS frame does not include the additional NDP and SIFS duration (see 9.3.2.4). Also, if the CTS were the first frame of an NDP sequence, it would not be possible to determine the destination address of the NDP.

A STA shall transmit an NDP as follows:

a) A SIFS interval after sending a PPDU that is an NDP announcement and that does not contain an MPDU that requires an immediate response.

b) A SIFS interval after successfully receiving a correctly formed and addressed immediate response to a PPDU that is an NDP announcement and that contains an MPDU that requires an immediate response.

c) A SIFS interval after transmitting an NDP if the NDP announcement contains an ASEL Command subfield equal to TXASSI, TXASSI-CSI, or RXASSI and the ASEL Data subfield is equal to a value greater than 0 and if the number of NDPs sent before this one is less than the value in the ASEL Data subfield + 1.

   NOTE—The total number of sent NDPs is equal to the value of in the ASEL Data subfield + 1.

d) A SIFS interval after receiving an NDP from a STA whose NDP announcement contained one or more +HTC frames with the Calibration Position subfield equal to 1, when the receiving STA supports transmitting sounding PPDUs for which more than one channel dimension can be estimated (i.e., more than one column of the MIMO channel matrix).

This rule enables the NDP receiver to know that it will receive an NDP and can determine the source and destination of the NDP. It enables the receiver and transmitter to know when the immediate response and NDP will be transmitted relative to the frame containing the NDP announcement indication.

A STA that has transmitted an NDP announcement in a frame that requires an immediate response and that does not receive the expected response shall terminate the NDP sequence at that point (i.e., the STA does not transmit an NDP in the current NDP sequence).

A STA that has received an NDP announcement in a +HTC with the Calibration Position equal to 1 or 2, and that does not receive the NDP PPDU expected shall terminate the NDP sequence at that point (i.e., does not transmit an NDP in the current NDP sequence) and not transmit any further frames that are a part of this calibration sequence shown in Step 1 of Figure 9-38.

Feedback information generated from the reception of an NDP is transmitted using any of the feedback rules and signaling as appropriate, e.g., immediate or delayed.

9.31.2 Transmission of an NDP

A STA that transmits an NDP shall set the LENGTH, SOUNDING, STBC, MCS, and NUM_EXTEN_SS parameters of the TXVECTOR as specified in this subclause.

— LENGTH shall be set to 0.
— SOUNDING shall be set to SOUNDING.
— STBC shall be set to 0.
— MCS shall indicate two or more spatial streams.

The number of spatial streams sounded is indicated by the MCS parameter of the TXVECTOR and shall not exceed the limit indicated by the Channel Estimation Capability field in the Transmit Beamforming Capabilities field transmitted by the STA that is the intended receiver of the NDP. The MCS parameter may be set to any value, subject to the constraint of the previous sentence, regardless of the value of the Supported MCS Set field of the HT Capabilities field at either the transmitter or recipient of the NDP. A STA shall set the NUM_EXTEN_SS parameter of the TXVECTOR to 0 in the PHY-TXSTART.request primitive corresponding to an NDP transmission.

A STA shall not transmit an NDP announcement with a RA corresponding to another STA unless it has received an HT Capabilities element from the destination STA in which the Receive NDP Capable field is equal to 1.

9.31.3 Determination of NDP destination

The destination of an NDP is determined at the NDP receiver by examining the NDP announcement as follows:

— The destination of the first NDP in the NDP sequence is equal to the RA of any MPDU within NDP announcement.
— If Calibration Position subfield is equal to 1 in the NDP announcement at the NDP receiver, the destination of the second NDP is equal to the TA of that frame. Otherwise, the destination of the second and any subsequent NDPs is equal to the destination of the previous NDP.

See S.4 for an illustration of these rules.

9.31.4 Determination of NDP source

The source of an NDP is determined at the NDP receiver by examining the NDP sequences's starting PPDU as follows:

- If any MPDU within the NDP announcement contains two or more addresses, the source of the first NDP is equal to the TA of that frame.
- Otherwise (i.e., the NDP announcement contains one address), the source of the first NDP is equal to the RA of the MPDU to which the NDP announcement is a response.
- If the Calibration Position subfield is equal to 1 in an MPDU in the NDP announcement, the source of the second NDP is equal to the RA of that MPDU. Otherwise, the source of the second and any subsequent NDPs is equal to the source of the previous NDP.

See S.4 for an illustration of these rules.

9.32 Mesh forwarding framework

9.32.1 General

The term *mesh forwarding* refers to forwarding of MSDUs and MMPDUs on paths determined by the mesh path selection between mesh STAs at the link layer. The mesh paths are contained in the forwarding information. The forwarding information, for instance, the lifetime of a mesh path, may be updated as a consequence of mesh forwarding.

The forwarding of MSDUs and MMPDUs within an MBSS is described in 9.32.4, 9.32.5, 9.32.6, and 9.32.9. The forwarding of MSDUs and MMPDUs between the MBSS and the DS at proxy mesh gates is described in 13.11.3.

9.32.2 Forwarding information

Forwarding information is created by the active mesh path selection protocol and is utilized for MSDU/MMPDU forwarding as described in 9.32.4 and 9.32.6.2.

The basic forwarding information to a destination mesh STA consists of the destination mesh STA address, the next-hop address, the precursor list, and the lifetime of this forwarding information.

An entry in the precursor list contains the precursor mesh STA address and the lifetime of this entry. If an existing entry in a precursor list is updated, the lifetime is the maximum of the current and the updated value. If the lifetime of a precursor expires, it will be deleted from the precursor list. Precursors are used to identify legitimate transmitters of individually addressed frames (see 9.32.4.2) and for the notification of link failures (in case of HWMP, see 13.10.11).

The forwarding information shall be considered as invalid if its lifetime has expired. Also, forwarding information is marked as invalid when certain conditions are met in the processing of mesh path selection elements, e.g., path error processing in HWMP (13.10.11.4).

The active path selection protocol may define additional parameters in the forwarding information. Details on the additional parameters of the forwarding information constructed by the hybrid wireless mesh protocol (HWMP) are described in 13.10.8.4.

9.32.3 Frame addressing in an MBSS

Mesh Data frames and Multihop Action frames enable multihop MSDU and MMPDU forwarding in an MBSS using the Mesh Control field described in 8.2.4.7.3. In this subclause, addressing of the Mesh Data and Multihop Action frames and MSDU/MMPDU forwarding behavior are described.

Table 9-15 shows the valid combinations of address fields in Mesh Data frames and Multihop Action frames along with the corresponding value of the Address Extension Mode subfield in the Mesh Control field.

NOTE—To DS and From DS fields are located in the Frame Control field (see 8.2.4.1.4). The Address Extension Mode subfield is located in the Mesh Flags subfield in the Mesh Control field (see 8.2.4.7.3). Address 1, Address 2, and Address 3 fields are located in the MAC header (see 8.2.3). The Address 4 field is located in the MAC header if both To DS and From DS fields are 1; otherwise, the Address 4 field is located in the Mesh Address Extension subfield of the Mesh Control field (see 8.2.3 and 8.2.4.7.3). Address 5 and Address 6 fields are located in the Mesh Control field if they are present (see 8.2.4.7.3).

Table 9-15—Valid address field usage for Mesh Data and Multihop Action frames

Supported frames	To DS From DS field	Address Extension Mode value (binary)	Address 1	Address 2	Address 3	Address 4	Address 5	Address 6
Mesh Data (individually addressed)	11	00	RA	TA	DA = Mesh DA	SA = Mesh SA	*Not Present*	*Not Present*
Mesh Data (group addressed)	01	00	DA	TA	SA = Mesh SA	*Not Present*	*Not Present*	*Not Present*
Mesh Data (proxied, individually addressed)	11	10	RA	TA	Mesh DA	Mesh SA	DA	SA
Mesh Data (proxied, group addressed)	01	01	DA	TA	Mesh SA	SA	*Not Present*	*Not Present*
Multihop Action (individually addressed)	00	01	RA	TA	DA = Mesh DA	SA = Mesh SA	*Not Present*	*Not Present*
Multihop Action (group addressed)	00	00	DA	TA	SA = Mesh SA	*Not Present*	*Not Present*	*Not Present*

In individually addressed Mesh Data and Multihop Action frames, Address 1 and Address 2 correspond to the mesh STA receiver address (RA) and the mesh STA transmitter address (TA) for a particular mesh link. Address 3 and Address 4 correspond to the destination end station and the source end station of a mesh path. The Address Extension Mode subfield in the Mesh Control field indicates the presence of an optional Mesh Address Extension subfield in the Mesh Control field. When the Extension Mode subfield equals 10 (binary), the Mesh Control field includes Address 5 and Address 6 that correspond to the end-to-end destination address (DA) and source address (SA) of STAs that communicate over the mesh path, for instance, external STAs that communicate over the mesh BSS via proxy mesh gates (see Figure 9-42).

NOTE—The forwarding of individually addressed Mesh Data frames uses only mesh STA addresses in fields Address 1, Address 2, Address 3, and Address 4. This allows intermediate mesh STAs to forward Mesh Data frames without necessarily having any knowledge of the addresses of the source and destination end stations, which might be external addresses. Thus, proxy information only needs to be maintained by proxy mesh gates and by source mesh STAs.

The term source mesh STA refers to the first mesh STA on a mesh path. A source mesh STA may be a mesh STA that is the initial source of an MSDU/MMPDU or a mesh STA that receives an MSDU/MMPDU from a mesh path or from a STA outside the mesh BSS and translates and forwards the MSDU/MMPDU on the mesh path. The address of the source mesh STA is referred to as the Mesh SA.

The term destination mesh STA refers to the final mesh STA on a mesh path. A destination mesh STA may be a mesh STA that is the final destination of an MSDU/MMPDU or a mesh STA that receives an MSDU/MMPDU from a mesh path and translates and forwards the MSDU/MMPDU on another mesh path or to a STA outside of the mesh BSS. The address of the destination mesh STA is referred to as the Mesh DA.

In group addressed Mesh Data frames, Address 1 and Address 2 correspond to the group address and the mesh STA transmitter address (TA). Address 3 corresponds to the mesh source address (mesh SA) of the group addressed Mesh Data frame. The Address Extension Mode indicates the presence of an optional address extension field Address 4 in the Mesh Control field that corresponds to the source address (SA) of external STAs that communicate over the mesh BSS via proxy mesh gates.

NOTE—The reason for not using the four-address MAC header format for group addressed traffic is to avoid interactions with existing implementations. Earlier revisions of this standard defined the four-address MAC header format without defining procedures for its use. As a result there is a large number of deployed devices that use the four-address frame format in ways that would affect and be affected by mesh traffic if four-address group addressed frames were to be used.

Figure 9-42 illustrates the addressing of a Mesh Data frame that contains an MSDU transmitted and forwarded on a mesh path from a mesh STA collocated with a portal (STA 1) to a mesh STA collocated with an AP (STA 2) where the source is a STA outside of the mesh BSS (STA 33) that is reachable via the portal and the destination is an IEEE 802.11 STA associated with the AP (STA 22).

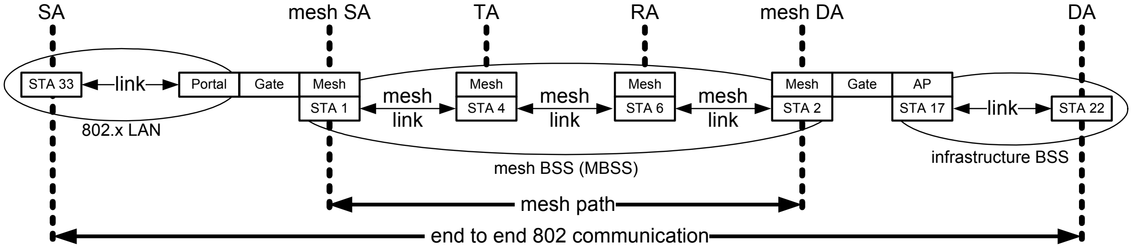


Figure 9-42—Example addressing for a Mesh Data frame

Details on how these address mappings work in forwarding processing are described in 9.32.4 and 9.32.5.

9.32.4 Addressing and forwarding of individually addressed Mesh Data frames

9.32.4.1 At source mesh STAs (individually addressed)

MSDUs sent by a mesh STA (as a consequence of an MA-UNITDATA.request with an individual destination address) and destined to another mesh STA in the MBSS shall be transmitted using a frame with the four-address MAC header format (with the Address Extension Mode subfield in the Mesh Control field set to 00 (binary)), where the four address fields are set as follows (see row "Mesh Data (individually addressed)" in

Table 9-15):

- Address 1: The address of the next-hop mesh STA (toward the destination mesh STA according to the forwarding information—see 9.32.2)
- Address 2: The address of the transmitter mesh STA
- Address 3: The address of the destination mesh STA
- Address 4: The address of the source mesh STA

MSDUs that are sent by a mesh STA as a consequence of a MA-UNITDATA.request with an individual destination address and are destined to an address that is different from the mesh STA at the end of a mesh path shall be transmitted using a frame with the four-address MAC header format [with the Address Extension Mode subfield in the Mesh Control field set to 10 (binary)], where the Mesh Address Extension subfield in the Mesh Control field carries the addresses of the end stations, as specified in row "Mesh Data (proxied, individually addressed)" of Table 9-15. The additional addresses 5 and 6 are defined as follows:

- Address 5: The address of the destination end mesh STA (may be the same as Address 3 if the destination is the mesh STA at the end of the mesh path)
- Address 6: The address of the source end mesh STA (may be the same as Address 4 if the source is the mesh STA at the beginning of the mesh path)

NOTE—The destination address is distinct from the mesh STA at the end of the mesh path in two cases: 1) when the destination is an external address and 2) when the destination is a mesh STA distinct from the destination mesh STA at the end of the mesh path. The former case is described in 13.11.3. The latter case might occur if a source mesh STA sends the MSDU to another intermediate mesh STA that sends the MSDU on a different mesh path to the destination mesh STA in the MBSS.

The Mesh TTL subfield in the Mesh Control field shall be set to the value of dot11MeshTTL.The MSDUs are forwarded multiple hops, limited by the Mesh TTL value.

The source mesh STA shall set the Mesh Sequence Number subfield in the Mesh Control field to a value from a modulo-2^{32} counter that is incremented by 1 for each new MSDU transmitted with a Mesh Control field and for each new MMPDU transmitted using a Multihop Action frame.

9.32.4.2 At intermediate and destination mesh STAs (individually addressed)

On receipt of an individually addressed Mesh Data frame, a mesh STA shall perform the following:

a) The mesh STA shall decipher the frame and check it for authenticity. If it is not from a peer mesh STA, the frame shall be silently discarded.
b) The mesh STA shall check to see whether the MAC address in the Address 3 field is a known destination address; if it is an unknown destination address, the mesh STA may perform any of the following three actions:
   1) Silently discard the frame.
   2) Trigger a path discovery procedure depending on the path selection protocol that is currently active in the mesh BSS. For HWMP, see 13.10.9.3 Case A.
   3) Inform the mesh STA in Address 2 that the destination is unreachable depending on the path selection protocol that is currently active in the mesh BSS. For HWMP, see 13.10.11.3 Case B.
c) If Address 2 is not one of the precursors for this destination mesh STA (see 9.32.2), the frame shall be discarded.

If the frame is not discarded and one or more MSDUs are collected from the frame, the mesh STA may detect duplicate MSDUs according to 9.32.7 and discard them.

If Address 3 does not match the mesh STA's own address, but is a known individual destination MAC address in the forwarding information then the following actions are taken:

- The lifetime of the forwarding information to the destination (Address 3) is set to its initial value.
- The lifetime of the forwarding information to the source (Address 4) is set to its initial value.
- The lifetime of the precursor list entry for the precursor to the destination (Address 2) is set to the maximum of the initial value and the current value.
- The lifetime of the precursor list entry for the precursor to the source (next hop to the destination) is set to the maximum of the initial value and the current value.
- The Mesh TTL in the corresponding Mesh Control field of the collected MSDU is decremented by 1. If zero has been reached, the MSDU shall be discarded.
- If the MSDU has not been discarded, the mesh STA shall forward the MSDU via a frame with the Address 1 field set to the MAC address of the next-hop mesh STA as determined from the forwarding information (see 9.32.2) and the Address 2 field set to its own MAC address and queue the frame for transmission.

If Address 3 matches the mesh STA's own MAC address, the following actions are taken:

- The lifetime of the forwarding information to the source (Address 4) is set to its initial value.
- The lifetime of the precursor list entry for the precursor to the destination (Address 2) is set to the maximum of the initial value and the current value.
- If the Address Extension Mode subfield in the Mesh Control field is 00 (binary), the MA-UNITDATA.indication primitive is passed from the MAC sublayer entity to the LLC sublayer entity or entities.
- If the Address Extension Mode subfield in the Mesh Control field is 10 (binary) and Address 5 is equal to Address 3, the mesh STA is the final destination of the MSDU, and the MA-UNITDATA.indication primitive is passed from the MAC sublayer entity to the LLC sublayer entity or entities.
- If the Address Extension Mode subfield in the Mesh Control field is 10 (binary) and Address 5 is a known destination MAC address in the forwarding information (mesh STA), the mesh STA shall forward the MSDU via a frame as described in 9.32.4.1 with the Address 3 field set to the MAC Address of the Address 5 field.
- If the Address Extension Mode subfield in the Mesh Control field is 10 (binary), the MSDU is forwarded according to 13.11.3.2 in all other cases.

If Address 3 matches the group address, the mesh STA shall perform the procedures as given in 9.32.5.2.

Note that during the forwarding process at intermediate mesh STAs, the content of the MSDU is not changed.

9.32.5 Addressing and forwarding of group addressed Mesh Data frames

9.32.5.1 At source mesh STAs (group addressed)

MSDUs sent by a mesh STA (as a consequence of a MA-UNITDATA.request with a group destination address) shall be transmitted using a group addressed Mesh Data frame (with the Address Extension Mode subfield in the Mesh Control field set to 00 (binary)) (see row "Mesh Data (group addressed)" in Table 9-15). An implementation may circumvent the unreliability of group addressed transmissions by using multiple individually addressed Mesh Data frames, which are individually acknowledged. In such case, the frame may be converted to individually addressed frames and transmitted as individually addressed Mesh Data frames to each peer mesh STA as described in 9.32.4.1 with the Address 3 field set to the group address. The circumstances for choosing this method are outside the scope of the standard.

In group addressed Mesh Data frames, the address fields are set as follows:

- Address 1: The group address

— Address 2: The address of the transmitter mesh STA

— Address 3: The address of the source mesh STA

The source mesh STA shall set the Mesh TTL subfield in the Mesh Control field to dot11MeshTTL in order to control the hop count. The MSDUs are forwarded multiple hops, limited by the Mesh TTL value. For example, if the Mesh TTL subfield is 1, MSDUs are delivered only to immediate neighbors.

The source mesh STA shall set the Mesh Sequence Number subfield in the Mesh Control field to a value from a modulo-2^{32} counter that is incremented by 1 for each new MSDU transmitted with a Mesh Control field and for each new MMPDU transmitted using a Multihop Action frame.

Procedures that enhance the reliability or efficiency of group addressed transmissions are outside the scope of this standard.

9.32.5.2 At recipient mesh STAs (group addressed)

On receipt of a group addressed Mesh Data frame with Address 1 (DA) equal to the group address, or on receipt of an individually addressed Mesh Data frame with Address 3 (Mesh DA) equal to the group address, a mesh STA shall perform the following:

a) The mesh STA shall decipher the frame and check it for authenticity. If it is not from a peer mesh STA, the frame shall be silently discarded.

b) If the frame is not discarded and one or more MSDUs are collected from the frame, the mesh STA may detect duplicate MSDUs according to 9.32.7 and discard them.

c) The mesh STA decrements the Mesh TTL in the Mesh Control field. If the Mesh TTL value has reached zero, the corresponding MSDU shall not be forwarded to other mesh STAs.

d) If the Mesh TTL value has not reached zero and if dot11MeshForwarding is true, the mesh STA shall forward the MSDU via a group address Mesh Data frame with the Address 2 field set to its own MAC address.

e) If the Address Extension Mode is 01 (binary) and the recipient mesh STA is a proxy mesh gate and if the Mesh TTL value has not reached zero and if dot11MeshForwarding is true, the MSDU is forwarded according to 13.11.3.2.

When the SA and the Mesh SA are not identical (the source address is therefore an external address), the MSDU shall be forwarded by using a frame with the three-address MAC header format [with the Address Extension Mode subfield in the Mesh Control field set to 01 (binary)] as specified in row “Mesh Data (proxied, group addressed)” of Table 9-15. Otherwise, the MSDU shall be forwarded by using a frame with the three-address MAC header format [with the Address Extension Mode subfield in the Mesh Control field set to 00 (binary)] as specified in row “Mesh Data (group addressed)” of Table 9-15.

An implementation may circumvent the unreliability of group addressed transmissions by using multiple individually addressed Mesh Data frames, which are individually acknowledged. In such case, the frame may be converted to individually addressed frames and transmitted as an individually addressed Mesh Data frame to each peer mesh STAs as described in 9.32.4.2 with the Address 3 field set to the group address. If the Address Extension Mode subfield in the Mesh Control field in the group addressed Mesh Data frame is equal to 01 (binary), the Address Extension Mode subfield in the Mesh Control field in the individually addressed Mesh Data frames is set to 10 (binary), the Address 5 field is set to the group address, and the Address 6 field set to the Source Address contained in the Address 4 field of the group address Mesh Data frame. The circumstances for choosing this method and the ability to determine all the addresses of the neighbor peer mesh STAs are beyond the scope of the standard.

If one or more MSDUs collected from the frame have not been discarded, the MA-UNITDATA.indication primitive is passed from the MAC sublayer entity to the LLC sublayer entity or entities.

9.32.6 Addressing of Management frames and MMPDU forwarding

9.32.6.1 General

All MMPDUs except MMPDUs transmitted using Multihop Action frames are transmitted over only one hop to peer mesh STAs.

NOTE—In several cases, the reception and processing of an Action frame leads to the transmission of a new Action frame of the same type that might include an identical or a modified version of the contents from the elements of the received Action frame. This is called propagation in contrast to forwarding.

A mesh STA may convert a group addressed management frame to individually addressed management frames and transmit them as individually addressed frames to each peer mesh STA, if the frame is intended to be delivered only to its peer mesh STAs. The circumstances for choosing this method are outside the scope of the standard.

9.32.6.2 MMPDU forwarding using individually addressed Multihop Action frames

MMPDUs sent by a mesh STA and destined to another mesh STA in the MBSS using individually addressed Multihop Action frames (see 8.5.18) shall be transmitted using a management frame with the three-address MAC header format (with the Address Extension Mode subfield in the Mesh Control field set to 01 (binary)), where the four address fields are set as follows (see row “Multihop Action (individually addressed)”) in Table 9-15:

- Address 1: The address of the next-hop mesh STA (toward the destination mesh STA according to the forwarding information—see 9.32.2.
- Address 2: The address of the transmitter mesh STA.
- Address 3: The address of the destination mesh STA.
- Address 4: The address of the source mesh STA.

The source mesh STA shall set the Mesh TTL subfield in the Mesh Control field to the value of dot11MeshTTL, and set the Mesh Sequence Number subfield in the Mesh Control field to a value from a modulo-2^{32} counter that is incremented by 1 for each new MSDU transmitted with a Mesh Control field and for each new MMPDU transmitted using a Multihop Action frame.

At intermediate and destination mesh STAs, on receipt of an individually addressed Multihop Action frame, the address matching, the reception procedures, the forwarding information update, and the Mesh TTL decrement are performed as described in 9.32.4.2, and the MMPDU is forwarded according to the forwarding information and the procedures in 9.32.4.2.

At intermediate mesh STAs, frame fields following the Mesh Control field are not required to be examined.

If the Address 3 in the received Multihop Action frame matches the mesh STA’s own MAC address or the group address, the mesh STA (destination mesh STA) shall process the content of the MMPDU.

9.32.6.3 MMPDU forwarding using group addressed Multihop Action frames

MMPDUs sent by a mesh STA and destined to all other mesh STAs in the MBSS using group addressed Multihop Action frames (see 8.5.18) shall be transmitted using a management frame with the three-address MAC header format (with the Address Extension Mode subfield in the Mesh Control field set to 00 (binary)), where the three address fields are set as follows (see row “Multihop Action (group addressed)” in Table 9-15):

- Address 1: The group address
- Address 2: The address of the transmitter mesh STA

— Address 3: The address of the source mesh STA

An implementation may circumvent the unreliability of group addressed transmissions by using multiple individually addressed Multihop Action frames, which are individually acknowledged. In such case, the frame may be converted to individually addressed frames and transmitted as individually addressed Multihop Action frames to each peer mesh STA as described in 9.32.6.2 with the Address 3 field set to the group address. The circumstances for choosing this method are outside the scope of the standard.

The source mesh STA shall set the Mesh TTL subfield in the Mesh Control field to dot11MeshTTL, and set the Mesh Sequence Number subfield in the Mesh Control field to a value from a modulo-2^{32} counter that is incremented by 1 for each new MSDU transmitted with a Mesh Control field and for each new MMPDU transmitted using a Multihop Action frame.

At recipient mesh STAs, on receipt of Multihop Action frame, the address matching, the reception procedures, the forwarding information update, and the Mesh TTL decrement are performed as described in 9.32.5.2, and the MMPDU is forwarded according to the forwarding information and the procedures in 9.32.5.2.

If the Address 1 in the received Multihop Action frame matches the group address, the mesh STA shall process the content of the MMPDU.

Procedures that enhance the reliability or efficiency of group addressed transmissions are outside the scope of this standard.

9.32.7 Detection of duplicate MSDUs/MMPDUs

A mesh STA may receive multiple copies of the same MSDU or MMPDU from different neighbor peer mesh STAs.

The filtering of such duplicates is facilitated through the inclusion of a Mesh Sequence Number subfield in the Mesh Control field in Mesh Data frames and Multihop Action frames as specified in 8.2.4.7.3.

The receiving mesh STA shall keep a cache of recently received <Mesh SA, Mesh Sequence Number> tuples. The Mesh Source Address (Mesh SA) is contained in Address 4 for individually addressed Mesh Data frames and Multihop Action frames. The Mesh Source Address (Mesh SA) is contained in Address 3 for group addressed Mesh Data frames.

A mesh STA shall reject an MSDU/MMPDU with a Mesh Control field as a duplicate if it matches a <Mesh SA, Mesh Sequence Number> tuple of an entry in the cache.

The rules in 9.3.2.10 also apply to the filtering of duplicates sent by the same neighbor peer mesh STA.

9.32.8 Mesh STAs that do not forward

A mesh STA that has dot11MeshForwarding equal to false does not forward either MSDUs, or MMPDUs of type Multihop Action. The circumstances in which a mesh STA may be allowed to become a non-forwarding entity and the authority to set dot11MeshForwarding to false are beyond the scope of this standard.

A mesh STA that does not forward is a special case of a mesh STA. Such mechanism depends on whether the path selection protocol provides a mechanism to allow mesh STAs not to participate in forwarding. The HWMP path selection protocol provides such a mechanism; see 13.10.

9.32.9 Frame forwarding and unknown destination

A source mesh STA in the MBSS might not able to forward an MSDU that it has received as a consequence of an MA-UNITDATA.request with an individual destination address. This is the case if the destination of the MSDU is unknown to the mesh STA. The destination is unknown to a mesh STA if the mesh STA has no forwarding information for this destination or if the destination is not in its proxy information as an external STA (see 13.11.4.2). Note that the procedure to determine that an address is unknown depends on the active path selection protocol. It may require an attempt to establish a path to the destination (see 13.8).

If the source mesh STA is not able to forward the frame because its destination is unknown, the mesh STA shall assume that the destination is outside the MBSS and shall forward the frame to known mesh gates in the MBSS as an individually addressed frame according to the procedures for frame addressing and data forwarding of individually addressed frames at source mesh STAs in an MBSS (9.32.4.1). The MSDU shall be transmitted using a frame with the four-address MAC header format (with the Address Extension Mode subfield in the Mesh Control field set to 10 (binary)), where the Mesh Address Extension subfield in the Mesh Control field carries the address of the destination end station, as specified in row “Mesh Data (proxied, individually addressed)” of Table 9-15. The address fields are set as follows:

- Address 1: The address of the next-hop mesh STA (toward the known mesh gate in the MBSS according to the forwarding information—see 9.32.2).
- Address 2: The address of the source mesh STA.
- Address 3: The address of the known mesh STA in the MBSS.
- Address 4: The address of the source mesh STA.
- Address 5: The address of the destination end mesh STA, which is the unknown destination address of the MSDU
- Address 6: The address of the source mesh STA, which is the same as Address 4

If there is no mesh gate available, the mesh STA shall silently discard the frame.

Discovery of mesh gates by mesh STAs is performed using propagated elements, such as a GANN (13.11.2). Other methods specific to the HWMP path selection protocol are also available, such as the proactive PREQ (13.10.4.2) or the proactive RANN (13.10.4.3), when the Gate Announcement subfield in the Flags field in these HWMP elements is set to 1.

10. MLME

10.1 Synchronization

10.1.1 General

STAs in a single infrastructure BSS or IBSS are synchronized to a common clock using the mechanisms defined in 10.1.

STAs in mesh BSSs use a synchronization method that is part of the extensible synchronization framework. The synchronization in an MBSS is described in 13.13.

A STA for which dot11OCBActivated is true is not a member of a BSS and, therefore, is not required to synchronize to a common clock or to use these mechanisms.

A Timing Synchronization Function (TSF) keeps the timers for all STAs in the same BSS synchronized. All STAs in which dot11OCBActivated is false maintain a local TSF timer. STAs in which dot11OCBActivated is true may maintain a TSF timer for purposes other than synchronization.

10.1.2 Basic approach

10.1.2.1 TSF for infrastructure networks

In an infrastructure BSS, the AP shall be the timing master for the TSF. The AP shall initialize its TSF timer independently of any simultaneously started APs in an effort to minimize the synchronization of the TSF timers of multiple APs. The AP shall periodically transmit special frames called *Beacon frames* that contain the value of its TSF timer in order to synchronize the TSF timers of other STAs in a BSS. A receiving STA shall accept the timing information in Beacon frames sent from the AP servicing its BSS. If a STA’s TSF timer is different from the timestamp in the received Beacon frame, the receiving STA shall set its local TSF timer to the received timestamp value.

Beacon frames shall be generated for transmission by the AP once every dot11BeaconPeriod TUs.

10.1.2.2 TSF for an IBSS

The TSF in an IBSS shall be implemented via a distributed algorithm that shall be performed by all of the members of the BSS. Each STA in the IBSS shall transmit Beacon frames according to the algorithm described in this clause. Each STA in an IBSS shall adopt the TSF value received from any Beacon frame or probe response from the IBSS of which it is a member and which has a TSF value later than its own TSF timer.

10.1.2.3 TSF for an MBSS

The TSF in an MBSS is provided by the MBSS’s active synchronization method. A mesh STA shall initialize its TSF timer according to the MBSS’s active synchronization method. The mesh STA shall periodically transmit Beacon frames that contain the value of its TSF timer to announce its local time reference. Mesh STAs receiving a Beacon frame use the timing information in the Beacon frame as specified by the MBSS’s active synchronization method. See 13.13.2 for details.

10.1.3 Maintaining synchronization

10.1.3.1 General

Each STA shall maintain a TSF timer with modulus 2^{64} counting in increments of microseconds. STAs expect to receive Beacon frames at a nominal rate. The interval between Beacon frames is defined by the dot11BeaconPeriod parameter of the STA. A STA sending a Beacon frame shall set the value of the Beacon frame's timestamp so that it equals the value of the STA's TSF timer at the time that the data symbol containing the first bit of the timestamp is transmitted to the PHY plus the transmitting STA's delays through its local PHY from the MAC-PHY interface to its interface with the WM [e.g., antenna, light-emitting diode (LED) emission surface].

10.1.3.2 Beacon generation in infrastructure networks

The AP shall define the timing for the entire BSS by transmitting Beacon frames according to dot11BeaconPeriod. This defines a series of TBTTs exactly dot11BeaconPeriod TUs apart. Time 0 is defined to be a TBTT with the Beacon frame being a DTIM. At each TBTT, the AP shall schedule a Beacon frame as the next frame for transmission according to the medium access rules specified in Clause 9. The beacon period is included in Beacon and Probe Response frames, and a STA shall adopt that beacon period when joining the BSS, i.e., the STA sets its dot11BeaconPeriod variable to that beacon period.

NOTE—Though the transmission of a Beacon frame may be delayed because of CSMA deferrals, subsequent Beacon frames are scheduled at the undelayed nominal beacon interval. This is shown in Figure 10-1.

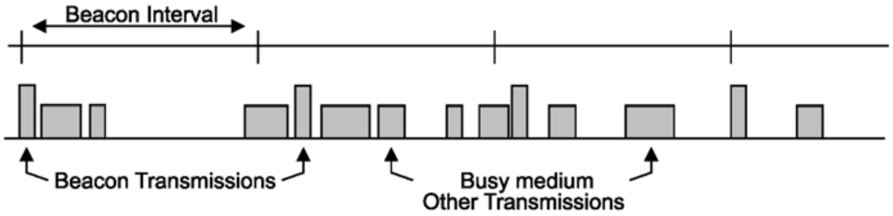


Figure 10-1—Beacon transmission on a busy network

If a STA that does not support short slot time associates with an AP that supports Clause 19 operation, the AP shall use long slot time beginning at the first Beacon subsequent to the association of the long slot time STA.

An AP whose last transmitted values for the Tx STBC subfield and Rx STBC subfield of the HT Capabilities Info field of the HT Capabilities element are both nonzero may transmit an STBC Beacon frame and group addressed traffic using the basic STBC MCS, as defined in 9.7.3. An AP that transmits an STBC Beacon shall set the Dual Beacon field to 1 in transmitted HT Operation elements. The STBC Beacon field shall be set to 1 to identify an STBC Beacon frame. The TBTT for the STBC Beacon frame shall be offset by half of a beacon interval from the TBTT of the non-STBC Beacon frame. Except for the setting of the STBC Beacon field, TIM field, and TSF field, all other fields inside the STBC Beacon frame shall be identical to the non-STBC Beacon frame.

10.1.3.3 Beacon generation in an IBSS

Beacon generation in an IBSS is distributed. The beacon period is included in Beacon and Probe Response frames, and STAs shall adopt that beacon period when joining the IBSS. All members of the IBSS participate in beacon generation. Each STA shall maintain its own TSF timer that is used for dot11BeaconPeriod timing. The beacon interval within an IBSS is established by the STA at which the MLME-START.request primitive is performed to create the IBSS. This defines a series of TBTTs exactly dot11BeaconPeriod TUs apart. Time zero is defined to be a TBTT. At each TBTT the STA shall

a) Suspend the decrementing of the backoff timer for any pending non-Beacon transmission,
b) Calculate a random delay uniformly distributed in the range between zero and twice aCWmin × aSlotTime,
c) Wait for the period of the random delay, decrementing the random delay timer using the same algorithm as for backoff,
d) Cancel the remaining random delay and the pending Beacon frame transmission, if a Beacon frame arrives from the IBSS of which the STA is a member before the random delay timer has expired,
e) Send a Beacon frame if the random delay has expired and no Beacon frame has arrived from the IBSS of which the STA is a member during the delay period,
f) If the ATIM Window in use within the IBSS is greater than 0, then
    1) Resume decrementing the backoff timer for any pending transmission allowed inside the ATIM window and
    2) At the end of the ATIM Window duration resume the backoff for any pending frames intended for transmission outside the ATIM Window,
g) If the ATIM Window in use within the IBSS is 0, then resume decrementing the backoff timer for any pending transmissions.

Figure 10-2 illustrates Beacon transmission in an IBSS.

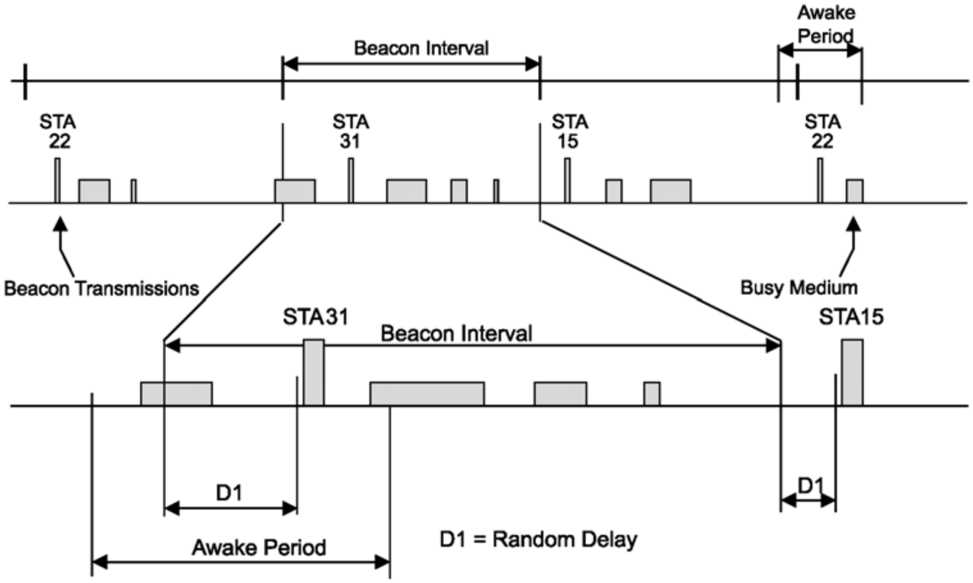


Figure 10-2—Beacon transmission in an IBSS

A STA that has joined an IBSS shall transmit Beacon frames only during the awake period of the IBSS. This is described in more detail in 10.2.

10.1.3.4 Beacon generation in an MBSS

Beacon generation in an MBSS is described in 13.13.3.1.

10.1.3.5 Beacon reception

STAs shall use information from the CF Parameter Set element of all received Beacon frames, without regard for the BSSID, to update their NAV as specified in 9.4.3.3.

STAs in an infrastructure network shall use information that is not in the CF Parameter Set element in received Beacon frames only if the BSSID field is equal to the MAC address currently in use by the STA contained in the AP of the BSS. Non-AP STAs in an infrastructure network that support the Multiple BSSID capability shall use other information in received Beacon frames only if the BSSID field of a non-AP STA is equal to the MAC address currently in use by the STA contained in the AP of the BSS corresponding to the transmitted BSSID or if the BSSID field of a non-AP STA is equal to one of the nontransmitted BSSIDs.

STAs in an IBSS shall use information that is not in the CF Parameter Set element in any received Beacon frame for which the IBSS subfield of the Capability field is 1, the content of the SSID element is equal to the SSID of the IBSS, and the TSF value is later than the receiving STA's TSF timer. Use of this information is specified in 10.1.5.

STAs in an MBSS shall use information in received Beacon frames as described in 13.13.3.2.

When dot11MgmtOptionMultiBSSIDActivated is true and the non-AP STA is associated to the BSS corresponding to the nontransmitted BSSID, a non-AP STA shall support frame filtering for up to two BSSIDs, one for the transmitted BSSID and one for the nontransmitted BSSID, where the non-AP STA shall discard all data frames and management frames except Beacon, Probe Response, and TIM broadcast frames that use the transmitted BSSID as the transmit address.

10.1.3.6 Multiple BSSID procedure

Implementation of the Multiple BSSID capability is optional for a WNM STA. A STA that implements the Multiple BSSID capability has dot11MgmtOptionMultiBSSIDImplemented set to true. When dot11MgmtOptionMultiBSSIDImplemented is true, dot11WirelessManagementImplemented shall be set true. A STA that has a value of true for dot11MgmtOptionMultiBSSIDActivated is defined as a STA that supports the Multiple BSSID capability. A STA for which dot11MgmtOptionMultiBSSIDActivated is true shall set the Multiple BSSID field of the Extended Capabilities element to 1.

The nontransmitted BSSID profile shall include the SSID element (see 8.4.2.2) and Multiple BSSID-Index element (see 8.4.2.76) for each of the supported BSSIDs. The AP may optionally include all other elements in the nontransmitted BSSID profile. The AP may include two or more Multiple BSSID elements containing elements for a given BSSID index in one Beacon frame. If two or more are given, the profile is considered to be the complete set of all elements given in all such Multiple BSSID elements sharing the same BSSID index. Since the Multiple BSSID element is also present in Probe Response frames, an AP may choose to advertise the complete or a partial profile of a BSS corresponding to a nontransmitted BSSID only in the Probe Response frames. In addition, the AP may choose to only include a partial list of nontransmitted BSSID profiles in the Beacon frame or to include different sets of nontransmitted BSSID profiles in different Beacon frames.

When a station receives a Beacon frame with a Multiple BSSID element that consists of a nontransmitted BSSID profile with only the mandatory elements, it may inherit the complete profile from a previously received Beacon frame or Probe Response frame, or send a Probe Request frame to obtain the complete BSSID profiles. Each Beacon element not transmitted in a nontransmitted BSSID subelement is inherited from previous Beacon or Probe Response in which the element is present, except for the Quiet element, which shall take effect only in the Beacon frame that contains it and not carry forward as a part of the inheritance. An AP is not required to include all supported nontransmitted BSSID profiles in a Probe Response frame, and may choose to only include a subset based on any criteria. When a nontransmitted BSSID profile is present in the Multiple BSSID element of the Probe Response frame, the AP shall include all elements that are specific to this BSS. If any of the optional elements are not present in a nontransmitted BSSID profile, the corresponding values are the element values of the transmitted BSSID.

A non-AP STA derives its nontransmitted BSSID value according to 8.4.2.48 and 8.4.2.76.

The Partial Virtual Bitmap field in the transmitted BSSID Beacon frame shall indicate the presence or absence of traffic to be delivered to all stations associated to a transmitted or nontransmitted BSSID. The first 2^n bits of the bitmap are reserved for the indication of group addressed frame for the transmitted and all nontransmitted BSSIDs. The AID space is shared by all BSSs and the lowest AID value that shall be assigned to a station is 2^n (see 8.4.2.7).

If the Contention Free Period is supported and if more than one BSS's CFPCount becomes 0 in the same Beacon frame, the AP shall concatenate the Contention Free Periods of all CFPs that coincide and shall not transmit a CF-End or CF-End+Ack until the end of the concatenated CFP, indicated with a single CF-End or CF-End+Ack, if required. The CF Parameter Set in the transmitted BSSID contains times that are an aggregate of CFP times of the nontransmitted BSSIDs.

Multiple BSSID rate selection is defined in 9.7.7.

10.1.3.7 TSF timer accuracy

Upon receiving a Beacon frame with a valid FCS and BSSID or SSID, as described in 10.1.3.5, a STA shall update its TSF timer according to the following algorithm: The received timestamp value shall be adjusted by adding an amount equal to the receiving STA's delay through its local PHY components plus the time since the first bit of the timestamp was received at the MAC/PHY interface. In the case of an infrastructure BSS, the STA's TSF timer shall then be set to the adjusted value of the timestamp. In the case of an IBSS, the STA's TSF timer shall be set to the adjusted value of the received timestamp, if the adjusted value of the timestamp is later than the value of the STA's TSF timer. The accuracy of the TSF timer shall be no worse than ±0.01%.

When an STA is associated to a BSS with a nontransmitted BSSID, it shall use the TSF from the transmitted BSSID beacon frame.

10.1.4 Acquiring synchronization, scanning

10.1.4.1 General

A STA shall operate in either a Passive Scanning mode or an Active Scanning mode depending on the current value of the ScanMode parameter of the MLME-SCAN.request primitive.

Active scanning is prohibited in some frequency bands and regulatory domains. The MAC of a STA receiving an MLME-SCAN.request primitive shall use the regulatory domain information it has to process the request and shall return a result code of NOT_SUPPORTED to a request for any active scan if regulatory domain information indicates an active scan is illegal. Where regulations permit active scanning after certain conditions are met, the active scan shall proceed after those conditions are met.

Upon receipt of the MLME-SCAN.request primitive, a STA shall perform scanning. The SSID parameter indicates the SSID for which to scan. The SSID List parameter indicates one or more SSIDs for which to scan. To become a member of a particular ESS using passive scanning, a STA shall scan for Beacon frames containing that ESS's SSID, returning all Beacon frames matching the desired SSID in the BSSDescriptionSet parameter of the corresponding MLME-SCAN.confirm primitive with the appropriate bits in the Capabilities Information field indicating whether the Beacon frame came from an infrastructure BSS or IBSS. If the value of dot11RMMeasurementPilotActivated is greater than 1, the STA shall additionally scan for Measurement Pilot frames, returning in the BSSDescriptionFromMeasurementPilotSet parameter all Measurement Pilot frames that equal the requested BSSID of the corresponding MLME-SCAN.request primitive and are not already members of the BSSDescriptionSet. To actively scan, the STA shall transmit Probe request frames containing the desired SSID or one or more SSID List elements. When the SSID List element is present in the Probe Request frame, one or more of the SSID elements may include a wildcard SSID (see 8.4.2.2). The exact procedure for determining the SSID or SSID List values in the MLME-SCAN.request primitive is not specified in this standard. When a STA scans for a BSS whose AP does not support the SSID List element, or for a BSS for which AP support of the SSID List element is unknown, the SSID element with an SSID or wildcard SSID shall be included in the MLME-SCAN.request primitive. Upon completion of scanning, an MLME-SCAN.confirm primitive is issued by the MLME indicating all of the BSS information received.

NOTE—MLME-SCAN.request primitives and resulting Probe Request frames may include a Request element that can be used to request radio measurement information from the scanned BSSs. Requested radio measurement information from the scanned BSSs is included in the Probe Response frames and in the MLME-SCAN.confirm primitive.

Upon receipt of an MLME-JOIN.request primitive, the nonmesh STA shall use the synchronization procedure described in 10.1.4.5. The MLME-JOIN.request primitive is not used to start synchronization in an MBSS. The synchronization in an MBSS is described in 13.13.

Upon receipt of an MLME-SCAN.request primitive with the SSID parameter equal to the wildcard SSID, the STA shall passively scan for any Beacon or Measurement Pilot frames, or actively transmit Probe request frames containing the wildcard SSID, as appropriate depending upon the value of ScanMode. Upon completion of scanning, an MLME-SCAN.confirm primitive is issued by the MLME indicating all of the BSS information received.

If a STA's scanning does not result in finding a BSS with the desired SSID and of the desired type, or does not result in finding any BSS, the STA may start an IBSS upon receipt of the MLME-START.request primitive. The MAC of a STA receiving an MLME-START.request primitive shall use the regulatory domain information it has to process the request and shall return a result code of NOT_SUPPORTED to the request if regulatory domain information indicates starting the IBSS is illegal.

When scanning MBSSs, the STA shall process the procedures described in 13.2.

When a STA starts a BSS, that STA shall determine the BSSID of the BSS. If the BSSType indicates an infrastructure BSS, then the STA shall start an infrastructure BSS and the BSSID shall be equal to the STA's dot11StationID. The value of the BSSID shall remain unchanged, even if the value of dot11StationID is changed after the completion of the MLME-START.request primitive. If the BSSType indicates an IBSS, the STA shall start an IBSS, and the BSSID shall be an individual locally administered IEEE MAC address as defined in 9.2 of IEEE Std 802-2001. The remaining 46 bits of that MAC address shall be a number selected in a manner that minimizes the probability of STAs generating the same number, even when those STAs are subjected to the same initial conditions. The value SSID parameter shall be used as the SSID of the new BSS. It is important that designers recognize the need for statistical independence among the random number streams among STAs.

10.1.4.2 Passive scanning

If the ScanType parameter indicates a passive scan, the STA shall listen to each channel scanned for no longer than a maximum duration defined by the MaxChannelTime parameter.

10.1.4.3 Active scanning

10.1.4.3.1 Introduction

Active scanning involves the generation of Probe request frames and the subsequent processing of received Probe Response frames. The details of the active scanning procedures are as specified in the following subclauses.

10.1.4.3.2 Sending a probe response

STAs, subject to the criteria below, receiving Probe Request frames shall respond with a probe response only if:

a) The Address 1 field in the probe request is the broadcast address or the specific MAC address of the STA, and either item b) or item c) below.

b) The STA is a mesh STA and the Mesh ID in the probe request is the wildcard Mesh ID or the specific Mesh ID of the STA.

c) The STA is not a mesh STA and

   1) The SSID in the probe request is the wildcard SSID, the SSID in the probe request is the specific SSID of the STA, or the specific SSID of the STA is included in the SSID List element, and

   2) The Address 3 field in the probe request is the wildcard BSSID or the BSSID of the STA.

Additionally, STAs with dot11InterworkingServiceActivated equal to true, receiving Probe Request frames containing an Interworking field in the Extended Capabilities element set to 1 shall examine the Interworking element in the received Probe Request frame and respond with a probe response only if

— The HESSID field, if present in the Interworking element, is the wildcard HESSID or the HESSID of the STA, and

— The Access Network Type field in the Interworking element is the wildcard Access Network Type or the Access Network Type of the STA.

Only APs and STAs in an IBSS or in an MBSS respond to probe requests. A result of the procedures defined in this subclause is that in each infrastructure BSS and IBSS there is at least one STA that is awake at any given time to receive and respond to probe requests. In an MBSS, STAs might not be awake at any given time to respond to probe requests. In an infrastructure BSS or in an IBSS, a STA that sent a Beacon frame shall remain in the Awake state and shall respond to probe requests, subject to criteria in the next paragraph, until a Beacon frame with the current BSSID is received. If the STA is contained within an AP, it shall remain in the Awake state and always respond to probe requests, subject to criteria in the next paragraph. There may be more than one STA in an IBSS that responds to any given probe request, particularly in cases where more than one STA transmitted a Beacon frame following the most recent TBTT, either due to not receiving successfully a previous Beacon frame or due to collisions between beacon transmissions.

In an infrastructure BSS or in an IBSS, STAs receiving Probe Request frames shall respond with a probe response when the SSID in the probe request is the wildcard SSID or matches the specific SSID of the STA or when the specific SSID of the STA is included in the SSID List element. Furthermore, a STA with dot11RadioMeasurementActivated true receiving a probe request with a DSSS Parameter Set element containing a Current Channel field value that is not the same as the value of dot11CurrentChannel shall not respond with a probe response. An AP shall respond to all probe requests meeting the above criteria. In an IBSS a STA that transmitted a Beacon frame since the last TBTT shall respond to group addressed Probe Request frames. A STA in an IBSS shall respond to Probe Request frames sent to the individual address of the STA.

An associated mesh STA that receives a Probe Request frame shall not respond with a Probe Response frame when dot11RadioMeasurementActivated is true and the Probe Request frame contains a DSSS Parameter Set element with its Current Channel field value different from the value of dot11CurrentChannelNumber.

Probe Response frames shall be sent as directed frames to the address of the STA that generated the probe request. The SSID List element shall not be included in a Probe Request frame in an IBSS.

Requested Element IDs in the Request element shall be included in the Probe Response if the responding STA supports it. In an improperly formed Request element, a STA may ignore the first element requested that is not ordered properly and all subsequent elements requested. In the probe response frame, the STA shall return the requested elements in the same order as requested in the Request element.

If dot11RadioMeasurementActivated is true and if the Request element of the Probe Request includes the RCPI element ID, the STA shall include in the Probe Response an RCPI element containing the measured RCPI value of the received Probe Request frame. If no measurement result is available, the RCPI value shall be set to indicate that a measurement is not available.

10.1.4.3.3 Active scanning procedure

Upon receipt of the MLME-SCAN.request primitive with ScanType indicating an active scan, a STA shall use the following procedure:

For each channel to be scanned:

a) Wait until the ProbeDelay time has expired or a PHYRxStart.indication primitive has been received.
b) Perform the Basic Access procedure as defined in 9.3.4.2.
c) Send a probe request to the broadcast destination address, with the SSID and BSSID from the MLME-SCAN.request primitive. When the SSID List is present in the MLME-SCAN.request primitive, send one or more probe request frames, each with an SSID indicated in the SSID List and the BSSID from the MLME-SCAN.request primitive.
d) Set to 0 and start a ProbeTimer.
e) If PHY-CCA.indication (busy) primitive has not been detected before the ProbeTimer reaches MinChannelTime, then set NAV to 0 and scan the next channel, else when ProbeTimer reaches MaxChannelTime, process all received probe responses.
f) Set NAV to 0 and scan the next channel.

See Figure 10-3.

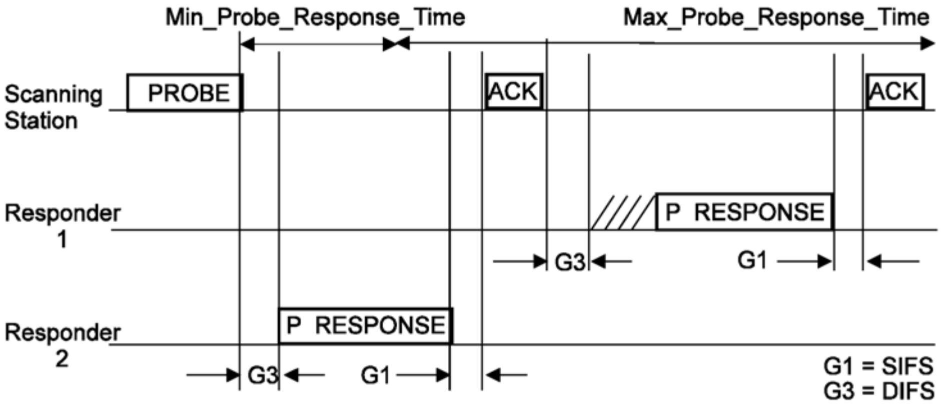


Figure 10-3—Probe response

When all channels in the ChannelList have been scanned, the MLME shall issue an MLME-SCAN.confirm primitive with the BSSDescriptionSet containing all of the information gathered during the scan.

10.1.4.4 Initializing a BSS

Upon receipt of an MLME-START.request primitive, a STA shall determine the BSS's BSSID (as described in 10.1.4), select channel synchronization information, select a beacon period, select the operational rate set, initialize and start its TSF timer, and begin transmitting Beacon frames.

A STA shall include a Country element in the transmission of Beacon frames if dot11MultiDomainCapabilityActivated, dot11SpectrumManagementRequired, or dot11RadioMeasurementActivated is true. See 8.3.3.2 for the description of a properly formed Beacon frame.

10.1.4.5 Synchronizing with a BSS

Upon receipt of an MLME-JOIN.request primitive, a STA shall adopt the BSSID in the request. Upon receipt of a Beacon frame from the BSS, a STA shall adopt the channel synchronization information (applicable only if the STA contains an FH PHY), and TSF timer value of the parameters in the Beacon frame using the algorithm described in 10.1.3.7, and the MLME shall issue an MLME-JOIN.confirm primitive indicating the operation was successful.

In addition to these synchronization parameters, a STA joining an infrastructure BSS adopts each of the parameters found in the BssDescription of the MLME-JOIN.request primitive except Local time, Capability Information, BSSBasicRateSet parameters, and HT Capabilities element. Local time is not adopted but is used as a local variable in adopting the TSF as described in 10.1.3.7. The Capability Information reflects the capabilities of the sender and is not adopted but may be used to determine local configuration or behavior. The BSSBasicRateSet parameter is not adopted but may determine if the STA can join the BSS.

If the JoinFailureTimeout timer expires prior to the receipt of a Beacon frame from the BSS, the MLME shall issue an MLME-JOIN.confirm primitive indicating the operation was unsuccessful.

If dot11MultiDomainCapabilityActivated is true, a STA that is joining an infrastructure BSS and receives a Beacon or Probe Response frame containing a Country element shall adopt the applicable parameters included in that Country element, and the dot11RegDomainsSupportedEntry shall be set to Other.

If a Hopping Pattern Parameters element is present in the Beacon or Probe Response frame, and if dot11MultiDomainCapabilityActivated is true, a STA that is joining an infrastructure BSS shall adopt the pattern parameters in the element and calculate the hopping patterns using one of the algorithms defined in 8.4.2.11 or 8.4.2.12. Using the appropriate pattern, set, and index values from the FH Parameter Set element, the STA shall adopt the values in use by the BSS when joining. The dot11RegDomainsImplementedValue shall be set to Other when the STA is operating using Country element settings.

In addition to adopting the synchronization parameters as described in the first paragraph of this subclause, a STA joining an IBSS shall adopt each of the parameters found in the BSSDescription of the MLME-JOIN.request primitive according to the rule found for that parameter in the "IBSS adoption" column of the matching row of the BSSDescription table found in 6.3.3.3.2. Parameters adopted by a STA when joining an IBSS shall not be changed by the STA except when adopting parameters following the reception of a Beacon frame with a later timestamp as described in 10.1.5.

In addition to the table entries in 6.3.3.3.2, if dot11MultiDomainCapabilityActivated is true, a STA that is joining an IBSS and receives a Beacon or Probe Response frame containing a Country element shall adopt the applicable parameters included in that Country element, and the dot11RegDomainsSupportedEntry shall be set to Other.

In addition to the table entries in 6.3.3.3.2, if a Hopping Pattern Parameters element is present in the Beacon or Probe Response frame, and if dot11MultiDomainCapabilityActivated is true, a STA that is joining an

IBSS shall adopt the pattern parameters in the element and calculate the hopping patterns using one of the algorithms defined in 8.4.2.11 or 8.4.2.12. Using the appropriate pattern, set, and index values from the FH Parameter Set element, the STA shall adopt the values in use by the IBSS when joining. The dot11RegDomainsImplementedValue shall be set to Other when the STA is operating using Country element settings.

10.1.4.6 Operation of Supported Rates and Extended Supported Rates elements

Supported Rate and Extended Supported Rate information in Beacon and Probe Response management frames is used by STAs in order to avoid associating with a BSS if they do not support all the data rates in the BSSBasicRateSet parameter or all of the BSS membership requirements in the BSSMembershipSelectorSet parameter.

If the combined total of the number of rates in the OperationalRateSet parameter and the number of BSS membership selectors does not exceed eight, the Extended Supported Rates element is optional for inclusion in all of the frame types that include the Supported Rates element. If the combined total of the number of rates in the OperationalRateSet parameter and the number of BSS membership selectors exceeds eight, an Extended Supported Rates element shall be included to specify the remaining supported rates and BSS membership selectors in all of the frame types that include the Supported Rates element.

The Supported Rate information in Beacon and Probe Response management frames is delivered to the management entity in a STA via the BSSBasicRateSet parameter in the MLME-SCAN.confirm primitive. The BSS membership selector information in Beacon and Probe Response management frames is delivered to the management entity in a STA via the BSSMembershipSelectorSet parameter in the MLME-SCAN.confirm primitive. Together, these parameters are used by the management entity in a STA to avoid associating with a BSS if the STA cannot receive and transmit all the data rates in the BSSBasicRateSet parameter or does not support all of the features represented in the BSSMembershipSelectorSet parameter.

NOTE—A STA that was implemented before the existence of the BSSMembershipSelectorSet parameter interprets each BSS membership selector in the Supported Rates element that is contained in the BSSMembershipSelectorSet parameter of the transmitting STA as though it were a rate from the BSSBasicRateSet parameter. The value of each BSS membership selector does not match a rate that is known to the STA and, therefore, the management entity in the STA avoids associating with the BSS because it determines that the STA cannot receive or transmit at what appears to be a required rate.

A STA that is implemented after the existence of the BSSMembershipSelectorSet parameter includes each octet of the Supported Rates element that is encoded with the MSB (bit 7) equal to 1 and that it does not recognize as a rate in its BSSMembershipSelectorSet parameter. The STA then determines if it can support all of the features represented in its BSSMembershipSelectorSet parameter before attempting to join the network. If some BSSMembershipSelectorSet parameter values are not recognized by the STA, the STA does not attempt to join the network.

If the DSSS-OFDM bit is 1 in the transmitted Capability Information field of an MMPDU, then any supported rates transmitted in that frame that include rates that are common to both DSSS-OFDM and ERP-OFDM shall be interpreted by receiving and transmitting STA to indicate support for both DSSS-OFDM and ERP-OFDM at the indicated rate. However, if any of those rates are indicated as basic (a rate in the BSSBasicRateSet parameter), then the basic rate designation shall be interpreted by receiving and transmitting STA to apply only for the ERP-OFDM modulation and rate. If the PBCC bit is 1 in the transmitted capability field of an MMPDU, then any supported rates transmitted in that frame that include rates that are common to both PBCC and CCK shall be interpreted by receiving and transmitting STA to indicate support for both PBCC and CCK at the indicated rate. However, if any of those rates are indicated as basic, then the basic rate designation shall be interpreted by receiving and transmitting STA to apply only for the CCK modulation and rate. That is, if the rate is indicated as basic, the basic designation does not apply to DSSS-OFDM, PBCC, or ERP-PBCC.

10.1.5 Adjusting STA timers

In an infrastructure BSS, STAs shall adopt the TSF timer value in a Beacon frame or probe response coming from the AP in their BSS by using the algorithm in 10.1.3.7.

In response to an MLME-JOIN.request primitive, a STA joining an IBSS shall initialize its TSF timer to 0 and shall not transmit a Beacon frame or probe response until it hears a Beacon frame or probe response from a member of the IBSS with a matching SSID. Consequently, the STA joining an IBSS adopts the timer from the next Beacon frame or probe response from its IBSS.

All Beacon and Probe Response frames carry a Timestamp field. A STA receiving such a frame from another STA in an IBSS with the same SSID shall compare the Timestamp field with its own TSF time. If the Timestamp field of the received frame is later than its own TSF timer, the STA in the IBSS shall adopt all parameters contained in the Beacon frame according to the rule for that parameter found in the “IBSS adoption” column of the matching row of the BSSDescription table found in 6.3.3.3.2. Parameters adopted by a STA due to the receipt of a later timestamp shall not be changed by the STA except when adopting parameters due to a subsequently received Beacon frame with a later timestamp.

10.1.6 Timing synchronization for FH PHYs

NOTE—This subclause pertains only to STAs using an FH PHY.

The TSF described here provides a mechanism for STAs in an FH system to synchronize their transitions from one channel to another (their “hops”). Every STA shall maintain a table of all of the hopping sequences that are used in the system. All of the STAs in a BSS shall use the same hopping sequence. Each Beacon frame and probe response includes the channel synchronization information necessary to determine the hop pattern and timing for the BSS.

STAs shall use their TSF timer to time their frequency hopping. dot11CurrentDwellTime specifies the length of time that STAs shall stay on each frequency in their hopping sequence. Once STAs are synchronized, they have the same TSF timer value.

STAs in the BSS shall issue an appropriate PLME service primitive for the PHY in use to tune to the next frequency in the hopping sequence when

TSF timer MOD dot11CurrentDwellTime = 0

10.1.7 Terminating a BSS

An infrastructure BSS may be terminated at any time. A STA may cease support for an IBSS that it formed at any time. Upon receipt of an MLME-STOP.request primitive, a STA shall stop transmitting Beacon and Probe Response frames, and deauthenticate all associated STAs.

10.1.8 Supported rates and extended supported rates advertisement

A STA shall include rates from its OperationalRateSet parameter and BSS membership selectors from its BSSMembershipSelectorSet parameter in frames it transmits containing Supported Rates elements and Extended Supported Rates elements according to the rules described in this subclause.

For a STA supporting a combined total of eight or fewer data rates and BSS membership selectors, inclusion of the Extended Supported Rates element is optional in all of the frame types that include the Supported Rates element.

If the combined total of the number of rates in the OperationalRateSet parameter and the number of BSS membership selectors exceeds eight, then an Extended Supported Rate element shall be generated to specify the supported rates and BSS membership selectors that are not included in the Supported Rates element. If the BSSMembershipSelectorSet parameter contains at least one BSS membership selector, then at least one BSS membership selector value from the BSSMembershipSelectorSet parameter shall be included in the Supported Rates element.

NOTE—Inclusion of at least one BSS membership selector in the Supported Rates element causes a receiving STA that does not process the Extended Supported Rates element to still receive a BSS membership selector (which it considers to be a basic rate) that it does not support. Any values from the BSSMembershipSelectorSet parameter that are not transmitted in the Supported Rates element are transmitted in the Extended Supported Rates element.

10.2 Power management

10.2.1 Power management in an infrastructure network

10.2.1.1 General

STAs changing Power Management mode shall inform the AP of this fact using the Power Management bits within the Frame Control field of transmitted frames. A STA shall remain in its current Power Management mode until it informs the AP of a Power Management mode change via a frame exchange that includes an acknowledgment from the AP. Power Management mode shall not change during any single frame exchange sequence, as described in Annex G.

NOTE—This means the Power Management bit is the same for all MPDUs in an A-MPDU.

The AP shall buffer individually addressed BUs addressed to STAs operating in a PS mode. These buffered BUs shall be transmitted only at designated times.

If any STA in its BSS is in PS mode, the AP shall buffer all group addressed BUs and deliver them to all STAs immediately following the next Beacon frame containing a DTIM transmission.

The STAs that currently have buffered BUs within the AP are identified in a TIM, which shall be included as an element within all Beacon frames generated by the AP. A STA shall determine that a BU is buffered for it by receiving and interpreting a TIM.

STAs operating in PS modes shall periodically listen for Beacon frames, as determined by the STA's ListenInterval and the ReceiveDTIMs parameter in the MLME-POWERMGT.request primitive.

In a BSS operating under the DCF, or during the CP of a BSS using the PCF, upon determining that a BU is currently buffered in the AP, a STA operating in the PS mode shall transmit a short PS-Poll frame to the AP, which shall respond with the corresponding buffered BU immediately, or acknowledge the PS-Poll and respond with the corresponding BU at a later time. If the TIM indicating the buffered BU is sent during a CFP, a CF-Pollable STA operating in the PS mode does not send a PS-Poll frame, but remains active until the buffered BU is received (or the CFP ends).

A non-AP QoS STA may be in PS mode before the setup of DLS or Block Ack. Once DLS is set up, both of the QoS STAs associated with a DLS link suspend the PS mode and shall be awake. When a STA enters normal (non-APSD) PS mode, any downlink Block Ack agreement without an associated schedule is suspended for the duration of this PS mode. BUs for a TID without a schedule are sent using Normal Ack following a PS-poll as described in rest of 10.2.1. Uplink Block Ack, Block Acks for any TID with a schedule, and any Block Acks to APSD STA continue to operate normally.

A STA may use both WNM-Sleep mode and PS mode simultaneously.

10.2.1.2 STA Power Management modes

A STA may be in one of two different power states:

- *Awake:* STA is fully powered.
- *Doze:* STA is not able to transmit or receive and consumes very low power.

The manner in which a STA transitions between these two power states shall be determined by the STA's Power Management mode and reflected in dot11PowerManagementMode. These modes are summarized in Table 10-1.

Table 10-1—Power Management modes

Active mode or AM	STA may receive frames at any time. In Active mode, a STA shall be in the Awake state. A STA on the polling list of a PCF shall be in Active mode for the duration of the CFP.
PS	STA listens to selected Beacon frames (based upon the ListenInterval parameter of the MLME-ASSOCIATE.request or MLME-REASSOCIATE.request primitive) and sends PS-Poll frames to the AP if the TIM element in the most recent Beacon frame indicates an individually addressed BU is buffered for that STA. The AP shall transmit buffered individually addressed BUs to a PS STA only in response to a PS-Poll from that STA, during the CFP in the case of a CF-Pollable PS STA, or during a scheduled or unscheduled APSD service period for the STA. In PS mode, a STA shall be in the Doze state and shall enter the Awake state to receive selected Beacon frames, to receive group addressed transmissions following certain received Beacon frames, to transmit, and to await responses to transmitted PS-Poll frames or (for CF-Pollable STAs) to receive CF transmissions of buffered BUs.

The Power Management mode of a STA is selected by the PowerManagementMode parameter of the MLME-POWERMGT.request primitive. Once the STA updates its Power Management mode, the MLME shall issue an MLME-POWERMGT.confirm primitive indicating the success of the operation.

To change Power Management modes, a STA shall inform the AP through a successful frame exchange as described in Annex G initiated by the STA and that includes an ACK frame or a BlockAck frame from the AP. The Power Management subfield(s) in the Frame Control field of the frame(s) sent by the STA in this exchange indicates the Power Management mode that the STA shall adopt upon successful completion of the entire frame exchange, except where it is reserved (see 8.2.4.1.7). The Power Management bit shall be ignored in frame exchanges initiated by the AP. A non-AP STA shall not change power management mode using a frame exchange that does not receive an ACK or BlockAck from the AP, or using a BlockAckReq frame.

NOTE—A PS-Poll frame exchange does not necessarily result in an ACK from the AP, so a non-AP STA cannot change power management mode using a PS-Poll frame.

A STA that is changing from Doze to Awake in order to transmit shall perform CCA until a frame sequence is detected by which it can correctly set its NAV, or until a period of time equal to the ProbeDelay has transpired.

10.2.1.3 AP TIM transmissions

The TIM shall identify the STAs for which traffic is pending and buffered in the AP. This information is coded in a *partial virtual bitmap*, as described in 8.4.2.7. In addition, the TIM contains an indication whether group addressed traffic is pending. Every STA is assigned an AID by the AP as part of the association process. AID 0 (zero) is reserved to indicate the presence of buffered group addressed BUs. The AP shall identify those STAs for which it is prepared to deliver buffered BUs by setting bits in the TIM's partial virtual bitmap that correspond to the appropriate AIDs.

10.2.1.4 TIM types

Two different TIM types are distinguished: TIM and DTIM. After a DTIM, the AP shall transmit buffered group addressed BUs, before transmitting any individually addressed frames.

The AP shall transmit a TIM with every Beacon frame. Every dot11DTIMPeriod, a TIM of type *DTIM* is transmitted within a Beacon frame, rather than an ordinary TIM.

Figure 10-4 illustrates the AP and STA activity under the assumptions that no PCF is operating and that a DTIM is transmitted once every three TIMs. The top line in Figure 10-4 represents the time axis, with the beacon interval shown together with a DTIM Interval of three beacon intervals. The second line depicts AP activity. The AP schedules Beacon frames for transmission every beacon interval, but the Beacon frames may be delayed if there is traffic at the TBTT. This is indicated as "busy medium" on the second line. For the purposes of this figure, the important fact about Beacon frames is that they contain TIMs, some of which are DTIMs. Note that the second STA with ReceiveDTIMs equal to false does not power-on its receiver for all DTIMs.

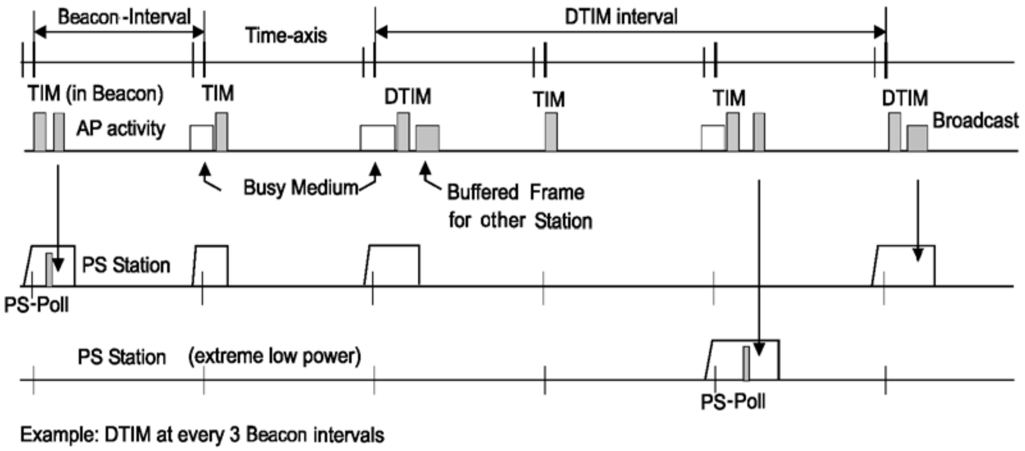


Figure 10-4—Infrastructure power management operation (no PCF operating)

The third and fourth lines in Figure 10-4 depict the activity of two STAs operating with different power management requirements. Both STAs power-on their receivers when they need to listen for a TIM. This is indicated as a ramp-up of the receiver power prior to the TBTT. The first STA, for example, powers up its receiver and receives a TIM in the first Beacon frame; that TIM indicates the presence of a buffered BU for the receiving STA. The receiving STA then generates a PS-Poll frame, which elicits the transmission of the buffered BU from the AP. Group addressed BUs are sent by the AP subsequent to the transmission of a Beacon frame containing a DTIM. The DTIM is indicated by the DTIM count field of the TIM element having a value of 0.

10.2.1.5 Power management with APSD

10.2.1.5.1 Power Management with APSD procedures

QoS APs capable of supporting automatic power save delivery (APSD) shall signal this capability through the use of the APSD subfield in the Capability Information field in Beacon, Probe Response, and (Re)Association Response management frames.

QoS STAs use the Power Management field in the Frame Control field of a frame to indicate whether it is in active or PS mode. As APSD is a mechanism for the delivery of downlink data and bufferable management frames to power-saving STAs, the frames transmitted by a STA in PS mode that is using APSD have the Power Management bit in the Frame Control field set to 1, thereby causing buffering to take place at the AP.

APSD defines two delivery mechanisms, namely *unscheduled APSD* (U-APSD) and *scheduled APSD* (S-APSD). STAs may use U-APSD to have some or all of their BUs delivered during unscheduled SPs. STAs may use S-APSD to schedule delivery of some or all of their BUs during scheduled SPs.

If there is no unscheduled SP in progress, the unscheduled SP begins when the AP receives a trigger frame from a STA, which is a QoS data or QoS Null frame using an AC the STA has configured to be trigger-enabled. An A-MPDU that contains one or more trigger frames acts as a trigger frame. An unscheduled SP ends after the AP has attempted to transmit at least one BU using a delivery-enabled AC and destined for the STA, but no more than the number indicated in the Max SP Length field of the QoS Capability element of the STA’s (Re)Association Request frame if the field has a nonzero value.

In order to configure an AP to deliver BUs during an unscheduled SP, a STA designates one or more of its ACs to be delivery-enabled and one or more of its AC to be trigger-enabled. A STA may configure an AP to use U-APSD using two methods. First, the STA may set individual U-APSD Flag bits in the QoS Info subfield of the QoS Capability element carried in (Re)Association Request frames. When a U-APSD Flag bit is 1, it indicates that the corresponding AC is both delivery- and trigger-enabled. When all four U-APSD Flag subfields are equal to 1 in (Re)Association Request frames, all the ACs associated with the STA are trigger- and delivery-enabled during (re)association. When all four U-APSD Flag subfields are equal to 0 in (Re)Association Request frames, none of the ACs associated with the STA is trigger- or delivery-enabled during (re)association.

NOTE—Bufferable MMPDUs are transmitted using AC_VO. Thus the AC of an MMPDU is, by definition, AC_VO.

Alternatively, the STA may designate one or more AC as trigger-enabled and one or more AC as delivery-enabled by sending an ADDTS Request frame per AC to the AP with the APSD subfield set to 1 and the Schedule subfield set to 0 in the TS Info field in the TSPEC element. APSD settings in a TSPEC request take precedence over the static U-APSD settings carried in the QoS Capability element. In other words, a TSPEC request overwrites any previous U-APSD setting of an AC. The request may be sent for ACs for which the ACM subfield is 0.

A STA may set an AC to be trigger- or delivery-enabled for its own use by setting up TSPECs with the APSD subfield set to 1 and the Schedule subfield set to 0 in the uplink or downlink direction, respectively. An uplink TSPEC plus a downlink TSPEC, or a bidirectional TSPEC with the APSD subfield equal to 1 and the Schedule subfield equal to 0, makes an AC both trigger- and delivery-enabled. An uplink TSPEC plus a downlink TSPEC, or a bidirectional TSPEC with the APSD and the Schedule subfields both equal to 0, makes an AC neither trigger- nor delivery-enabled.

A scheduled SP starts at fixed intervals of time specified in the Service Interval field. In order to use a scheduled SP for a TS when the access policy is controlled channel access, a STA shall send an ADDTS Request frame to the AP with the APSD subfield of the TS Info field in the TSPEC element set to 1. To use a scheduled SP for a TS for a AC when the access policy is contention-based channel access, a STA shall send an ADDTS Request frame to the AP with the APSD and Schedule subfields of the TS Info field in the TSPEC element both set to 1. If the APSD mechanism is supported by the AP and the AP accepts the corresponding ADDTS Request frame from the STA, the AP shall respond to the ADDTS Request frame with a response containing the Schedule element indicating that the requested service can be accommodated by the AP. The first scheduled SP starts when the lower order 4 octets of the TSF timer equals the value specified in the Service Start Time field. A STA using scheduled SP shall first wake up to receive downlink individually addressed BUs buffered and/or polls from the AP/HC. The STA shall wake up subsequently at a fixed time interval equal to the SI. The AP may modify the service start time by indicating so in the

Schedule element in a successful ADDTS Response frame (which is sent in response to an ADDTS Request frame) and in Schedule frames (which are sent at other times).

A scheduled SP begins at the scheduled wakeup time that corresponds to the SI and the service start time indicated in the Schedule element sent in response to a TSPEC. The STA shall wake up at a subsequent time when

(TSF – service start time) mod minimum SI = 0.

If scheduled services periods are supported in a BSS, a STA may use both unscheduled and scheduled APSD on different ACs at the same time. When a STA establishes scheduled delivery for an AC the AP shall not transmit BUs using that AC during an SP that is initiated by a trigger frame, and it shall not treat BUs using the AC that are received from the STA as trigger frames. The AP shall decline any ADDTS Request frame that indicates the use of both scheduled and unscheduled APSD to be used on the same AC at the same time.

APSD shall be used only to deliver individually addressed BUs. Group addressed BU delivery shall follow the frame delivery rules defined for group addressed BUs as defined in 10.2.1.7.

10.2.1.5.2 U-APSD Coexistence

A non-AP STA that uses U-APSD may not be able to receive all AP transmitted frames during the service period due to interference observed at the non-AP STA. Although the AP may not observe the same interference, it is able to determine that the frames were not received correctly by the non-AP STA. The U-APSD Coexistence capability enables the non-AP STA to indicate a requested transmission duration to the AP for use during U-APSD service periods. Use of the transmission duration enables the AP to transmit frames during the service period and improve the likelihood that the non-AP STA receives the frames when the non-AP STA is experiencing interference. The U-APSD Coexistence capability reduces the likelihood that the AP transmits frames during the service period that are not received successfully.

A STA that has a value of true for dot11MgmtOptionUAPSDCoexistenceActivated is defined as a STA that supports U-APSD Coexistence. A STA for which dot11MgmtOptionUAPSDCoexistenceActivated is true shall set the U-APSD Coexistence field of the Extended Capabilities element to 1, set to 0 otherwise.

A non-AP STA that is associated to an AP where both have previously advertised support for the U-APSD Coexistence capability may transmit ADDTS Request frames including the U-APSD Coexistence element to the AP.

The content of the ADDTS Request frame excluding the U-APSD Coexistence element is referred to in subsequent text as the Base ADDTS Request. Upon successful reception of the ADDTS Request frame, the AP shall process the content of the Base ADDTS Request frame as described in 10.2.1.5. If the AP determines that the Base ADDTS Request cannot be granted it shall respond as described in 10.2.1.5, without processing the U-APSD Coexistence element. If the AP determines the Base ADDTS Request can be granted, it shall process the U-APSD Coexistence element. If the AP supports transmission of frames during the U-APSD service period for the duration value specified in the U-APSD Coexistence element Interval/Duration field, it shall grant the ADDTS Request as described in 10.2.1.5. Otherwise, it shall deny the ADDTS Request.

If the AP has previously granted an ADDTS Request with U-APSD Coexistence, a non-AP STA that continues using QoS services provided by an ADDTS Request frame without U-APSD coexistence may terminate the use of U-APSD Coexistence by transmitting an ADDTS Request frame without the U-APSD Coexistence element. If the non-AP STA wants to terminate use of all QoS services provided by an ADDTS Request frame including U-APSD Coexistence, it may transmit a DELTS Request frame to the AP.

The non-AP STA may transmit multiple ADDTS Request frames to the AP where the last successfully received ADDTS Request frame will override any previously received ADDTS Request frame.

An AP that supports U-APSD Coexistence and accepts an ADDTS request limits the U-APSD Coexistence Service Period according to the parameters specified in the ADDTS frame U-APSD Coexistence element, and shall transmit frames to the requesting non-AP STA according to the rules in 9.2.4.2 and the following rules:

- If the non-AP STA specified a nonzero TSF 0 Offset value in the U-APSD Coexistence element, the AP should not transmit frames to the non-AP STA outside of the U-APSD Coexistence Service Period, which begins when the AP receives the U-APSD trigger frame and ends after the transmission period specified by the result of the following calculation:

  $\mathit{End\ of\ transmission\ period} = T + (\mathit{Interval} - ((T - \mathit{TSF\ 0\ Offset}) \bmod \mathit{Interval}))$

  where

  T is the time the U-APSD trigger frame was received at the AP
  Interval is the UAPSD Coexistence element Duration/Interval field value (see 8.4.2.93)

  or upon the successful transmission of a frame with EOSP bit set to 1, whichever is earlier.

- If the non-AP STA specified a TSF 0 Offset value of 0 in the U-APSD Coexistence element, the AP should not transmit frames to the non-AP STA outside of the U-APSD Coexistence Service Period, which begins when the AP receives a U-APSD trigger frame and ends after the transmission period specified by the result of the following calculation:

  $\mathit{End\ of\ transmission\ period} = T + \mathit{Duration}$

  where

  T is the time the U-APSD trigger frame was received at the AP
  Duration is the UAPSD Coexistence element Duration/Interval field value (see 8.4.2.93)

  or upon the successful transmission of a frame with EOSP bit set to 1, whichever is earlier.

Throughout the U-APSD Coexistence Service Period, the AP shall set the More bit to 1 if it has more frames to be transmitted and it can determine the frame might be received successfully before the service period expires.

The AP should set the EOSP bit to 1 in the frame that it expects to be the last frame transmitted to the non-AP STA during the U-APSD Coexistence duration. If the last frame expected to be transmitted cannot be successfully transmitted to the non-AP STA before the termination of the U-APSD SP, the AP should transmit a QoS null frame with the EOSP bit set to 1.

The non-AP STA may enter Doze State at the end of the U-APSD Coexistence Service Period.

10.2.1.6 AP operation during the CP

APs shall maintain a Power Management status for each currently associated STA that indicates in which Power Management mode the STA is currently operating. APs that implement and signal their support of APSD shall maintain an APSD and an access policy status for each currently associated STA that indicates whether the STA is presently using APSD and shall maintain the schedule (if any) for the STA. An AP shall, depending on the Power Management mode of the STA, temporarily buffer BUs destined to the STA. An AP implementing APSD shall, if a STA is using APSD and is in PS mode, temporarily buffer BUs destined to that STA. No BUs addressed directly to STAs operating in the Active mode shall be buffered for power management reasons.

The following rules describe operation during the CP:

a) BUs destined for PS STAs shall be temporarily buffered in the AP. The algorithm to manage this buffering is beyond the scope of this standard, with the exception that if the AP is QoS-enabled, it shall preserve the order of arrival of frames on a per-TID, per-STA basis.

b) Nonbufferable MMPDUs and BUs destined for STAs in the Active mode shall be directly transmitted to those STAs.

c) At every beacon interval, the AP shall assemble the partial virtual bitmap containing the buffer status per destination for STAs in the PS mode and shall send this out in the TIM field of the Beacon frame. At every beacon interval, the APSD-capable AP shall assemble the partial virtual bitmap containing the buffer status of nondelivery-enabled ACs (if there exists at least one nondelivery-enabled AC) per destination for STAs in PS mode and shall send this out in the TIM field of the Beacon frame. When all ACs are delivery-enabled, the APSD-capable AP shall assemble the partial virtual bitmap containing the buffer status for all ACs per destination. If FMS is enabled, the AP shall include the FMS Descriptor element in every Beacon frame. The FMS Descriptor element shall indicate all FMS group addressed frames that the AP buffers.

d) If a STA has set up a scheduled SP, it shall automatically wake up at each SP. Therefore, the APSD-capable AP shall transmit frames associated with admitted traffic with the APSD subfield equal to 1 in the TSPECs buffered for the STA during a scheduled SP. If the STA has set up to use unscheduled SPs, the AP shall buffer BUs using delivery-enabled ACs until it has received a trigger frame using a trigger-enabled AC from the non-AP STA, which indicates the start of an unscheduled SP. A trigger frame received by the AP from a STA that already has an unscheduled SP underway shall not trigger the start of a new unscheduled SP. The AP transmits BUs destined for the STA and using delivery-enabled ACs during an unscheduled SP. The bit for AID 0 (zero) in the bit map control field of the TIM element shall be set to 1 when group addressed traffic is buffered, according to 8.4.2.7.

e) If any associated STAs are in PS mode, all group addressed BUs except those with a service class of StrictlyOrdered shall be buffered.

f) When dot11MgmtOptionFMSActivated is false, the AP shall transmit all buffered group addressed BUs immediately after every DTIM.

When dot11MgmtOptionFMSActivated is true and the AP has established an FMS delivery interval for a multicast stream, the AP shall transmit all group addressed BUs belonging to particular FMS stream immediately after the DTIM that has the Current Count field value of the FMS Counter field set to 0 for that particular FMS stream.

The More Data field of each group addressed frame shall be set to indicate the presence of further buffered group addressed BUs. If the AP is unable to transmit all of the buffered group addressed BUs before the primary or secondary TBTT following the DTIM, the AP shall set the bit for AID 0 (zero) in the TIM element to 1 for a single BSSID or set the corresponding group address bit to 1 for multiple BSSIDs, as defined in 8.4.2.7, and when dot11MgmtOptionFMSActivated is true, shall set the appropriate bits in the FMS Descriptor element as described in 8.4.2.77 to indicate for which group addresses there are still buffered BUs, until all buffered group addressed BUs have been transmitted. When the AP transmits an STBC DTIM or TIM Beacon frame, the AP shall retransmit all group addressed BUs that were transmitted following the non-STBC DTIM or TIM Beacon frame except that they are transmitted using the basic STBC MCS. It may be the case that a complete set of buffered group addressed BUs is sent over a period of time during which non-STBC and STBC transmissions are interleaved, but the transition from non-STBC group addressed transmissions to STBC group addressed transmissions shall be preceded by the transmission of an STBC Beacon frame and the transition from STBC group addressed transmissions to non-STBC group addressed transmissions shall be preceded by the transmission of a non-STBC Beacon frame.

g) A single buffered BU for a STA in the PS mode shall be forwarded to the STA after a PS-Poll has been received from that STA. For a STA using U-APSD, the AP transmits one BU destined for the STA from any AC that is not delivery-enabled in response to PS-Poll from the STA. When all ACs associated with the STA are delivery-enabled, AP transmits one BU from the highest priority AC. The AP can respond with either an immediate data or management frame or with an ACK, while delaying the responding data or management frame.

For a STA in PS mode and not using U-APSD, the More Data field of the response data or management frame shall be set to indicate the presence of further buffered BUs for the polling STA. For a STA using U-APSD, the More Data field shall be set to indicate the presence of further buffered BUs that do not use delivery-enabled ACs. When all ACs associated with the STA are delivery-enabled, the More Data field shall be set to indicate the presence of further buffered BUs using delivery-enabled ACs. If there are buffered BUs to transmit to the STA, the AP may set the More Data bit in a QoS +CF-Ack frame to 1, in response to a QoS data frame to indicate that it has one or more pending BUs buffered for the PS STA identified by the RA in the QoS +CF-Ack frame. An AP may also set the More Data bit in an ACK frame to 1 in response to a QoS data frame to indicate that it has one or more pending BUs buffered for the PS STA identified by the RA in the ACK frame, if that PS STA has set the More Data Ack subfield in the QoS Capability element to 1.

Further PS-Poll frames from the same STA shall be acknowledged and ignored until the BU has either been successfully delivered or presumed failed due to maximum retries being exceeded. This prevents a retried PS-Poll from being treated as a new request to deliver a buffered BU.

h) At each scheduled APSD SP for a STA, the APSD-capable AP (i.e., an AP for which dot11APSDOptionImplemented is true) shall attempt to transmit at least one BU, using admitted TSPECs with the APSD and Schedule subfields both set to 1, that are destined for the STA. At each unscheduled SP for a STA, the AP shall attempt to transmit at least one BU, but no more than the value specified in the Max SP Length field in the QoS Capability element from delivery-enabled ACs, that are destined for the STA.

The More Data bit of the individually addressed data or bufferable management frame using delivery-enabled ACs and destined for that STA indicates that more BUs are buffered for the delivery-enabled ACs. The More Data bit equal to 1 in data or bufferable management frames using nondelivery-enabled ACs and destined for that STA indicates that more BUs are buffered for the nondelivery-enabled ACs. For all frames except for the final frame of the SP, the EOSP subfield of the QoS Control field of the QoS data frame shall be set to 0 to indicate the continuation of the SP. An AP may also set the More Data bit to 1 in a QoS +CF-Ack frame in response to a QoS data frame to indicate that it has one or more pending BUs buffered for the target STA identified by the RA in the QoS +CF-Ack frame. If the QoS data frame is using a delivery-enabled AC, the More Data bit in the QoS +CF-Ack frame indicates more BUs for all delivery-enabled ACs. If the QoS data frame is not using a delivery-enabled AC, the More Data bit in the QoS +CF-Ack frame indicates more BUs for all ACs that are not delivery-enabled.

The AP considers an APSD STA to be in Awake state after it has sent a QoS +CF-Ack frame, with the EOSP subfield in the QoS Control field equal to 0, to the APSD STA. If necessary, the AP may generate an extra QoS Null frame, with the EOSP set to 1. When the AP has transmitted an individually addressed frame to the STA with the EOSP subfield set to 1 during the SP except for retransmissions of that frame, the AP shall not transmit any more frames to that STA using this mechanism until the next SP. The AP shall set the EOSP subfield to 1 to indicate the end of the SP in APSD.

i) If the AP does not receive an acknowledgment to an individually addressed data or bufferable management frame sent to a STA in PS mode following receipt of a PS-Poll from that STA, it may retransmit the frame for at most the lesser of the maximum retry limit and dot11QAPMissingAckRetryLimit times before the next Beacon frame, but it shall retransmit that frame at least once before the next Beacon frame, time permitting and subject to its appropriate lifetime limit. If an acknowledgment to the retransmission is not received, it may wait until after the next Beacon frame to further retransmit that frame subject to its appropriate lifetime limit.

j) If the AP does not receive an acknowledgment to an individually addressed data frame containing all or part of an MSDU or A-MSDU sent with the EOSP subfield equal to 1, it shall retransmit that frame at least once within the same SP, subject to applicable retry or lifetime limit. The maximum number of retransmissions within the same SP is the lesser of the maximum retry limit and dot11QAPMissingAckRetryLimit. If an acknowledgment to the retransmission of this last frame in the same SP is not received, it may wait until the next SP to further retransmit that frame, subject to its applicable retry or lifetime limit.

NOTE—An AP that transmits an A-MPDU containing data MPDUs in which the EOSP field is set to 1 and that receives a BlockAck that does not acknowledge all of those MPDUs, cannot transmit any missed data MPDUs within the current service period because the destination STA might now be asleep.

k) An AP may delete buffered BUs for implementation-dependent reasons, including the use of an aging function and availability of buffers. The AP may base the aging function on the Listen Interval specified by the STA in the (Re)Association Request frame or the WNM-Sleep Interval specified by the non-AP STA in the WNM-Sleep Mode Request frame.

l) When an AP is informed that a STA has changed to the Active mode, then the AP shall send buffered BUs (if any exist) to that STA without waiting for a PS-Poll. When an AP is informed that an APSD-capable STA is not using APSD, then the AP shall send buffered BUs (if any exist) to that STA according to the rules corresponding to the current PS mode of the STA.

10.2.1.7 AP operation during the CFP

APs shall maintain a Power Management status for each currently associated CF-Pollable STA that indicates in which Power Management mode the STA is currently operating. An AP shall, for STAs in PS mode, temporarily buffer BUs addressed to the STA.

The following rules describe operation during the CFP:

a) BUs destined for PS STAs shall be temporarily buffered in the AP. The algorithm to manage this buffering is beyond the scope of this standard.

b) Nonbufferable MMPDUs and all BUs destined to STAs in the Active mode shall be transmitted as defined in Clause 9.

c) Prior to every CFP, and at each beacon interval within the CFP, the AP shall assemble the partial virtual bitmap containing the buffer status per destination for STAs in the PS mode, set to 1 the bits in the partial virtual bitmap for STAs the PC is intending to poll during this CFP, and shall send this out in the TIM field of the DTIM. The bit for AID 0 (zero) in the Bit Map Control field of the TIM element shall be set to 1 when group addressed traffic is buffered, according to 8.4.2.7.

d) All group addressed MSDUs except those with a service class of StrictlyOrdered shall be buffered if any associated STAs are in the PS mode, regardless of whether those STAs are CF-Pollable.

e) When dot11MgmtOptionFMSActivated is false, the AP shall transmit all buffered group addressed BUs immediately after every DTIM (Beacon frame with DTIM Count field of the TIM element equal to 0).

When dot11MgmtOptionFMSActivated is true and the AP has set up an FMS delivery interval for a multicast stream, the AP shall send all group addressed BUs belonging to a particular FMS stream immediately after the DTIM with the Current Count field value of the FMS Counter field set to 0 for that particular FMS stream.

The More Data field shall be set to 1 in the headers of all but the final frame containing one of these buffered group addressed BUs to indicate the presence of further buffered group addressed BUs. If the AP is unable to transmit all of the buffered group addressed BUs before the non-STBC or STBC TBTT following the DTIM, the AP shall set the bit for AID 0 (zero) in the TIM element to 1 for a single BSSID or set the corresponding group addressed bit to 1 for multiple BSSIDs, as defined in 8.4.2.7, and when dot11MgmtOptionFMSActivated is true, shall set the appropriate bits in the FMS Descriptor element as described in 8.4.2.77 to indicate for which group addresses there are still buffered BUs, until all buffered group addressed BUs have been transmitted. When the AP transmits an STBC DTIM or TIM Beacon frame, the AP shall retransmit all group addressed BUs that were transmitted following the non-STBC DTIM or TIM Beacon frame except that they are transmitted using the basic STBC MCS. It may be the case that a complete set of buffered group addressed BUs is sent over a period of time during which non-STBC and STBC transmissions are interleaved, but the transition from non-STBC group addressed transmissions to STBC group addressed transmissions shall be preceded by the transmission of a STBC Beacon frame and the transition from STBC group addressed transmissions to non-STBC group addressed transmissions shall be preceded by the transmission of a non-STBC Beacon frame.

f) Buffered BUs for STAs in the PS mode shall be forwarded to the CF-Pollable STAs under control of the PC. Transmission of these buffered BUs as well as CF-Polls to STAs in the PS mode that were indicated in the DTIM in accordance with paragraph c) of this subclause shall begin immediately after transmission of buffered group addressed frames (if any), and shall occur in order by increasing AID of CF-Pollable STAs. A CF-Pollable STA for which the TIM element of the most recent Beacon frame indicated buffered BUs shall be in the Awake state at least until the receipt of an individually addressed frame from the AP in which the Frame Control field does not indicate the existence of more buffered BUs. After acknowledging the last of the buffered BUs, the CF-Pollable STA operating in the PS mode may enter the Doze state until the next DTIM is expected.

g) An AP shall have an aging function to delete pending traffic buffered for an excessive time period. The exact specification of the aging function is beyond the scope of this standard.

h) When an AP detects that a CF-Pollable STA has changed from the PS mode to the Active mode, then the AP shall queue any buffered BUs addressed to that STA for transmission to that CF-Pollable STA as directed by the AP's PC.

10.2.1.8 Receive operation for STAs in PS mode during the CP

A STA in PS mode shall operate as follows to receive a BU from the AP when no PC is operating and during the CP when a PC is operating.

The following rules describe operation of a STA in PS mode during the CP:

a) The STA shall wake up early enough to be able to receive the first Beacon frame scheduled for transmission at the time corresponding to the last TBTT plus the ListenInterval.

b) When the STA detects that the bit corresponding to its AID is 1 in the TIM, the STA shall issue a PS-Poll, or a trigger frame if the STA is using U-APSD and all ACs are delivery-enabled, to retrieve the buffered BU. The PS-Poll or trigger shall be transmitted after a random delay uniformly distributed between zero and aCWmin slots following a DIFS.

c) The STA shall remain in the Awake state until it receives the BU in response to its poll or it receives another Beacon frame whose TIM indicates that the AP does not have any BUs buffered for this STA. If the bit corresponding to the STA's AID is 1 in the subsequent TIM, the STA shall issue another PS-Poll to retrieve the buffered BU. When a STA that is using U-APSD and has all ACs delivery-enabled detects that the bit corresponding to its AID is 1 in the TIM, the STA shall issue a trigger frame or a PS-Poll frame to retrieve the buffered BU.

d) If the More Data field in the received data or bufferable management frame indicates that more traffic for that STA is buffered, the STA, at its convenience, shall Poll until no more BUs are buffered for that STA.

e) When dot11MgmtOptionFMSActivated is false and ReceiveDTIMs is true, the STA shall wake up early enough to be able to receive either every non-STBC DTIM or every STBC DTIM sent by the AP of the BSS.

   When dot11MgmtOptionFMSActivated is true and ReceiveDTIMs is true and the STA has been granted by the AP an alternate delivery interval for a multicast stream, the STA shall wake up before the non-STBC DTIM or STBC DTIM having Current Count of FMS Counter field set to 0 for that particular FMS stream.

   A STA that stays awake to receive group addressed BUs shall elect to receive all group addressed non-STBC transmissions or all group addressed STBC transmissions and remain awake until the

More Data field of the appropriate type (non-STBC or STBC) of group addressed BUs indicates there are no further buffered group addressed BUs of that type, or until a TIM is received indicating there are no more buffered group addressed BUs of that type, or until an FMS Descriptor element is received indicating that there are no further buffered group addressed BUs for which the STA has previously received an FMS Response element in a frame that has a value in address1 that matches its MAC address or that has an address1 value that is a group address corresponding to a group of which it is a member and that was transmitted by the AP with which it is associated and which had an Element Status value in FMS Status subelement of Accept. If a STA receives a QoS +CF-Ack frame from its AP with the More Data bit equal to 1, then the STA shall operate exactly as if it received a TIM with its AID bit equal to 1. If a STA has set the More Data Ack subfield in QoS Capability element to 1, then if it receives an ACK frame from its AP with the More Data bit equal to 1, the STA shall operate exactly as if it received a TIM with its AID bit equal to 1. For example, a STA that is using the PS-Poll delivery method shall issue a PS-Poll frame to retrieve a buffered BU. See also 9.3.6.

10.2.1.9 Receive operation for STAs in PS mode during the CFP

A STA in PS mode that is associated as CF-Pollable shall operate as follows in a BSS with an active PC to receive BUs from the AP during the CFP:

a) The STA shall enter the Awake state so as to receive the Beacon frame (which contains a DTIM) at the start of each CFP.

b) To receive group addressed BUs, the STA shall wake up early enough to be able to receive either every non-STBC DTIM or every STBC DTIM that may be sent during the CFP. A STA receiving group addressed BUs shall elect to receive all group addressed non-STBC transmissions or all group addressed STBC transmissions and remain awake until the More Data field of the frames containing the group addressed BUs indicates there are no further buffered group addressed BUs of that type, or until a TIM is received indicating there are no more group addressed BUs of that type buffered or until an FMS Descriptor element is received indicating that there are no further buffered group addressed BUs for which the STA has previously received an FMS Response element in a frame that has an address1 value that matches its MAC address or that has an address1 value that is a group address corresponding to a group of which it is a member and that was transmitted by the AP with which it is associated and which had an Element Status value in FMS Status subelement of Accept. See also 9.3.6.

c) When the STA detects that the bit corresponding to its AID is 1 in the DTIM at the start of the CFP (or in a subsequent TIM during the CFP), the STA shall remain in the Awake state for at least that portion of the CFP through the time that the STA receives an individually addressed BU from the AP carried in a frame with the More Data field in the Frame Control field indicating that no further traffic is buffered.

d) If the More Data field in the Frame Control field of the last data or bufferable management frame received from the AP indicates that more traffic for the STA is buffered, then, when the CFP ends, the STA may remain in the Awake state and transmit PS-Poll frames during the CP to request the delivery of additional buffered BUs, or may enter the Doze state during the CP (except at TBTTs for DTIMs expected during the CP), awaiting the start of the next CFP.

10.2.1.10 Receive operation using APSD

A STA using APSD shall operate as follows to receive a BU from the AP:

a) If a scheduled SP has been set up, the STA wakes up at its scheduled start time. (The STA shall wake up early enough to receive transmissions at the scheduled SP.)

b) If the STA is initiating an unscheduled SP, the STA wakes up and transmits a trigger frame to the AP. When one or more ACs are not delivery-enabled, the STA may retrieve BUs using those ACs by sending PS-Poll frames to the AP.

c) The STA shall remain awake until it receives a QoS data frame or QoS Null frame addressed to it, with the EOSP subfield in the QoS Control field equal to 1.

d) The STA may send additional PS-Poll frames if the More Data subfield is 1 in downlink individually addressed data or bufferable management frames that do not use any delivery-enabled ACs. The STA may send additional trigger frames if the More Data subfield is 1 in downlink individually addressed data or bufferable management frames that use delivery-enabled ACs.

10.2.1.11 STAs operating in the Active mode

A STA operating in this mode shall have its receiver activated continuously; such STAs do not need to interpret the TIM elements in Beacon frames.

10.2.1.12 AP aging function

Any AP aging function shall not cause the buffered BU to be discarded after any period that is shorter than the ListenInterval of the STA for which the BUs are buffered. The exact specification of the aging function is beyond the scope of this standard.

NOTE—This aging function is independent of (i.e., in addition to) other causes of MSDU discard within the MAC, such as due to the operation of a per-TS MSDU lifetime, or related to dot11QAPEDCATableMSDULifetime.

10.2.1.13 PSMP power management

An AP transmits a PSMP frame containing a schedule only for STAs that are awake.

A STA with an established PSMP session (see 10.4.6) shall be awake at the start of the session's SP and shall remain awake until the end of the SP unless permitted to return to sleep as described in this subclause.

NOTE—A STA in power save mode can also be determined to be awake following receipt of a trigger frame according to the operation of the U-APSD protocol (as defined in 10.2.1.5), following receipt of a PS-Poll frame (as defined in 10.2.1.8), or following a DTIM Beacon (as defined in 10.2.1.8).

The AP may signal the end of the SP for all awake associated PSMP-capable STAs by setting the More PSMP field to 0 or by sending CF-End frame instead of the next PSMP frame.

NOTE 1—The AP can also signal the end of an SP on a per-STA basis using the EOSP field set to 1 in the QoS Control field, as defined in 8.2.4.5.3 and 10.2.1.6. This field remains set to 1 for any retransmissions of the same frame, and no more new frames are sent to this particular STA in the current SP.

NOTE 2—If a STA is awake at the start of a scheduled PSMP session, the operation of the More Data field in the Frame Control field and the TIM element are defined by the S-APSD rules in 10.2.1.5, 10.2.1.6, and 10.2.1.10.

A STA shall wake up at the start of the next PSMP frame if the More PSMP field is equal to 1 in the current PSMP frame, unless the STA has been permitted to return to sleep through the reception of a frame addressed to it with the EOSP field equal to 1 or the maximum SP interval has elapsed.

Within a PSMP sequence, a PPDU containing MPDUs addressed to a STA shall not start after expiry of the STA’s PSMP-DTT. A STA completes the reception of any PPDU that starts before the end of the PSMP-DTT. If no frames addressed to a STA begin within a PSMP-DTT, it can assume that no frame addressed to it will arrive during this PSMP sequence.

The STA shall be awake to receive at the start of the PSMP-DTT determined from a STA_INFO field that has the STA_INFO Type subfield equal to 2 and the AID field matching the STA’s AID where the PSMP-DTT Duration subfield is not equal to 0.

10.2.1.14 TDLS Peer Power Save Mode

TDLS Peer Power Save Mode (TDLS Peer PSM) is a power save mechanism that can be used between TDLS peer STAs, and which is based on a periodic wakeup schedule. A STA supports TDLS Peer PSM if dot11TDLSPeerPSMActivated is true. A STA supporting this capability may indicate support through any TDLS Setup Request frame or TDLS Setup Response frame. A STA indicating this support shall signal this by setting the TDLS Peer PSM Support subfield in the Extended Capabilities element included in the TDLS Setup Request frame or TDLS Setup Response frame to 1. A station that signals support for this capability is capable of acting in both the TDLS Peer PSM initiator and the TDLS Peer PSM responder role.

A STA that intends to enter TDLS Peer PSM (TDLS Peer PSM initiator) shall send a TDLS Peer PSM Request frame to the TDLS peer STA (TDLS Peer PSM responder), including a proposed periodic Wakeup Schedule. A TDLS Peer PSM Request frame shall not be transmitted to a STA that did not indicate support for TDLS Peer PSM. When the TDLS Peer PSM responder accepts the proposed Wakeup Schedule, it shall respond with a TDLS Peer PSM Response frame indicating status code 0 (“Successful”). Otherwise, the TDLS Peer PSM responder shall respond with a TDLS Peer PSM Response frame indicating the appropriate status code for rejecting the schedule. An alternative schedule shall be included in the TDLS Peer PSM Response frame when the status code is equal to 2 (“TDLS Wakeup Schedule rejected but alternative schedule provided”). The alternative schedule may be used by the TDLS Peer PSM initiator to generate a new TDLS Peer PSM Request frame. After successfully transmitting or receiving a TDLS Peer PSM Response frame indicating status code 0 (“Successful”), the TDLS Peer PSM initiator and TDLS Peer PSM responder have established a periodic wakeup schedule between them. The wakeup schedule remains valid until one of the following occurs:

- The TDLS direct link is torn down;
- The STAs explicitly update the existing wakeup schedule; or
- No MPDUs containing data have been exchanged for Idle Count consecutive Awake Windows.

A STA transmitting a TDLS Peer PSM Request frame shall remain in the wake state until it received the corresponding TDLS Peer PSM Response frame. A TDLS Peer PSM Request frame may be transmitted via the AP path or via the direct path (which is up to the implementer to decide). A TDLS Peer PSM Response frame shall be transmitted over the direct path.

The timing of the periodic schedule of the TDLS Peer PSM Awake Windows is based on the Offset field, the Interval field, the Awake Window Slots field, and the Maximum Awake Window Duration field of the Wakeup Schedule element that is contained in the TDLS Peer PSM Setup Request frame that established TDLS Peer PSM operation on the link.

Awake Windows begin at TSF values that satisfy the equation TSF mod Interval = Offset. The interval between the start of two successive Awake Windows is equal to the time in microseconds of the Interval field. The periodic wakeup schedule may be unrelated to the target beacon transmission time (TBTT) or the beacon interval.

Awake Windows end when the Awake Window Slot Counter reaches 0 or when the Maximum Awake Window Duration has been reached, whichever comes first.

The Awake Window Slot Counter counts down backoff slots that are determined using AIFS[AC_BE] in the same manner that normal backoff slots are determined according to 9.19.2.5.

The initial value of the Awake Window Slot Counter at the start of the Awake Window shall be equal to the value in the Awake Window Slots field of the Wakeup Schedule element that is contained in the TDLS Peer PSM Setup Request frame that established TDLS Peer PSM operation on the link.

The Awake Window Slot Counter begins counting at the beginning of the Awake Window and stops counting when it reaches 0.

A value of 0 in the Maximum Awake Window Duration field of the Wakeup Schedule element that is contained in the TDLS Peer PSM Setup Request frame that established TDLS Peer PSM operation on the link means that the end of the Awake Window duration is determined only by the Awake Window Slot Counter.

A value of 0 in the Awake Window Slots field of the Wakeup Schedule element that is contained in the TDLS Peer PSM Setup Request frame that established TDLS Peer PSM operation on the link means that the duration of the Awake Window is determined only by the Maximum Awake Window Duration.

The Maximum Awake Window Duration field and the Awake Window Slots field shall not both be set to 0 in a TDLS Peer PSM Setup Request frame.

The Awake Window Slots field in a TDLS Peer PSM Setup Request frame should be set to a value that is larger than CWmin[AC_BE].

A TDLS Peer PSM service period is a contiguous period of time during which one or more individually addressed frames are transmitted between two TDLS peer STAs when at least one STA employs TDLS Peer PSM. A TDLS Peer PSM service period may be initiated during an Awake Window. A TDLS peer STA in power save mode may enter a doze state when it has successfully transmitted to and received from the corresponding TDLS peer STA in power save mode a QoS frame with the end of service period (EOSP) subfield equal to 1, ending the TDLS Peer PSM service period. A TDLS peer STA in power save mode may enter a doze state when it has successfully received from the corresponding TDLS peer STA in active mode a QoS frame with the EOSP subfield equal to 1.

Either STA may update an existing schedule by initiating a TDLS Peer PSM Request/Response exchange. If the TDLS Peer PSM Response frame indicates status code 0 (“Successful”), a new wakeup schedule is established for the TDLS direct link. Otherwise, the existing schedule still applies. The new schedule takes effect after the termination of the current TDLS Peer PSM service period.

After the successful PSM setup, a STA informs its TDLS peer STA that it will enter power save mode per direct link by setting the Power Management field to 1 in an MPDU requiring acknowledgement. The STA enters power save mode after successful transmission of the MPDU. The power save status on one direct link is independent of the power save status on other links (direct or with the AP) the STA may have.

If a TDLS peer STA enters power save mode when a Wakeup Schedule is active, it shall be awake at the beginning of each scheduled periodic Awake Window, and stay awake for the duration of the Awake Window or until the end of a TDLS Peer PSM service period. Otherwise, it may enter a doze state, depending on the current requirements to be awake, imposed by other links. A TDLS peer STA that did not enter power save mode shall remain in the awake state.

When both TDLS peer STAs set the More Data ACK subfield in their QoS Capability element to 1, then the More Data field inside an ACK frame set to 0 shall have the same function as the EOSP subfield inside a QoS frame set to 1. Transmission of an ACK frame with the More Data subfield set to 0 under these conditions is equivalent to a successful transmission of a QoS frame with the EOSP subfield set to 1.

When waking up at the beginning of an Awake Window, if a STA has no buffered BU to send to a TDLS Peer STA that had the More Data Ack subfield in its QoS Capability element equal to 1 during the TDLS setup exchange, the TDLS STA may send a QoS-Null frame with the EOSP subfield of the QoS Control field set to 1, and the More Data subfield of the Frame Control field set to 0. If the TDLS peer STA that is the recipient of this QoS-Null frame also has no buffered BU to deliver either, and it had the More Data Ack subfield in its QoS Capability element equal to 1 during the TDLS setup exchange, then the TDLS peer STA shall respond with an ACK frame that has the More Data subfield set to 0. The STA may discard the QoS-Null frame if it has not been successfully transmitted at the end of the Awake Window. Before the successful transmission of the QoS-Null frame, if a Data or QoS-Null frame with an EOSP subfield equal to 1 is received from the TDLS peer STA the STA may cancel the pending transmission of the QoS-Null frame after the transmission of an ACK response frame with the More Data subfield set to 0.

To keep track of the connectivity over the direct link and to maintain the wakeup schedule, TDLS peer STAs may start an acknowledged frame exchange at least once per Idle Count consecutive Awake Windows, as a keepalive. For instance a QoS-Null frame may be used as a keepalive frame. When a TDLS Peer PSM Response frame was successfully transmitted or received and no subsequent TDLS Peer PSM service period has started for Idle Count consecutive wakeup periods, the TDLS peer STAs shall delete the wakeup schedule for this link, which means that the related periodic wakeup no longer occurs (i.e., the TDLS peer STAs no longer have to wake up during this period) and that a wakeup schedule no longer exists for this link. When traffic arrives at a TDLS peer STA in TDLS Peer PSM mode for a link with no existing wakeup schedule, the STA shall send a TDLS Peer PSM Request frame through the AP path to the TDLS peer STA to activate a new wakeup schedule. When both TDLS peer STAs enter active mode while a wakeup schedule is active, no more TDLS Peer PSM service periods will occur, causing the wakeup schedule to be deleted.

If a TDLS peer STA does not receive an acknowledgment to an individually addressed QoS frame sent with the EOSP subfield equal to 1 that terminates a TDLS Peer PSM service period, it shall retransmit that frame at least once within the same service period, subject to the applicable retry or lifetime limit. The maximum number of retransmissions within the same service period is the lesser of the maximum retry limit and dot11TDLSPeerSTAMissingAckRetryLimit. If an acknowledgment to the retransmission of this last frame in the same service period is not received, the TDLS peer STA may wait until the next Awake Window to further retransmit that frame, subject to its applicable retry or lifetime limit. When the TDLS peer STA has transmitted an individually addressed frame that terminates a TDLS Peer PSM service period then, except for retransmissions of that frame, the TDLS peer STA shall not transmit any more frames to the TDLS peer STA until the next Awake Window.

A TDLS peer STA that has an active Wakeup Schedule shall not decrement a backoff count outside the Awake Windows, if that backoff precedes a frame that is destined for transmission on the related TDLS direct link.

At the start of an Awake Window, the backoff procedure shall be invoked at an EDCAF if there is a frame available for transmission at that EDCAF, and the backoff timer for that EDCAF has been 0 for at least one backoff slot.

Outside of its Awake Windows, and during Awake Windows when on the base channel, a TDLS peer STA can engage in communications with the AP.

10.2.1.15 TDLS Peer U-APSD

10.2.1.15.1 General

A STA supports the TDLS Peer U-APSD Buffer STA function if dot11TDLSPeerUAPSDBufferSTAActivated is true. A STA supporting this capability may indicate support through any TDLS Setup Request frame or TDLS Setup Response frame. A STA indicates support by setting the TDLS Peer U-APSD Buffer STA Support subfield in the Extended Capabilities element included in the TDLS Setup Request frame or TDLS Setup Response frame to 1. Support for the TDLS Peer U-APSD Buffer STA function means that the STA has the capability to buffer BUs destined to the TPU sleep STA, and to deliver them during unscheduled service periods.

To operate as the TPU Sleep STA in TDLS Peer U-APSD, a STA shall configure its TDLS Peer U-APSD capable TDLS peer STA by setting one or more U-APSD Flag subfields inside the QoS Info subfield of the QoS Capability element carried in a TDLS Setup Response frame to 1, or by setting one or more U-APSD Flag subfields inside the QoS Info subfield of the EDCA Parameter Set element carried in a TDLS Setup Confirm frame to 1.

A STA that configured TDLS Peer U-APSD at a TDLS peer STA enters power save mode on a TDLS direct link after the successful transmission to the TDLS peer STA over the direct link of an acknowledged MPDU with the Power Management field equal to 1. The STA that transmitted the frame with the Power Management field equal to 1 is then referred to as a TPU sleep STA. The STA that received the frame with the Power Management field equal to 1 is referred to as a TPU buffer STA. A TPU sleep STA may be a TPU buffer STA at the same time and on the same link, by sending a frame to the TDLS peer STA with the Power Management subfield of the Frame Control field set to 1 (this transmission will be preceded by the transmission of a Peer Traffic Indication frame and the subsequent receipt of a trigger frame that starts a service period). The power save status on one direct link is independent of the power save status on other links (direct or with the AP) the STA may have.

The procedure to trigger and terminate an unscheduled SP between TPU buffer STA and a TPU sleep STA are described in 10.2.1.5 and 10.2.1.6, where the TPU buffer STA shall take the role of the AP and the TPU sleep STA shall take the role of the non-AP STA using U-APSD, except that a frame with the EOSP field equal to 1 shall not act as a trigger frame.

10.2.1.15.2 TDLS Peer U-APSD Behavior at the TPU buffer STA

BUs at a TPU buffer STA destined for a TPU sleep STA shall be temporarily buffered at the TPU buffer STA. The algorithm to manage this buffering is beyond the scope of this standard, except that the TPU buffer STA shall preserve the order of buffered BUs on a per-TID, per-STA basis.

A TPU buffer STA shall transmit an individually addressed TDLS Peer Traffic Indication frame to a TPU sleep STA, through the AP, if and only if all of the following conditions are met:

- A BU destined for a TPU sleep STA was placed into a buffer at the TPU buffer STA;
- The buffer into which the BU was placed contained no other frames with the same RA; and
- One or more periods of dot11TDLSPeerUAPSDIndicationWindow beacon intervals have expired after the last service period.

The TDLS Peer Traffic Indication frame shall be transmitted through the AP path.

The transmitted TDLS Peer Traffic Indication frame shall indicate the nonempty AC(s), by setting the corresponding AC Traffic Available subfield of the TDLS Peer Traffic Indication frame to 1.

A PTI Control element may be included in the TDLS Peer Traffic Indication frame, to allow the TPU sleep STA to not start a service period when the indicated traffic has already been received by the TPU sleep STA.

The TID field contained in the PTI Control element (if included) shall be set to the TID of the latest MPDU that has been transmitted over the TDLS direct link to the TPU sleep STA that is the destination of the TDLS Peer Traffic Indication frame that contains the PTI Control element.

The Sequence Control field contained in the PTI Control element (if included) shall be set to the sequence number of the latest MPDU that has been transmitted over the TDLS direct link to the TPU sleep STA that is the destination of the TDLS Peer Traffic Indication frame that contains the PTI Control element.

After transmitting a TDLS Peer Traffic Indication frame without a PTI Control element, the TPU buffer STA shall stay awake at least until the corresponding or a subsequent TDLS Peer Traffic Response frame is received.

After transmitting a TDLS Peer Traffic Indication frame without a PTI Control element, the TPU buffer STA shall stay awake at least until the MPDU following the MPDU indicated in the Sequence Control field of the PTI Control element is successfully transmitted.

When no corresponding TDLS Peer Traffic Response frame has been received within dot11TDLSResponseTimeout after sending a TDLS Peer Traffic Indication frame, the STA shall tear down the direct link.

10.2.1.15.3 TDLS Peer U-APSD Behavior at the TPU sleep STA

When a TPU sleep STA receives a TDLS Peer Traffic Indication frame without a PTI Control element, the TPU sleep STA shall initiate a service period with the TPU buffer STA during which it shall transmit at least a TDLS Peer Traffic Response frame. The TDLS Peer Traffic Response frame shall echo the Dialog Token and the Link Identifier from the corresponding TDLS Peer Traffic Indication frame. The TDLS Peer Traffic Response frame shall be sent via the direct path.

When a TPU sleep STA receives a TDLS Peer Traffic Indication frame with a PTI Control element, and the TPU sleep STA has not received from the TPU buffer STA the MPDU following the MPDU that is indicated in the TDLS Peer Traffic Indication frame, the TPU sleep STA shall initiate a service period with the TPU buffer STA to retrieve the buffered traffic for the AC(s) for which no unscheduled SP is currently active.

10.2.1.16 FMS power management

10.2.1.16.1 General

Implementation of FMS is optional for a WNM STA. A STA that has a value of true for dot11MgmtOptionFMSActivated is defined as a STA that supports FMS. A STA for which dot11MgmtOptionFMSActivated is true shall set the FMS field of the Extended Capabilities element to 1.

10.2.1.16.2 FMS general procedures

When dot11MgmtOptionFMSActivated is true at the AP, the AP shall include the FMS Descriptor element in every Beacon frame. The FMS Descriptor indicates the FMS group addressed buffered BUs at the AP. If there are no buffered BUs for FMS streams accepted by the AP, the AP shall set the Length field in the FMS Descriptor element to 1. The AP shall include the FMS Descriptor element for a nontransmitted BSSID in the Multiple BSSID element sent in a Beacon frame.

When dot11MgmtOptionFMSActivated is true at the AP, the AP shall support from one to eight different FMS Stream delivery intervals for any number of FMS streams. Corresponding to these eight delivery intervals are eight FMS counters. More than one FMSID may have the same delivery interval and therefore will share the same FMS Counter. An FMS Counter corresponds to each unique delivery interval of one or more FMS Streams.

Each FMS counter decrements once per DTIM beacon and when the FMS counter reaches 0, buffered group addressed BUs assigned to that particular interval are scheduled for delivery immediately following the next Beacon frame containing the DTIM transmission. After transmission of the buffered group addressed BUs, the AP shall reset the FMS counter to the delivery interval for the FMS streams associated with that FMS counter.

A non-AP STA that does not use FMS wakes every DTIM interval and follows group addressed BU reception rules as defined in 10.2.1.8.

A STA that supports FMS shall be capable of supporting a delivery interval of 1 for any stream.

10.2.1.16.3 FMS Request procedures

A non-AP STA that supports FMS may request use of FMS by sending an FMS Request frame or Reassociation Request frame that includes one or more FMS Request elements to an AP that supports FMS. Each FMS Request element includes one or more FMS subelements. Each FMS subelement identifies an FMS stream, the requested delivery interval and the maximum delivery interval for that stream. The FMS delivery interval shall be an integer multiple of the DTIM period.

Upon reception of an FMS Request frame or Reassociation Request frame, the AP shall transmit a corresponding single FMS Response frame or Reassociation Response frame that contains a corresponding FMS Response element for each FMS Request element in the same order received. Each FMS Response element shall contain an FMS Status subelement and zero or more other subelements drawn from Table 8-159 that corresponds to each FMS subelement in the FMS Request element, in the same order.

For each FMS subelement, the following rules apply:

If the AP accepts the FMS subelement and the requested delivery interval, the Element Status in the corresponding FMS Status subelement shall be set to Accept and the FMSID is assigned to a nonzero value. In addition:

- If the FMS stream identified in the FMS subelement matches a delivery interval already in use at the AP, the AP shall assign the FMS stream to use the FMS Counter ID assigned for that delivery interval.
- When an FMS Stream is accepted by the AP, the Current Count value for that FMS Stream is decremented by 1 for each DTIM Beacon frame in which the Current Count field appears.
- The AP may reschedule transmission of the FMS Stream identified by an FMSID to align the transmission time of the FMS stream to the transmission time of other FMS streams that the STA is already receiving at the same delivery interval. The AP has the following two options:
  - Notify the STAs using that FMS Stream. The AP shall keep the nonzero Current Count value the same across two consecutive Beacon frames in which the Current Count field appears. The algorithm by which the AP chooses to align or offset the different FMS counters is unspecified.
  - Transmit an unsolicited FMS Response frame to the group address included in the original FMS Response frame for the stream with the updated Delivery Interval field when the Current Count field value reaches 0. Since the AP transmits this FMS Response frame as a group addressed frame, the frame will be scheduled for delivery at the appropriate DTIM interval when all non-AP STAs are awake to receive the frame.
- An AP may terminate the use of FMS and resume default (non-FMS) transmission rules for any FMS stream identified by FMSID for any reason. To terminate the FMS stream, the AP shall send an unsolicited FMS Response frame to the group address included in the original FMS Response frame with Delivery Interval set to 0 and the Element Status in the FMS Status subelement set to “Terminate.”
- If the FMS subelement contained a nonzero delivery interval and the non-AP STA specified a maximum delivery interval as part of the FMS request, the AP shall not modify the delivery interval for the stream greater than the maximum delivery interval specified by the non-AP STA.
- An AP shall transmit MSDUs belonging to the same FMSID in the same order that they were received at the MAC Data SAP. MSDUs belonging to the different FMSIDs are transmitted by the AP at the appropriate DTIM in the order received at the MAC data SAP based on the interval configured for the FMS stream.

If the AP denies the FMS subelement for any reason, including requested delivery interval, maximum delivery interval and TCLAS, the Element Status in the corresponding FMS Status subelement shall be set to Deny.

If the AP selects an alternate delivery interval or alternate maximum delivery interval from the value specified in the FMS Request, the FMS Status subelement shall be set to Alternate Preferred, and the FMS subelement shall indicate the AP selected value(s).

To terminate the use of FMS for an FMS Stream identified by FMSID, the non-AP STA shall transmit an FMS Request frame with an FMS Request element and FMS subelement with the FMSID matching the FMS stream and the delivery interval set to 0.

The AP shall respond to a malformed FMS Request frame or Reassociation Request frame with an FMS Response frame or Reassociation Response frame that denies all FMS Request elements by including an FMS Status code "Deny, due to request format error or ambiguous classifier" in the Element Status field in each FMS Status subelement in the FMS Response element.

10.2.1.16.4 FMS Response procedures

Upon reception of an FMS Response element in a frame that has a value in Address1 that matches its MAC address or that is a group address corresponding to a group of which it is a member and that was transmitted by the AP with which it is associated, a non-AP STA that supports FMS shall use the following procedures, based on the value of the Element Status value in FMS Status subelement in the received FMS Response element.

- If the Element Status value in FMS Status subelement is Accept:
    - The AP has accepted the FMS subelement contained within the FMS Request element. If the FMS Request element specified a nonzero delivery interval, the AP will deliver the requested streams at the delivery interval as specified by the non-AP STA in the FMS Request element.
    - After receiving the FMS Response element, the non-AP STA shall be awake for the next DTIM beacon so that the non-AP STA can synchronize with the FMS Current Count for the requested FMS Stream. Once synchronized with the FMS Current Count, the non-AP STA need not wake up at every DTIM interval to receive group addressed BUs.
- If the Element Status value in FMS Status subelement is Deny:
    - The AP will not deliver the requested streams at the delivery interval as specified by the non-AP STA in the FMS Request element. If the AP denies the usage of FMS for a particular stream, the stream is transmitted at every DTIM interval.
- If the Element Status value in FMS Status subelement is Alternate Preferred due to AP changed the maximum delivery interval:
    - The AP does not deliver the requested streams at the delivery interval as specified by the non-AP STA in the FMS Request element. The delivery interval specified in the FMS Status subelement specifies a delivery interval that the AP is willing to accept for the specified streams if the non-AP STA sends another FMS Request with that delivery interval specified.
    - The non-AP STA may submit a new FMS Request that includes the delivery interval value received from the AP. If the AP accepts this new FMS Request, it shall respond as described in 10.2.1.16.2.
- If the Element Status value in FMS Status subelement is Alternate Preferred due to AP unable to provide requested TCLAS-based classifiers:
    - The AP does not deliver the requested streams at the delivery interval as specified by the non-AP STA in the FMS Request element. The TCLAS element(s) or TCLAS Processing element in the TCLAS Status subelement contains one or more fields or subfields whose values have been modified by the AP. The AP may include fewer TCLAS elements in the FMS Response

element than were present in the request; when the AP's response includes a single TCLAS element, it does not include a TCLAS processing element. If the AP changes a TCLAS element's Classifier Type field in the FMS Response element but is unable to suggest a value for the Classifier Mask field, it shall set that field to 0. If the AP changes a TCLAS element's Classifier Type field or Classifier Mask field in the FMS Response element but is unable to suggest values for one or more Classifier Parameter subfields, it shall set those fields to 0.

— A non-AP STA receiving a modified TCLAS element having a Classifier Mask field equal to 0 or Classifier Parameter subfields equal to 0 shall interpret these values as meaning that no suggested value has been provided by the AP.

10.2.1.17 TIM Broadcast

Implementation of TIM Broadcast is optional for a WNM STA. A STA that implements TIM Broadcast has dot11MgmtOptionTIMBroadcastImplemented set to true. When dot11MgmtOptionTIMBroadcastImplemented is true, dot11WirelessManagementImplemented shall be true. A STA that has a value of true for dot11MgmtOptionTIMBroadcastActivated is defined as a STA that supports TIM Broadcast. A STA for which dot11MgmtOptionTIMBroadcastActivated is true shall set the TIM Broadcast field of the Extended Capabilities element to 1. This subclause describes TIM Broadcast procedures for STAs that have dot11MgmtOptionTIMBroadcastActivated equal to true.

TIM frames have shorter duration than Beacon frames and are potentially transmitted at a higher data rate. TIM Broadcast allows a non-AP STA to receive a TIM element without receiving a Beacon frame that may reduce the required wake time in a power save mode. The shorter receive time will reduce the power consumption for non-AP STAs in a power save mode. The shorter receive time may reduce the power consumption for stations in a standby mode.

A non-AP STA may activate the TIM Broadcast service by including a TIM Broadcast Request element in a TIM Broadcast Request frame, Association Request frame or Reassociation Request frame that is transmitted to the AP, which specifies the requested interval between TIM frame transmissions (the TIM Broadcast Interval). On receipt of a properly formatted TIM Broadcast Request element in a TIM Broadcast Request frame, Association Request frame or Reassociation Request frame, the AP shall include a TIM Broadcast Response element in the corresponding TIM Broadcast Response frame, Association Response frame or Reassociation Response frame, when dot11MgmtOptionTIMBroadcastActivated is true. When the requested TIM Broadcast Interval is acceptable, the AP shall include a TIM Broadcast Response element specifying the requested TIM Broadcast Interval and a Status field indicating "accept" when no valid TSF timestamp is present in the TIM frames, or "accept, valid timestamp present in TIM frames" when a valid TSF timestamp is present in the TIM frames. When the AP overrides the request, it shall include a TIM Broadcast Response element with a Status field indicating "Overridden," and include in the TIM Broadcast Response element the smallest TIM Broadcast Interval that is currently active. Otherwise, the AP shall include a TIM Broadcast Response element with a Status field indicating "denied." The Status field in a TIM Broadcast Response element that is included in an Association Response frame or Reassociation Response frame has implications only for the TIM Broadcast negotiation.

An AP transmitting a TIM frame with a valid TSF timestamp shall set the value of the TIM frame timestamp as defined in 10.1.3, for the Beacon frame timestamp.

If the AP accepted at least one TIM Broadcast Request with a nonzero TIM Broadcast Interval, and at least one non-AP STA in PS mode still associated with the AP received in its latest TIM Broadcast Response a status field value equal to 0 (Accepted) in response to a TIM Broadcast Request with a nonzero TIM Broadcast Interval, the AP shall transmit one or two TIM frames per TIM Broadcast Interval. The AP shall not transmit TIM frames otherwise. When TIM Broadcast Intervals overlap, a transmitted TIM frame serves both intervals and does not need to be duplicated.

If the AP transmits two TIM frames per TIM Broadcast Interval, the AP shall transmit the high data rate TIM frame first, followed by the low data rate TIM frame.

The AP shall transmit the low data rate TIM frame at the same data rate or MCS as the Beacon frame. The AP shall transmit the high data rate TIM frame at a higher data rate or using an MCS that corresponds to a higher data rate. For Clause 19 and Clause 20 PHYs, if the Beacon frame is transmitted using ERP-DSSS/CCK, the AP shall transmit the high data rate TIM frame using ERP-OFDM, and its transmission is mandatory. Otherwise, transmitting the high data rate TIM frame is optional. If the high data rate TIM is not transmitted, the AP shall set the High Data Rate TIM field to 0 in the TIM Broadcast Response element.

The TIM Broadcast Interval from the latest received TIM Broadcast Response element together with the dot11BeaconPeriod define a series of TTTTs TIM Broadcast Interval × dot11BeaconPeriod TUs apart. Time 0 is a TTTT. Non-AP STAs may use the information in the High Rate TIM Rate and Low Rate TIM Rate fields to determine which of the two TIM frames they will be receiving. The first TIM frame per TIM Broadcast Interval is scheduled to be transmitted at the TTTT plus the indicated TIM Broadcast Offset. The offset may have a negative value that allows the TIM frame(s) to be transmitted before the TBTT. The value of the offset shall not be changed as long as an associated non-AP STA is using the TIM Broadcast service.

The AP should accept new TIM Broadcast Interval requests if this implies transmitting TIM frames more frequently. For instance, if the AP currently transmits TIM frames every fourth beacon period and it receives a new request for every 3 beacon periods, then the AP should accept the new request and transmit TIM frames both every 3 and every 4 beacon periods. The AP may override incongruent requests once available resources (such as counters) have been depleted. An incongruent request is a request that contains an interval that is not an integer divide or a multiple of a currently active TIM Broadcast interval.

The AP shall accept a TIM Broadcast Interval of 1.

The AP shall increase the value (modulo 256) of the Check Beacon field in the next transmitted TIM frame(s) when a critical update occurs to any of the elements inside the Beacon frame. The following events shall classify as a critical update:

a) Inclusion of a Channel Switch Announcement
b) Inclusion of an Extended Channel Switch Announcement
c) Modification of the EDCA parameters
d) Inclusion of a Quiet element
e) Modification of the DSSS Parameter Set
f) Modification of the CF Parameter Set
g) Modification of the FH Parameter Set
h) Modification of the HT Operation element

An AP may classify other changes in the Beacon frame as critical updates.

The non-AP STA shall attempt to receive the next Beacon frame when it receives a Check Beacon field that contains a value that is different from the previously received Check Beacon field.

When dot11MgmtOptionMultiBSSIDActivated is true, the bitmap of the TIM element is interpreted as specified in 8.4.2.7.

When dot11MgmtOptionMultiBSSIDActivated is true, the A1 field of the TIM frame is the Broadcast address, the A2 field and the A3 field are set to the transmitted BSSID.

10.2.1.18 WNM-Sleep mode

10.2.1.18.1 WNM-Sleep mode capability

Implementation of the WNM-Sleep mode capability is optional for a WNM STA. A STA that implements WNM-Sleep mode has dot11MgmtOptionWNMSleepModeImplemented set to true. When dot11MgmtOptionWNMSleepModeImplemented is true, dot11WirelessManagementImplemented shall be true. A STA where dot11MgmtOptionWNMSleepModeActivated is true is defined as a STA that supports WNM-Sleep mode. A STA supporting WNM-Sleep mode shall set the WNM-Sleep Mode field of the Extended Capabilities element to 1. When dot11MgmtOptionWNMSleepModeActivated is true, dot11MgmtOptionTFSActivated shall be true.

A STA with a value of true for dot11MgmtOptionWNMSleepModeActivated may send a WNM-Sleep Mode Request or WNM-Sleep Mode Response frame to a STA within the same infrastructure BSS whose last received Extended Capabilities element contained a value of 1 for the WNM-Sleep Mode bit in the Capabilities field. WNM-Sleep mode is a service that may be provided by an AP to its associated STAs. The WNM-Sleep mode is not supported in an IBSS.

WNM-Sleep Mode enables an extended power save mode for non-AP STAs in which a non-AP STA need not listen for every DTIM Beacon frame, and need not perform GTK/IGTK updates. The non-AP STA can sleep for extended periods as indicated by the WNM-Sleep Interval.

10.2.1.18.2 WNM-Sleep mode non-AP STA operation

To use the WNM-Sleep mode service, the non-AP STA’s SME shall issue an MLME-SLEEPMODE.request primitive to send a WNM-Sleep Mode Request frame. The MLME-SLEEPMODE.request primitive shall include a valid SleepMode parameter with a WNM-Sleep Mode element. The Action Type field in the WNM-Sleep Mode element shall be set to “Enter WNM-Sleep Mode,” and the Sleep Interval field shall be included. The Sleep Interval field shall be less than the BSS Max idle period (see 10.23.12). The MLME-SLEEPMODE.request primitive shall also include a valid TFSRequest parameter as defined in the TFS Request element that the AP shall use as triggers to set the STA’s TIM bit.

When a traffic filter for group addressed frames is enabled at the AP, the STA may request a notification frame (see 10.23.11.2) be sent when requesting the establishment of the traffic filter.

The receipt of an MLME-SLEEPMODE.confirm primitive with a valid SleepMode parameter indicates to the STA’s SME that the AP has processed the corresponding WNM-Sleep Mode Request frame. The content of the WNM-Sleep mode parameter in the WNM-Sleep Mode Response frame provides the status of WNM-Sleep Mode elements processed by the AP. The non-AP STA shall delete the GTKSA if the response indicates success. If RSN is used with management frame protection, the non-AP STA shall delete the IGTKSA if the response indicates success.

While in WNM-Sleep mode, the non-AP STA need not wake up every DTIM interval for group addressed frames.

The STA wakes up every Sleep interval to check whether the corresponding TIM bit is set or group addressed traffic is pending. The non-AP STA does not participate in GTK/IGTK updates.

To exit WNM-Sleep mode, the non-AP STA’s SME shall issue an MLME-SLEEPMODE.request primitive to send a WNM-Sleep Mode Request frame with an Action Type field in the WNM-Sleep Mode element set to “Exit WNM-Sleep Mode.”

10.2.1.18.3 WNM-Sleep mode AP operation

When an AP's SME receives an MLME-SLEEPMODE.indication primitive with a valid SleepMode parameter and an Action Type in the WNM-Sleep Mode element of "Enter WNM-Sleep Mode," it shall issue an MLME-SLEEPMODE.response primitive with SleepMode parameters indicating the status of the request.The value of the Action Type field of the WNM-Sleep Mode element in the WNM-Sleep Mode Response frame shall be set to "Enter WNM-Sleep Mode."

When WNM-Sleep mode is enabled for an associated STA, the AP shall stop sending all individually addressed MPDUs to the non-AP STA. The AP may disassociate or deauthenticate the STA at any time for any reason while the non-AP STA is in WNM-Sleep mode. An AP shall perform the TFS AP operation, as specified in 10.23.11 for non-AP STAs for which it has received TFS Request elements. The AP shall set the TIM bit corresponding to the AID of the associated STA for which the AP has queued either a TFS Notify frame or matching frame. An AP shall not perform GTK/IGTK updates for the STAs in WNM-Sleep Mode.

When an AP's SME receives an MLME-SLEEPMODE.indication primitive with a valid SleepMode parameter and an Action Type in the WNM-Sleep Mode element of "Exit WNM-Sleep Mode," the AP shall disable WNM-Sleep mode service for the requesting STA, and the AP's SME shall issue an MLME-SLEEPMODE.response primitive with a SleepMode parameter indicating the status of the associated request.

If RSN is used with management frame protection and a valid PTK is configured for the STA, the current GTK and IGTK shall be included in the WNM-Sleep Mode Response frame. If a GTK/IGTK update is in progress, the pending GTK and IGTK shall be included in the WNM-Sleep Mode Response frame. If RSN is used without management frame protection and a valid PTK is configured for the STA, the current GTK shall be sent to the STA in a GTK update following the WNM-Sleep Mode Response frame.

10.2.2 Power management in an IBSS

10.2.2.1 Introduction

Subclause 10.2.2 specifies the power management mechanism for use within an IBSS.

10.2.2.2 Basic approach

The basic approach is similar to the infrastructure case in that the STAs are synchronized, and group addressed BUs and the BUs that are to be transmitted to a power-conserving STA are first announced during a period when all STAs are awake. The announcement is done via an ATIM sent in an ATIM Window. A STA in the PS mode shall listen for these announcements to determine if it needs to remain in the Awake state. The presence of the ATIM window in the IBSS indicates if the STA may use PS mode. To maintain correct information on the power save state of other STAs in an IBSS, a STA needs to remain awake during the ATIM window. At other times the STA may enter the Doze state except as indicated in the following procedures.

When a BU is to be transmitted to a destination STA that is in a PS mode, the transmitting STA first transmits an ATIM frame during the ATIM Window, in which all the STAs including those operating in a PS mode are awake. The ATIM Window is defined as a specific period of time, defined by the value of the ATIM Window parameter in the IBSS Parameter Set supplied to the MLME-START.request primitive, following a TBTT, during which only RTS, CTS, or ACK Control frames; Beacon or ATIM management frames; or (QoS) Null data frames shall be transmitted. ATIM transmission times are randomized, after a Beacon frame is either transmitted or received by the STA, using the backoff procedure with the CW equal to aCWmin. Individually addressed ATIMs shall be acknowledged. If a STA transmitting an individually addressed ATIM does not receive an acknowledgment, the STA shall execute the backoff procedure for retransmission of the ATIM. Group addressed ATIMs shall not be acknowledged.

If a STA receives an individually addressed ATIM frame during the ATIM Window, it shall acknowledge the individually addressed ATIM and stay awake to receive the announced BUs for the entire beacon interval or until it has completed successful transmission to and reception from the source STA of the received ATIM, a frame with EOSP field equal to 1. If a STA does not receive an ATIM, it may enter the Doze state at the end of the ATIM Window. Transmissions of BUs announced by ATIMs are randomized after the ATIM Window, using the backoff procedure described in Clause 9.

It is possible that an ATIM may be received from more than one STA, and that a STA that receives an ATIM may receive more than a single BU from the transmitting STA. ATIM frames are only addressed to the destination STA of the BU.

An ATIM for a BU shall have a destination address identical to that of the BU.

After the ATIM interval, only individually addressed BUs that have been successfully announced with an acknowledged ATIM and group addressed BUs that have been announced with an ATIM shall be transmitted to STAs in the PS mode. These frames shall be transmitted using the DCF (for non-QoS STAs) or EDCAF (for QoS STAs).

Figure 10-5 illustrates the basic PS operation.

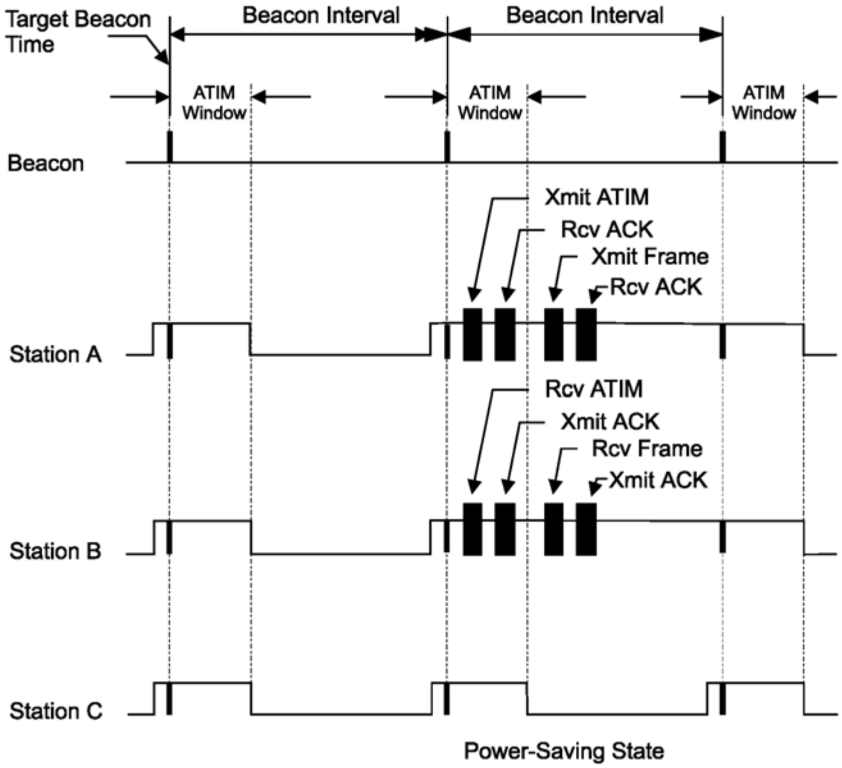


Figure 10-5—Power management in an IBSS—basic operation

The estimated power-saving state of another STA may be based on the power management information transmitted by that STA and on additional information available locally, such as a history of failed transmission attempts. The use of RTS/CTS in an IBSS may reduce the number of transmissions to a STA that is in PS mode. If an RTS is sent and a CTS is not received, the transmitting STA may assume that the destination STA is in PS mode. The method of estimating the power management state of other STAs in the IBSS is outside the scope of this standard.

10.2.2.3 Initialization of power management within an IBSS

The following procedure shall be used to initialize power management within a new IBSS, or to learn about the power management being used within an existing IBSS:

a) A STA joining an existing IBSS by the procedure in 10.1.4.4 shall update its ATIM Window with the value contained in the ATIM Window field of the IBSS Parameter Set element within the Beacon or Probe Response management frame received during the scan procedure.

b) A STA creating a new IBSS by the procedure in 10.1.4.4 shall set the value of the ATIM Window field of the IBSS Parameter Set element within the Beacon management frames transmitted to the value of its ATIM Window.

c) The start of the ATIM Window shall be the TBTT, defined in 10.1.3.3. The end of the ATIM Window shall be defined as

   TSF timer MOD dot11BeaconInterval = ATIMWindow, where ATIMWindow is the value of the ATIM Window parameter of the IBSS Parameter Set from the MLME-START.request or MLME-JOIN.request primitives.

d) The ATIM Window period shall be static during the lifetime of the IBSS.

e) An ATIM Window value of 0 shall indicate that power management is not usable within the IBSS.

10.2.2.4 STA power state transitions

A STA may enter PS mode if the value of the ATIM window in use within the IBSS is greater than 0. A STA shall not enter PS mode if the value of the ATIM window in use within the IBSS is equal to 0. A STA shall set the Power Management subfield in the Frame Control field of frames containing all or part of a BU or individually addressed Probe Request frame that it transmits using the rules in 8.2.4.1.7.

A STA in PS mode shall transition between Awake and Doze states according to the following rules:

a) If a STA is operating in PS mode, it shall enter the Awake state prior to each TBTT.

b) If a STA receives a group addressed ATIM management frame during the ATIM Window, the STA shall remain in the Awake state until either the STA receives a group addressed frame from the source STA with the EOSP field equal to 1 or the end of the next ATIM Window, whichever occurs first.

c) If a STA receives at least one individually addressed ATIM management frame containing the STA's individual address in the RA field during the ATIM Window then the STA shall remain in the Awake state at least until the earlier of the completion of the successful transmission to and reception from the source STA of each received ATIM, a frame with EOSP field equal to 1, and the end of the next ATIM Window.

d) If a STA transmits at least one individually addressed ATIM management frame containing a STA's individual address in the RA field during the ATIM Window then the STA shall remain in the Awake state at least until the earlier of the completion of the successful transmission to and reception from the destination STA of each transmitted ATIM, a frame with EOSP field equal to 1, and the end of the next ATIM Window.

e) If a STA transmits a Beacon, the STA shall remain in the Awake state until the end of the next ATIM Window.

f) If the STA has not transmitted an ATIM and does not receive either an individually addressed ATIM management frame containing the STA's individual address, or a group addressed ATIM management frame during the ATIM Window, the STA may return to the Doze state following the end of the current ATIM Window.

10.2.2.5 ATIM and frame transmission

If power management is in use within an IBSS, all STAs shall buffer individually addressed BUs for STAs that are known to be in PS mode. BUs may be sent to STAs in Active mode at any valid time.

The following rules describe operation of the ATIM and frame transmission to STAs in PS mode in an IBSS:

a) Following the reception or transmission of the Beacon frame, during the ATIM Window, the STA shall transmit an individually addressed ATIM management frame to each STA for which it has one or more buffered individually addressed BUs. If the STA has one or more buffered group addressed MSDUs (excluding those with a service class of StrictlyOrdered) or has one or more buffered group addressed MMPDUs, it shall transmit an appropriately addressed group addressed ATIM frame.

b) All STAs shall use the backoff procedure defined in 9.3.4.3 for transmission of the first ATIM following the Beacon frame. All remaining ATIMs shall be transmitted using the conventional DCF access procedure.

c) ATIM management frames shall be transmitted only during the ATIM Window.

d) A STA shall transmit no frame types other than RTS, CTS, and ACK Control frames, Beacon and ATIM management frames and (QoS)Null data frames during the ATIM Window.

e) Individually addressed ATIM management frames shall be acknowledged. If no acknowledgment is received, the ATIM shall be retransmitted using either the DCF (for non-QoS STAs) or the EDCAF (for QoS STAs). Group addressed ATIM management frames shall not be acknowledged.

f) If a STA is unable to transmit an ATIM during the ATIM Window, for example due to contention with other STAs, the STA should retain the buffered BUs and attempt to transmit the ATIM during the next ATIM Window.

g) Immediately following the ATIM Window, a STA shall begin transmission of any buffered group addressed frames for which an ATIM was previously transmitted. Following the transmission of any group addressed frames, any BUs addressed to STAs for which an acknowledgment for a previously transmitted ATIM frame was received shall be transmitted. All STAs shall use the backoff procedure defined in 9.3.4.3 for transmission of the first frame following the ATIM Window. All remaining frames shall be transmitted using either the DCF (for non-QoS STAs) or the EDCAF (for QoS STAs).

h) If a buffered BU is transmitted using fragmentation and if the BU has been partially transmitted when the next Beacon frame is sent, the STA shall retain the buffered BU and announce the remaining fragments by transmitting an ATIM during the next ATIM Window.

i) If a STA is unable to transmit a buffered BU during the beacon interval in which it was announced, for example due to contention with other STAs, the STA shall retain the buffered BU and announce the BU again by transmitting an ATIM during the next ATIM Window.

j) Following the transmission of all buffered BUs, a STA may transmit BUs without announcement to STAs that are known to be in the Awake state for the current beacon interval.

k) A STA may discard BUs buffered for later transmission to power-saving STAs if the STA determines that the BU has been buffered for an excessive amount of time or if other conditions internal to the STA implementation make it desirable to discard buffered BUs (e.g., buffer starvation). A BU should not be discarded that has been buffered for less than dot11BeaconPeriod. The algorithm to manage this buffering is beyond the scope of this standard.

l) A STA may transmit individually addressed or group addressed Null data frames within the ATIM window to indicate the STA's intent to change power management modes. The STA may transition into PS mode after acknowledgements have been successfully received for all individually addressed Null data frames or after the STA has transmitted group addressed Null data frames at least dot11BSSBroadcastNullCount times.

10.2.3 Power management in an MBSS

Power management in an MBSS is described in 13.14.

10.2.4 SM power save

A STA consumes power on all active receive chains, even though they are not necessarily required for the actual frame exchange. The SM Power Save feature allows a STA to operate with only one active receive chain for a significant portion of time.

The STA controls which receive chains are active through the PHY-RXCONFIG.request primitive specifying a PHYCONFIG_VECTOR parameter ACTIVE_RXCHAIN_SET that indicates which of its receive chains should be active.

In dynamic SM power save mode, a STA enables its multiple receive chains when it receives the start of a frame sequence addressed to it. Such a frame sequence shall start with a single-spatial stream individually addressed frame that requires an immediate response and that is addressed to the STA in dynamic SM power save mode. An RTS/CTS sequence may be used for this purpose. The receiver shall be capable of receiving a PPDU that is sent using an MCS that indicates more than one spatial stream a SIFS after the end of its response frame transmission. The receiver switches to the multiple receive chain mode when it receives the RTS addressed to it and switches back immediately when the frame sequence ends.

NOTE—A STA in dynamic SM power save mode cannot distinguish between an RTS/CTS sequence that precedes a MIMO transmission and any other RTS/CTS and, therefore, always enables its multiple receive chains when it receives an RTS addressed to itself.

The receiver can determine the end of the frame sequence through any of the following:

- It receives an individually addressed frame addressed to another STA.
- It receives a frame with a TA that differs from the TA of the frame that started the TXOP.
- The CS mechanism (see 9.3.2.1) indicates that the medium is idle at the TxPIFS slot boundary (defined in 9.3.7).

A STA in static SM power save mode maintains only a single receive chain active.

An HT STA may use the SM Power Save frame to communicate its SM Power Save state. A non-AP HT STA may also use SM Power Save bits in the HT Capabilities element of its Association Request to achieve the same purpose. The latter allows the STA to use only a single receive chain immediately after association.

A STA that has one or more DLS links shall notify all STAs with which it has a DLS link of any change in SM power save mode before operating in that mode.

Changes to the number of active receive chains are made only after the SM Power Save Mode indication has been successfully delivered (i.e., by acknowledgment of a frame carrying the HT Capabilities element or by acknowledgment of a SM Power Save frame). The SM Power Save Mode indication shall be transmitted using an individually addressed frame.

10.3 STA authentication and association

10.3.1 State variables

A STA (local) for which dot11OCBActivated is false keeps an enumerated state variable for each STA (remote) with which direct communication via the WM is needed. In this context, direct communication refers to the transmission of any class 2 or class 3 frame with an Address 1 field that matches the MAC address of the remote STA.

A STA for which dot11MeshActivated is true (i.e., a mesh STA) does not use procedures described in 10.3.5. Instead, a mesh STA uses a mesh peering management protocol (MPM) or a authenticated mesh peering exchange (AMPE) to manage states and state variables for each peer STA. See 13.3 and 13.5 for details.

A STA for which dot11OCBActivated is true does not use MAC sublayer authentication or association and does not keep this state variable.

For nonmesh STAs, this state variable expresses the relationship between the local STA and the remote STA. It takes on the following values:

- *State 1:* Initial start state, unauthenticated, unassociated.
- *State 2:* Authenticated, not associated.
- *State 3:* Authenticated and associated (Pending RSN Authentication).
- *State 4:* Authenticated and associated.

The state variable is kept within the MLME (i.e., is written and read by the MLME). The SME may also read this variable.

Mesh STAs manage the state variable as described in 13.3.2.

10.3.2 State transition diagram for nonmesh STAs

Figure 10-6 shows the state transition diagram for nonmesh STA states. Note that only events causing state changes are shown. The state of the sending STA given by Figure 10-6 is with respect to the intended receiving STA.

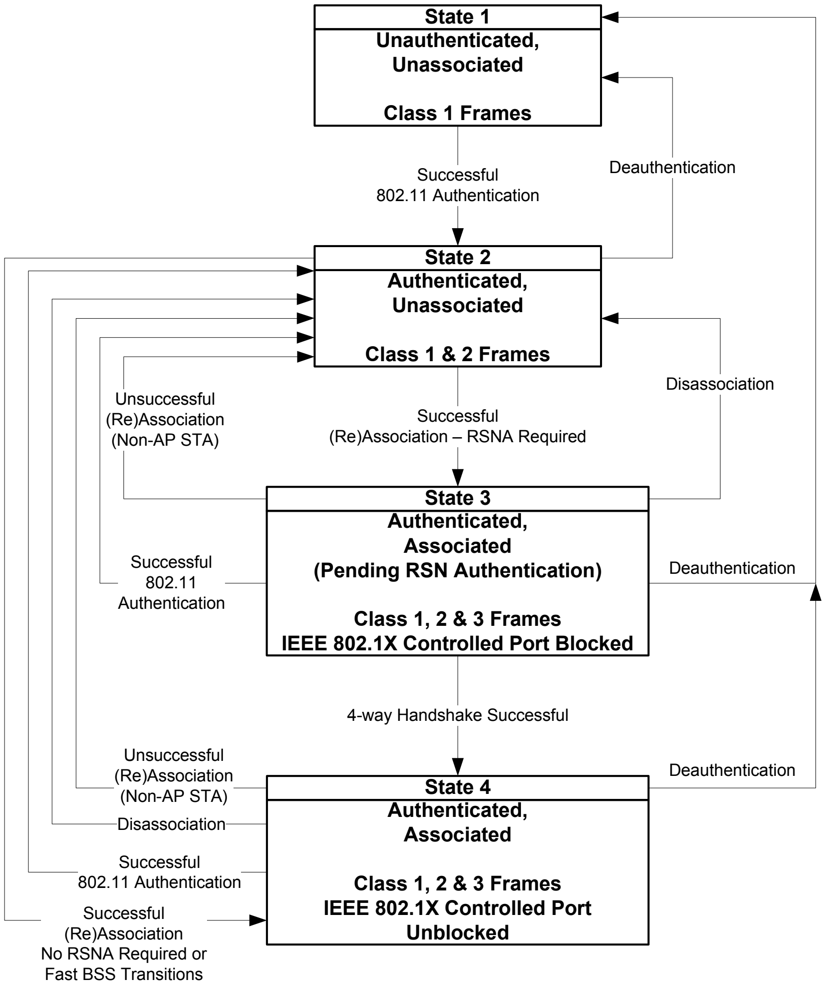

State 1
Unauthenticated, Unassociated
Class 1 Frames
Successful 802.11 Authentication
Deauthentication
State 2
Authenticated, Unassociated
Class 1 & 2 Frames
Disassociation
Unsuccessful (Re)Association (Non-AP STA)
Successful (Re)Association – RSNA Required
State 3
Authenticated, Associated (Pending RSN Authentication)
Class 1, 2 & 3 Frames
IEEE 802.1X Controlled Port Blocked
Successful 802.11 Authentication
Deauthentication
4-way Handshake Successful
Unsuccessful (Re)Association (Non-AP STA)
Deauthentication
State 4
Authenticated, Associated
Class 1, 2 & 3 Frames
IEEE 802.1X Controlled Port Unblocked
Disassociation
Successful 802.11 Authentication
Successful (Re)Association No RSNA Required or Fast BSS Transitions

Figure 10-6—Relationship between state and services

10.3.3 Frame filtering based on STA state

The current state existing between the transmitter and receiver STAs determines the IEEE 802.11 frame types that may be exchanged between that pair of STAs (see Clause 8). A unique state exists for each pair of transmitter and receiver STAs. The allowed frame types are grouped into classes and the classes correspond to the STA state. In State 1, only Class 1 frames are allowed. In State 2, either Class 1 or Class 2 frames are allowed. In State 3 and State 4, all frames are allowed (Classes 1, 2, and 3). The frame classes are defined as follows:

a) Class 1 frames
   1) Control frames
      i) RTS
      ii) CTS
      iii) ACK

        iv) CF-End+ACK
        v) CF-End
        vi) Within an IBSS, Block Ack (BlockAck)
        vii) Within an IBSS, Block Ack Request (BlockAckReq)
    2) Management frames
        i) Probe Request/Response
        ii) Beacon
        iii) Authentication
        iv) Deauthentication
        v) ATIM
        vi) Public Action
        vii) Self-protected Action
        viii) Within an IBSS, all Action frames and all Action No Ack frames
    3) Data frames
        i) Data frames between STAs in an IBSS
        ii) Data frames between peers using DLS

b) Class 2 frames
    1) Management frames
        i) Association Request/Response
        ii) Reassociation Request/Response
        iii) Disassociation

c) Class 3 frames
    1) Data frames
        i) Data frames between STAs in an infrastructure BSS or in an MBSS
    2) Management frames
        i) Within an infrastructure BSS or an MBSS, all Action and Action No Ack frames except those that are declared to be Class 1 or Class 2 frames (above)
    3) Control frames
        i) PS-Poll
        ii) Within an infrastructure BSS or an MBSS, Block Ack (BlockAck)
        iii) Within an infrastructure BSS or an MBSS, Block Ack Request (BlockAckReq)

Class 2 and Class 3 frames are not allowed in an IBSS. If a STA in an IBSS receives a Class 2 or Class 3 frame, it shall ignore the frame.

The use of the word “receive” in 10.3 refers to a frame that meets all of the filtering criteria specified in Clause 11 and Clause 9.

10.3.4 Authentication and deauthentication

10.3.4.1 General

This subclause describes the procedures used for IEEE 802.11 authentication and deauthentication. The states used in this description are defined in 10.3.1.

Successful authentication sets the STA's state to State 2, if it was in State 1. Unsuccessful authentication leaves the STA's state unchanged. The STA shall not transmit Class 2 frames unless in State 2 or State 3 or State 4. The STA shall not transmit Class 3 frames unless in State 3 or State 4.

Deauthentication notification sets the STA's state to State 1. The STA shall become authenticated again prior to sending Class 2 frames. Deauthentication notification when in State 3 or 4 implies disassociation as well. A STA may deauthenticate a peer STA at any time, for any reason.

If STA A in an infrastructure BSS receives a Class 2 or Class 3 frame from STA B that is not authenticated with STA A (i.e., the state for STA B is State 1), STA A shall discard the frame. If the frame has an individual address in the Address 1 field, the MLME of STA A shall send a Deauthentication frame to STA B.

Authentication is optional in an IBSS. In an infrastructure BSS, authentication is required. APs do not initiate authentication.

10.3.4.2 Authentication—originating STA

Upon receipt of an MLME-AUTHENTICATE.request primitive, the originating STA shall authenticate with the indicated STA using the following procedure:

a) If the STA is in an IBSS, the SME shall delete any PTKSA and temporal keys held for communication with the indicated STA by using the MLME-DELETEKEYS.request primitive (see 11.5.15).
b) The STA shall execute one of the following:
    1) For the Open System or Shared Key authentication algorithm, the authentication mechanism described in 11.2.3.2 or 11.2.3.3, respectively.
    2) For the FT authentication algorithm in an ESS, the authentication mechanism described in 12.5, or, if resource requests are included, 12.6.
    3) For SAE authentication in an ESS, IBSS, or MBSS, the authentication mechanism described in 11.3.
c) If the authentication was successful within the AuthenticateFailureTimeout, the state for the indicated STA shall be set to State 2 if it was State 1; the state shall remain unchanged if it was other than State 1.
d) The MLME shall issue an MLME-AUTHENTICATE.confirm primitive to inform the SME of the result of the authentication.

10.3.4.3 Authentication—destination STA

Upon receipt of an Authentication frame with authentication transaction sequence number equal to 1, the destination STA shall authenticate with the originating STA using the following procedure:

a) If Open System or Shared Key authentication algorithm is being used, the STA shall execute the procedure described in 11.2.3.2 or 11.2.3.3, respectively. These result in the generation of an MLME-AUTHENTICATE.indication primitive to inform the SME of the authentication request.
b) If FT authentication is being used, the MLME shall issue an MLME-AUTHENTICATE.indication primitive to inform the SME of the authentication request, including the FT Authentication Elements, and the SME shall execute the procedure as described in 12.5 or 12.6.
c) If SAE authentication is being used in an ESS, IBSS, or MBSS, the MLME shall issue an MLME-AUTHENTICATE.indication primitive to inform the SME of the authentication request, including the SAE Authentication Elements, and the SME shall execute the procedure as described in 11.3
d) If the STA is in an IBSS and management frame protection was not negotiated when the PTKSA(s) were created, the SME shall delete any PTKSA and temporal keys held for communication with the

originating STA by using the MLME-DELETEKEYS.request primitive (see 11.5.15).

e) Upon receipt of an MLME-AUTHENTICATE.response primitive, if the ResultCode is not SUCCESS, the MLME shall transmit an Authentication frame with the corresponding status code, as defined in 8.4.1.9, and the state for the originating STA shall be left unchanged. The Authentication frame is constructed using the appropriate procedure in 11.2.3.2, 11.2.3.3, 12.5 or 12.6.

f) Upon receipt of an MLME-AUTHENTICATE.response primitive, if the ResultCode is SUCCESS, the MLME shall transmit an Authentication frame that is constructed using the appropriate procedure in 11.2.3.2, 11.2.3.3, 12.5 or 12.6, with a status code of Successful, and the state for the originating STA shall be set to State 2 if it was in State 1.

If the STA is in an IBSS, if the SME decides to initiate an RSNA, and if the SME does not know the security policy of the peer, it may issue an individually addressed Probe Request frame to the peer by invoking an MLME-SCAN.request primitive to discover the peer's security policy.

10.3.4.4 Deauthentication—originating STA

The originating STA shall deauthenticate with the indicated STA using the following procedure:

a) The SME shall generate an MLME-DEAUTHENTICATE.request primitive containing the appropriate reason code for the STA deauthentication, as defined in Table 8-36 of 8.4.1.7.

b) On receipt of the MLME-DEAUTHENTICATE.request primitive, if the state for the indicated STA is State 2, State 3, or State 4, the MLME shall shall generate a Deauthentication frame to be transmitted to the indicated STA.

   NOTE—as the Deauthentication frame is a bufferable MMPDU, the transmission of this frame might be delayed by the operation of a power-saving protocol. The AID and the PTKSA are maintained (when applicable) until the frame is acknowledged or attempts to transmit the frame are abandoned.

c) The state for the indicated STA shall be set to State 1.

d) Once the Deauthentication frame is acknowledged or attempts to transmit the frame are abandoned, the MLME shall issue an MLME-DEAUTHENTICATE.confirm primitive to inform the SME of the deauthentication.

e) The SME, upon receipt of an MLME-DEAUTHENTICATE.confirm primitive, shall delete any PTKSA and temporal keys held for communication with the indicated STA by using the MLME-DELETEKEYS.request primitive (see 11.5.15) and by invoking MLME-SETPROTECTION.request(None).

f) If the STA is contained within an AP, its SME, upon receipt of an MLME-DEAUTHENTICATE.confirm primitive, shall release the AID assigned for the indicated STA, if the state for the indicated STA was State 3 or State 4.

g) If the STA is contained within an AP, its SME shall inform the DS of the disassociation, if the state for the indicated STA was State 3 or State 4.

h) If the STA is a mesh STA, its SME shall inform the mesh peering instance controller (see 13.3.4) of the deauthentication.

10.3.4.5 Deauthentication—destination STA

Upon receipt of a Deauthentication frame from a STA for which the state is State 2, State 3, or State 4, the destination STA shall deauthenticate with the originating STA using the following procedure:

a) If management frame protection was not negotiated when the PTKSA(s) were created, or if MFP is in use and the frame is not discarded per MFP processing, the MLME shall issue an MLME-DEAUTHENTICATE.indication primitive to inform the SME of the deauthentication, and set the state for the originating STA to State 1.

b) Upon receiving an MLME-DEAUTHENTICATE.indication primitive, the SME shall
   1) Delete any PTKSA and temporal keys held for communication with the originating STA by using the MLME-DELETEKEYS.request primitive (see 11.5.15) and by invoking MLME-SETPROTECTION.request(None).
   2) If the STA is contained within an AP, release the AID assigned for the indicated STA and shall inform the DS of the disassociation, if the state for the originating STA was State 3 or State 4.
   3) If the STA is a mesh STA, inform the mesh peering instance controller (see 13.3.4) of the deauthentication.

10.3.5 Association, reassociation, and disassociation

10.3.5.1 General

Subclause 10.3.5 describes the procedures used for IEEE 802.11 association, reassociation and disassociation.

The states used in this description are defined in 10.3.1.

Successful association enables a STA to exchange Class 3 frames. Successful association sets the STA's state to State 3 or State 4.

Successful reassociation enables a STA to exchange Class 3 frames. Unsuccessful reassociation when not in State 1 leaves the STA’s state unchanged (with respect to the AP that was sent the Reassociation Request (which may be the current STA)). Successful reassociation sets the STA’s state to State 3 or State 4 (with respect to the AP that was sent the Reassociation Request). Successful reassociation when not in State 1 sets the STA’s state to State 2 (with respect to the current AP, if this is not the AP that was sent the Reassociation Request). Reassociation shall be performed only if the originating STA is already associated in the same ESS.

Disassociation notification when not in State 1 sets the STA’s state to State 2. The STA shall become associated again prior to sending Class 3 frames. A STA may disassociate a peer STA at any time, for any reason.

If STA A in an infrastructure BSS receives a Class 3 frame from STA B that is authenticated but not associated with STA A (i.e., the state for STA B is State 2), STA A shall discard the frame. If the frame has an individual address in the Address 1 field, the MLME of STA A shall send a Disassociation frame to STA B.

Association is not applicable in an IBSS. In an infrastructure BSS, association is required. APs do not initiate association.

10.3.5.2 Non-AP STA association initiation procedures

The SME shall delete any PTKSA and temporal keys held for communication with the AP by using MLME-DELETEKEYS.request primitive (see 11.5.15) before invoking MLME-ASSOCIATE.request primitive.

If dot11InterworkingServiceActivated is true, the STA does not have credentials for the AP, and the STA is initiating an emergency services association procedure, the SME shall submit the MLME-ASSOCIATE.request with EmergencyServices parameter set to true.

Upon receipt of an MLME-ASSOCIATE.request primitive, a non-AP STA shall associate with an AP using the following procedure:

a) If the state for the AP is State 1, the MLME shall inform the SME of the failure of the association by issuing an MLME-ASSOCIATE.confirm primitive, and this procedure ends.

b) The MLME shall transmit an Association Request frame to the AP. If the MLME-ASSOCIATE.request primitive contained an RSNE with only one pairwise cipher suite and only one authenticated key suite, this RSNE shall be included in the Association Request frame. If the MLME-ASSOCIATE.request primitive contained the EmergencyServices parameter set to true, an Interworking element with the UESA field set to 1 shall be included in the Association Request frame.

c) If an Association Response frame is received with a status code of Successful, the state for the AP shall be set to State 4 or State 3 if RSNA Establishment is required. The state for any other AP which is State 3 or State 4 prior to the association request shall be set to State 2, and the MLME shall issue an MLME-ASSOCIATE.confirm primitive to inform the SME of the successful completion of the association.

d) If an Association Response frame is received with a status code other than Successful or the AssociateFailureTimeout expires the state for the AP shall be set to State 2, and the MLME shall issue an MLME-ASSOCIATE.confirm primitive to inform the SME of the failure of the association. The status code returned in the Association Response frame indicates the cause of the failed association attempt. Any misconfiguration or parameter mismatch, e.g., data rates required as basic rates that the STA did not indicate as supported in the STA's Supported Rates element, shall be corrected before the SME issues an MLME-ASSOCIATE.request primitive for the same AP. If the status code indicates the association failed because of a reason that is not related to configuration (e.g., the AP is unable to support additional associations) and the Association Response frame does not include a Timeout Interval element with Timeout Interval Type equal to 3 the SME shall not issue an MLME-ASSOCIATE.request primitive for the same AP until a period of at least 2 s has elapsed. If the status code indicates the association failed and the Association Response frame contains a Timeout Interval element with Timeout Interval Type equal to 3, the SME shall not issue an MLME-ASSOCIATE.request primitive for the same AP until the period specified in the Timeout Interval element has elapsed.

e) If an MLME-ASSOCIATE.confirm primitive is received with a ResultCode of SUCCESS, and RSNA is required, then the SME shall perform a 4-way handshake to establish an RSNA. As a part of a successful 4-way handshake, the SME enables protection by invoking MLME-SETPROTECTION.request(Rx_Tx)

f) Upon receipt of the MLME-SETPROTECTION.request(Rx_Tx), the MLME shall set the state of the STA to State 4.

10.3.5.3 AP association receipt procedures

Upon receipt of an Association Request frame from a non-AP STA for which the state is State 1, the AP’s MLME shall transmit an Association Response frame with an appropriate status code.

Upon receipt of an Association Request frame from a non-AP STA for which the state is State 2, State 3, or State 4, the AP's MLME shall associate with the non-AP STA using the following procedure:

a) The MLME shall issue an MLME-ASSOCIATE.indication primitive to inform the SME of the association request.

b) At an AP having dot11InterworkingServiceActivated equal to true, subsequent to receiving an MLME-ASSOCIATE.indication primitive with EmergencyServices set to true that does not include an RSNE, the SME shall accept the association request even if dot11RSNAActivated is true and dot11PrivacyInvoked is true thereby granting access, using unprotected frames (see 8.2.4.1.9), to the network for emergency services purposes.

c) Upon receiving an MLME-ASSOCIATE.indication primitive, when management frame protection is not in use, the SME shall delete any PTKSA and temporal keys held for communication with the STA by using the MLME-DELETEKEYS.request primitive (see 11.5.15)

d) In an RSNA, the AP shall check the values received in the RSNE to see whether the values received match the AP's security policy. If not, the association shall not be accepted.

e) If the AP's state for the non-AP STA is 4, and the AP has a valid security association for the non-AP STA, and has negotiated management frame protection with the non-AP STA, and an earlier, timed out SA Query procedure with the non-AP STA has not allowed a new association process to be started without an additional SA Query procedure,

   1) the SME shall reject the Association Request by generating an MLME-ASSOCIATE.response primitive with ResultCode "Association request rejected temporarily; try again later."

   2) The SME shall not modify any association state for the non-AP STA, and shall include in the MLME-ASSOCIATE.response primitive a Timeout Interval element with Timeout interval type set to 3 (Association Comeback time), specifying a comeback time when the AP would be ready to accept an association with this STA.

   3) Following this, if the SME is not already engaging in an SA Query with the STA, the SME shall issue one MLME-SAQuery.request primitive addressed to the STA every dot11AssociationSAQueryRetryTimeout TUs until a matching MLME-SAQuery.confirm primitive is received or dot11AssociationSAQueryMaximumTimeout TUs from the beginning of the SA Query procedure have passed.

   4) The SME shall specify a TransactionIdentifier parameter value in the MLME-SAQuery.request primitive, and increment the value by 1 for each subsequent MLME-SAQuery.request primitive, rolling over the value to 0 after the maximum allowed value is reached.

   5) The MLME may interpret reception of a valid protected frame as an indication of a successfully completed SA Query, and thereby generate an MLME-SAQuery.confirm primitive.

   6) If an MLME-SAQuery.confirm primitive with an outstanding transaction identifier is not received within dot11AssociationSAQueryMaximumTimeout period, the SME shall allow the association process to be started without starting an additional SA Query procedure.

f) The SME shall refuse an association request from a STA that does not support all the rates in the BSSBasicRateSet parameter.

g) The SME shall refuse an association request from an HT STA that does not support all the MCSs in the BSSBasicMCSSet parameter.

h) The SME shall generate an MLME-ASSOCIATE.response primitive addressed to the non-AP STA. When the association is not successful, the SME shall indicate a specific reason for the failure to associate in the ResultCode parameter as defined in 6.3.7.5.2. If the ResultCode is SUCCESS, and the SME has an existing SA with the non-AP STA, and an SA Query procedure with that non-AP STA has failed to receive a valid response, then the SME shall send an MLME-DISASSOCIATE.request primitive to the STA with Reason Code "Previous Authentication no longer valid."[40] If the ResultCode is SUCCESS, the association identifier assigned to the STA shall be included in the MLME-ASSOCIATE.response primitive, and the SME shall delete any PTKSA and temporal keys held for communication with the STA by using the MLME-DELETEKEYS.request primitive (see 11.5.15).

i) Upon receipt of an MLME-ASSOCIATE.response primitive, the MLME shall transmit an Association Response frame) to the STA.

j) When the ResultCode of the MLME-ASSOCIATE.response primitive is not SUCCESS, if management frame protection is in use the state for the STA shall be left unchanged. and if management frame protection is not in use set to State 3 if it was in State 4.

k) When the Association Response frame with a status code of Successful is acknowledged by the STA, the state for the STA shall be set to State 4 or State 3 if RSNA establishment is required.

[40] This MLME-DISASSOCIATE.request generates a protected Disassociation frame addressed to the STA.

l) If RSNA establishment is required, the SME shall attempt a 4-way handshake. Upon a successful completion of a 4-way handshake, the SME shall enable protection by invoking MLME-SETPROTECTION.request(Rx_Tx). Upon receipt of the MLME-SETPROTECTION.request(Rx_Tx), the MLME shall set the state for the STA to State 4.

m) The SME shall inform the DS of any changes in the association state.

10.3.5.4 Non-AP STA reassociation initiation procedures

Except when the association is part of a fast BSS transition, the SME shall delete any PTKSA and temporal keys held for communication with the AP by using the MLME-DELETEKEYS.request primitive (see 11.5.15) before invoking an MLME-REASSOCIATE.request primitive.

If dot11InterworkingServiceActivated is true and the STA was associated to the ESS for unsecured access to emergency services, the SME shall submit the MLME-REASSOCIATE.request with EmergencyServices parameter set to true.

Upon receipt of an MLME-REASSOCIATE.request primitive, a non-AP STA shall reassociate with an AP using the following procedure:

a) If the STA is not associated in the same ESS or the state for the new AP is State 1, the MLME shall inform the SME of the failure of the reassociation by issuing an MLME-REASSOCIATE.confirm primitive, and this procedure ends.

b) The MLME shall transmit a Reassociation Request frame to the new AP. If the MLME-REASSOCIATE.request primitive contained an RSNE with only one pairwise cipher suite and only one authenticated key suite, this RSNE shall be included in the Reassociation Request frame. If the MLME-REASSOCIATE.request primitive contained the EmergencyServices parameter set to true, an Interworking element with the UESA field set to 1 shall be included in the Reassociation Request frame.

c) If a Reassociation Response frame is received with a status code of Successful, the state variable for the new AP shall be set to State 4, or to State 3 if RSNA establishment is required and the FT Protocol is not used with respect to the new AP and, unless the old AP and new AP are the same, to State 2 with respect to the old AP, and the MLME shall issue an MLME-REASSOCIATE.confirm primitive to inform the SME of the successful completion of the reassociation.

d) If a Reassociation Response frame is received with a status code other than Successful or the ReassociateFailureTimeout expires:

    1) Except when the association is part of a fast BSS transition, the state for the AP shall be set to State 2 with respect to the new AP.

    2) The MLME shall issue an MLME-REASSOCIATE.confirm primitive to inform the SME of the failure of the reassociation. The ResultCode returned in the MLME-REASSOCIATE.confirm primitive indicates the cause of the failed reassociation attempt. Any misconfiguration or parameter mismatch, e.g., data rates required as basic rates that the STA did not indicate as supported in the STA's Supported Rates element, shall be corrected before the SME issues an MLME-REASSOCIATE.request primitive for the same AP. If the status code indicates the reassociation failed because of a reason that is not related to configuration (e.g., the AP is unable to support additional associations) and the Reassociation Response frame does not include a Timeout Interval element with Timeout Interval Type equal to 3 the SME shall not issue an MLME-REASSOCIATE.request primitive for the same AP until a period of at least 2 s has elapsed. If the status code indicates the reassociation failed and the Reassociation Response frame contains a Timeout Interval element with Timeout Interval Type equal to 3, the SME shall not issue an MLME-REASSOCIATE.request primitive for the same AP until the period specified in the Timeout Interval element has elapsed.

e) If an MLME-REASSOCIATE.confirm primitive is received with a ResultCode of SUCCESS, and RSNA is required, and the STA is in State 3, then the SME shall perform a 4-way handshake to establish an RSNA. As a part of a successful 4-way handshake, the SME shall enable protection by invoking MLME-SETPROTECTION.request(Rx_Tx).

f) Upon receipt of the MLME-SETPROTECTION.request(Rx_Tx), the MLME shall set the state of the STA to State 4.

10.3.5.5 AP reassociation receipt procedures

Upon receipt of an Reassociation Request frame from a non-AP STA for which the state is State 1, the AP's MLME shall transmit an Reassociation Response frame with an appropriate status code.

Upon receipt of a Reassociation Request frame from a STA for which the state is State 2, State 3, or State 4, the AP's MLME shall reassociate with the STA using the following procedure:

a) The MLME shall issue an MLME-REASSOCIATE.indication primitive to inform the SME of the reassociation request.

b) At an AP having dot11InterworkingServiceActivated equal to true, subsequent to receiving an MLME-REASSOCIATE.indication primitive with EmergencyServices set to true that does not include an RSN parameter, the SME shall accept the reassociation request even if dot11RSNAActivated is true and dot11PrivacyInvoked is true thereby granting access, using unprotected frames (see 8.2.4.1.9), to the network for emergency services purposes.

c) In an RSNA, the SME shall check the values received in the RSNE to see whether the values received match the AP's security policy. If not, the association shall not be accepted.

d) If the AP's state for the non-AP STA is 4, the non-AP STA has a valid security association, the non-AP STA has negotiated management frame protection, and the reassociation is not a part of a Fast BSS Transition, and an earlier, timed out SA Query procedure with the non-AP STA has not allowed a new association process to be started without an additional SA Query procedure:

   1) The SME shall reject the Reassociation Request by generating an MLME-REASSOCIATE.response primitive with ResultCode "Association request rejected temporarily; Try again later."

   2) The SME shall not modify any association state for the non-AP STA, and shall include in the MLME-REASSOCIATE.response primitive a Timeout Interval element with type set to 3 (Association Comeback time), specifying a comeback time when the AP would be ready to accept an association with this STA.

   3) Following this, if the SME is not in an ongoing SA Query with the STA, the SME shall issue one MLME-SAQuery.request primitive addressed to the STA every dot11AssociationSAQueryRetryTimeout TUs until a matching MLME-SAQuery.confirm primitive is received or dot11AssociationSAQueryMaximumTimeout TUs from the beginning of the SA Query procedure have passed.

   4) The SME shall insert the TransactionIdentifier in MLME-SAQuery.request primitive, increment this by 1 for each subsequent MLME-SAQuery.request primitive, and roll it over to 0 after the maximum allowed value in this field.

   5) An MLME may interpret reception of a valid protected frame as an indication of a successfully completed SA Query and thereby generate an MLME-SAQuery.confirm primitive.

   6) If an MLME-SAQuery.confirm primitive with an outstanding transaction identifier is not received within dot11AssociationSAQueryMaximumTimeout period, the SME shall allow the association process to be started without starting an additional SA Query procedure.

e) The SME shall refuse a reassociation request from a STA that does not support all the rates in the BSSBasicRateSet parameter.

f) The SME shall refuse a reassociation request from an HT STA that does not support all the MCSs in the BSSBasicMCSSet parameter.

g) The SME shall generate an MLME-REASSOCIATE.response primitive addressed to the non-AP STA. If the reassociation is not successful, the SME shall indicate a specific reason for the failure to reassociate in the ResultCode parameter as defined in 6.3.7.5.2.

h) If the ResultCode is SUCCESS, and the SME has an existing SA with the non-AP STA, and an SA Query procedure with that non-AP STA has failed to receive a valid response, then the SME shall issue an MLME-DISASSOCIATE.request primitive with Reason Code "Previous Authentication no longer valid."

NOTE—This MLME-DISASSOCIATE.request generates a protected Disassociation frame addressed to the STA.

i) If the ResultCode is SUCCESS, the association identifier assigned to the STA shall be included in this primitive. If the association is not part of a fast BSS transition and management frame protection is not in use, the SME shall delete any PTKSA and temporal keys held for communication with the STA by using MLME-DELETEKEYS.request primitive (see 11.5.15).

j) Upon receipt of an MLME-REASSOCIATE.response primitive, the MLME shall transmit a Reassociation Response frame to the STA.

k) When the Reassociation Response frame with a status value of Successful is acknowledged by the STA, the state variable for the STA shall be set to State 4, or to State 3 if RSNA establishment is required on the new AP and the FT Protocol is not used on the new AP.

l) When the ResultCode of the reassociation is not SUCCESS, if management frame protection is in use the state for the STA shall be left unchanged on the AP the Reassociation Request frame was sent to, When the ResultCode is not SUCCESS and management frame protection is not in use and the association is not part of a fast BSS transition, the state for the STA is set to State 3 if it was in State 4.

m) If RSNA establishment is required and FT is not in use, the SME shall attempt a 4-way handshake. Upon a successful completion of a 4-way handshake, the SME shall enable protection by invoking MLME-SETPROTECTION.request(Rx_Tx). Upon receipt of the MLME-SETPROTECTION.request(Rx_Tx), the MLME shall set the state for the STA to State 4.

n) The SME shall inform the DS of any changes in the association state.

10.3.5.6 Non-AP STA disassociation initiation procedures

The SME shall issue an MLME-DISASSOCIATE.request primitive that includes an appropriate Reason Code as defined in Table 8-36 of 8.4.1.7.

Upon receipt of an MLME-DISASSOCIATE.request primitive, a non-AP STA's MLME shall disassociate from an AP using the following procedure:

a) If the state for the AP is State 3 or State 4, the MLME shall transmit a Disassociation frame to the AP.

b) The state for the AP shall be set to State 2 if it was not State 1.

c) The MLME shall issue an MLME-DISASSOCIATE.confirm primitive to inform the SME of the successful completion of the disassociation.

d) Upon receiving a MLME-DISASSOCIATE.confirm primitive, the SME shall delete any PTKSA and temporal keys held for communication with the AP by using the MLME-DELETEKEYS.request primitive (see 11.5.15) and by invoking MLME-SETPROTECTION.request(None).

10.3.5.7 Non-AP STA disassociation receipt procedure

Upon receipt of a Disassociation frame from an AP for which the state is State 3 or State 4, if management frame protection was not negotiated when the PTKSA(s) were created, or if MFP is in use and the frame is not discarded per MFP processing, a non-AP STA shall disassociate from the AP using the following procedure:

a) The state for the AP shall be set to State 2.
b) The MLME shall issue an MLME-DISASSOCIATE.indication primitive to inform the SME of the disassociation.
c) Upon receiving the MLME-DISASSOCIATE.indication primitive, the SME shall delete any PTKSA and temporal keys held for communication with the AP by using the MLME-DELETEKEYS.request primitive (see 11.5.15) and by invoking MLME-SETPROTECTION.request(None).
d) If the reason code indicates a configuration or parameter mismatch as the cause of the disassociation, the SME shall not attempt to associate or reassociate with the AP until the configuration or parameter mismatch has been corrected.
e) If the reason code indicates the STA was disassociated for a reason other than configuration or parameter mismatch, the SME shall not attempt to associate or reassociate with the AP until a period of 2 s has elapsed.

10.3.5.8 AP disassociation initiation procedure

The SME shall issue an MLME-DISASSOCIATE.request primitive that includes an appropriate Reason Code as defined Table 8-36 of 8.4.1.7.

Upon receipt of an MLME-DISASSOCIATE.request primitive, an AP shall disassociate a STA using the following procedure:

a) If the state for the STA is State 3 or State 4, the AP shall shall generate a Disassociation frame to be transmitted to the indicated STA.

   NOTE—as the Disassociation frame is a bufferable MMPDU, the transmission of this frame might be delayed by the operation of a power-saving protocol. The AID and the PTKSA are maintained (when applicable) until the frame is acknowledged or attempts to transmit the frame are abandoned.

b) The state for the STA shall be set to State 2, if it was not State 1.
c) Once the Disassociation frame is acknowledged or attempts to transmit the frame are abandoned, the MLME shall issue an MLME-DISASSOCIATE.confirm primitive to inform the SME of the disassociation.
d) Upon receiving a MLME-DISASSOCIATE.confirm primitive, the SME shall delete any PTKSA and temporal keys held for communication with the STA by using the MLME-DELETEKEYS.request primitive (see 11.5.15) and by invoking MLME-SETPROTECTION.request(None).
e) Upon receiving an MLME-DISASSOCIATE.confirm primitive, the SME shall release the AID assigned for the indicated STA, if the state for the indicated STA was State 3 or State 4.
f) The SME shall inform the DS of the disassociation.

10.3.5.9 AP disassociation receipt procedure

Upon receipt of a Disassociation frame from a STA for which the state is State 3 or State 4, if management frame protection was not negotiated when the PTKSA(s) were created, or if MFP is in use and the frame is not discarded per MFP processing, the AP shall disassociate the STA using the following procedure:

a) The state for the STA shall be set to State 2.
b) The MLME shall issue an MLME-DISASSOCIATE.indication primitive to inform the SME of the disassociation.

c) Upon receiving a MLME-DISASSOCIATE.indication primitive the SME shall delete any PTKSA and temporal keys held for communication with the STA by using the MLME-DELETEKEYS.request primitive (see 11.5.15) and by invoking MLME-SETPROTECTION.request(None).
d) The SME shall inform the DS of the disassociation.
e) The SME shall release the AID assigned for the indicated STA.

10.3.6 Additional mechanisms for an AP collocated with a mesh STA

If a STA to AP mapping is added to the DS with the STA being a non-AP STA in an infrastructure BSS and the AP being an access point that interconnects through a DS with a mesh gate (that is, AP and mesh gate are collocated; see Figure 4-9), this mesh gate shall verify that the MAC address of the STA does not belong to a mesh STA in the MBSS. See 4.5.3.3 and Annex R for association and STA to AP mapping in the DS. If the mesh gate determines that the authenticated STA has a MAC address that is a MAC address of a mesh STA in the MBSS, then the collocated access point shall deauthenticate the STA with Reason Code “unspecified reason” or “MACADDRESS-ALREADY-EXISTS-IN-MBSS.”

The mechanism for verifying the MAC address of the authenticated STA depends on the active path selection protocol and might be vendor specific. See 13.10.13 when HWMP is the active path selection protocol.

10.4 TS operation

10.4.1 Introduction

A TSPEC describes the traffic characteristics and the QoS requirements of a TS. The main purpose of the TSPEC is to reserve resources within the HC and, in the case of HCCA and HEMM access policies, to modify the HC’s scheduling behavior. It also allows other parameters to be specified that are associated with the TS, such as a traffic classifier and acknowledgment policy.

A TS may have one or more TCLAS (within the discretion of the STA that sets up the stream) associated with it. The AP uses the parameters in the TCLAS elements to filter the MSDUs belonging to this TS for delivery as part of the TS.

TS may have zero or one Expedited Bandwidth Request (EBR) element associated with it. An AP uses the parameters in the EBR to understand the precedence level requested by a non-AP STA (see V.5.4). For example, the precedence level may be used to convey to the AP that the requested TS is for the purposes of placing an emergency call. Support for precedence levels greater than 18 is optional for STAs.

TSPEC, optional TCLAS, and optional EBR elements are transported on the air by the ADDTS Request frame and the ADDTS Response frame, and across the MLME SAP by the MLME-ADDTS primitives. In addition, a TS could be created if a STA sends a resource request to an AP prior to initiating a transition to that AP or in the Reassociation Request frame to that AP.

Following a successful negotiation, a TS is created, identified within the non-AP STA by its TSID and direction, and identified within the HC by a combination of TSID, direction, and STA address.

It is always the responsibility of the non-AP STA to initiate the creation of a TS regardless of its direction.

In the direct-link or TDLS direct-link case, it is the responsibility of the STA that is going to send the data to create the TS. In this case, the STA negotiates with the HC to gain TXOPs that it uses to send the data. There is no negotiation between the originator and recipient STAs concerning the TS: the originator can discover the capabilities of the recipient (rates, BlockAck) using the DLS.

In the case of traffic relayed by an AP, the sending and receiving STAs may both create individual TS for the traffic. Any traffic classifier created for the downlink TS applies equally regardless of whether the source is in the same BSS or reached through the DS.

A STA may simultaneously support up to eight TSs from the HC to itself and up to eight TSs from itself to other STAs, including the HC. The actual number it supports may be less due to implementation restrictions.

An HC may simultaneously support up to eight downlink TSs and up to eight uplink TSs per associated STA. The actual number it supports may be less due to implementation restrictions.

The traffic admitted in the context of a TSPEC can be sent using EDCA, HCCA, or HEMM. This depends on the access policy set in the TS Info field in the TSPEC. A TSPEC request may be set so that both HCCA and EDCA mechanisms (i.e., HEMM) are used.

When dot11SSPNInterfaceActivated is true, TSPEC processing by the HC may be subject to limitations received from the SSPN interface. The SSPN may limit access to certain QoS priorities, and further restrict the data rate and delay used with any priority.

10.4.2 TSPEC construction

TSPECs are constructed at the SME, from application requirements supplied via the SME, and with information specific to the MAC layer. Except as described in this subclause, there are no normative requirements on how any TSPEC is to be generated. However, N.3.2, describes parameter choice.

The value of the Minimum PHY Rate in a TSPEC shall satisfy the following constraints:

- it is in the AP's operational rate set, for an uplink TS.
- it is in the non-AP STA's operational rate set, for a downlink TS.
- it is in both the AP's operational rate set and non-AP STA's operational rate set, for a bidirectional TS.

10.4.3 TS life cycle

Figure 10-7 summarizes the TS life cycle (using the HMSC syntax defined in ITU-T Recommendation Z.120 (2004).

Initially a TS is inactive. A STA shall not transmit any QoS data frames in which the TID subfield of the QoS Control field matches an inactive TS.

A TS may be established by a Resource Request appearing in a message as part of a fast BSS transition from a STA. Such a TS is created in the accepted state. If the STA subsequently reassociates with this AP, then the TS becomes active. If the STA does not reassociate prior to the expiration of the reassociation timeout, then the TS becomes inactive.

Following a successful TS setup initiated by a non-AP STA, the TS becomes active, and either the non-AP STA or the HC may transmit QoS data frames whose TID contains this TSID (according to the Direction field). In the case of EDCA, the TID contains the UP value.

While the TS is active, the parameters of the TSPEC characterizing the TS can be renegotiated, when the renegotiation is initiated by the non-AP STA. This negotiation might succeed, resulting in a change to the TSPEC, or might fail, resulting in no change to the TSPEC.

An active TS becomes inactive following a TS deletion process initiated at either non-AP STA or HC. It also becomes inactive following a TS timeout detected at the HC, or if the HC within an AP when

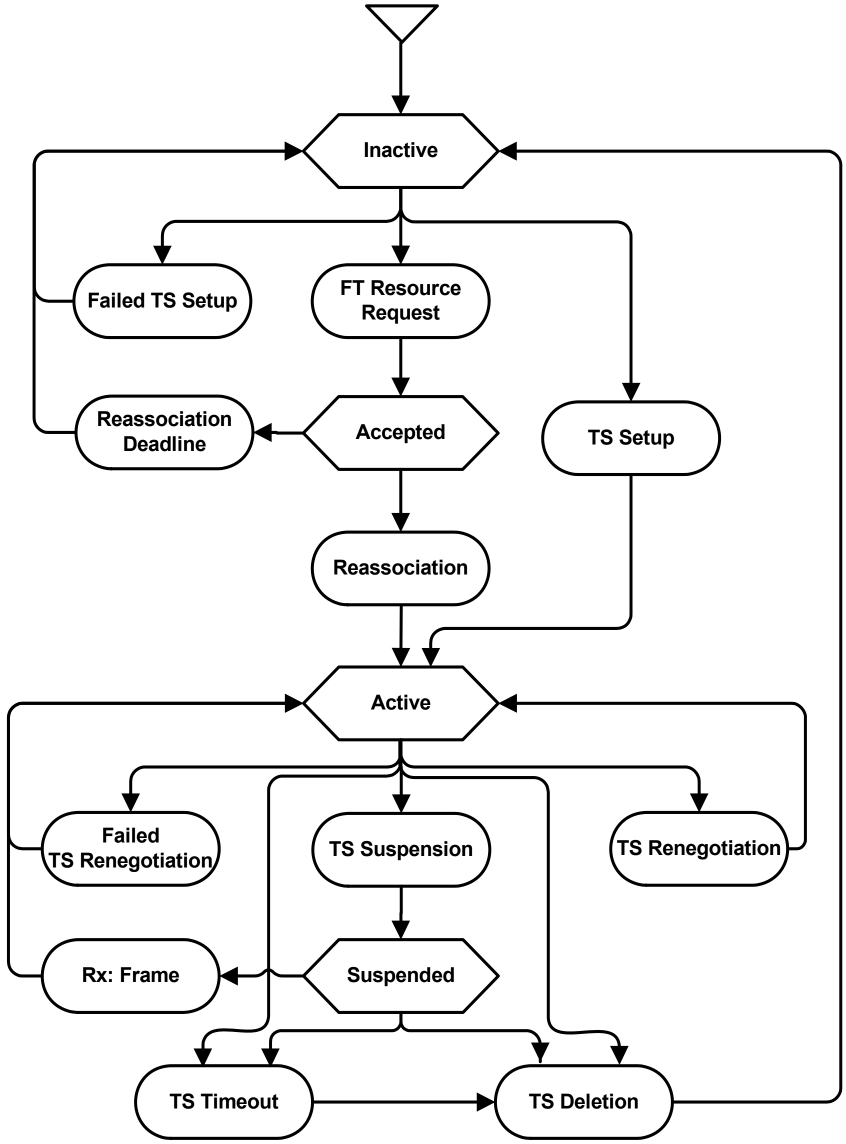


Figure 10-7—TS life cycle

dot11SSPNInterfaceActivated is true determines as defined in 10.24.5 that the non-AP STA's TS has exceeded the transmitted MSDU limit for the access category in which the TS was admitted. When an active TS becomes inactive, all the resources allocated for the TS are released.

An active TS may become suspended if no activity is detected for a duration of a suspension interval. Upon detection of activity, the TS may be reinstated. While the TS is in the suspended state, the HC shall not reclaim the resources assigned to the TS.

10.4.4 TS setup

Figure 10-8 shows the sequence of messages occurring at a TS setup. This message sequence in this figure and in the subsequent figures does not show the acknowledgment.

The non-AP STA's SME decides that a TS needs to be created. How it does this, and how it selects the TSPEC parameters, is beyond the scope of this standard. The SME generates an MLME-ADDTS.request primitive containing a TSPEC. A TSPEC may also be generated autonomously by the MAC without any

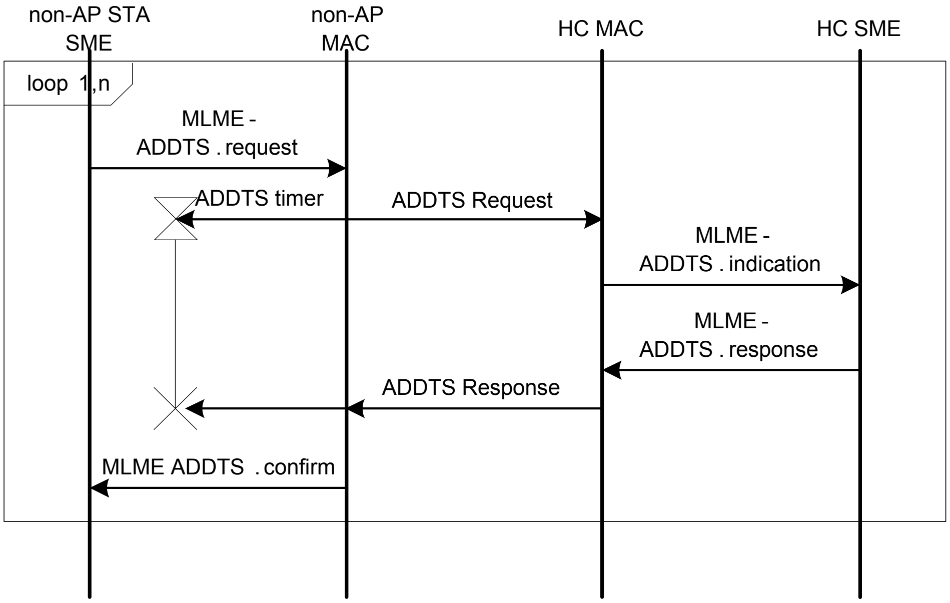


Figure 10-8—TS setup

initiation by the SME. However, if a TSPEC is generated subsequently by the SME, the TSPEC containing the same TSID generated autonomously by the MAC shall be overridden. If one or more TSPECs are initiated by the SME, the autonomous TSPEC shall be terminated.

The STA's MLME transmits the TSPEC in an ADDTS Request frame to the HC and starts a response timer called *ADDTS timer* of duration dot11ADDTSResponseTimeout.

The HC's MLME receives this management frame and generates an MLME-ADDTS.indication primitive to its SME containing the TSPEC.

The SME in the HC decides whether to admit the TSPEC with its associated TCLAS element(s) (if present) and TCLAS processing element (if present), as specified, refuse the TSPEC, or not admit but suggest an alternative TSPEC or TCLAS element(s) or TCLAS processing element. If the TSPEC is received from a non-AP STA by an AP when dot11SSPNInterfaceActivated is true, the HC shall use the permissions stored in dot11InterworkingEntry for that STA in the decision to admit or deny the request (see 10.24.5.3). The SME then generates an MLME-ADDTS.response primitive containing the TSPEC, zero or more TCLAS element(s) (only if present in the request) and TCLAS processing element (only if present in the request) and a ResultCode value. The contents of the TSPEC field, TCLAS element(s) (if present), TCLAS processing element (if present), and ResultCode field contain values specified in 6.3.26.5.2. The SME may include fewer TCLAS elements in the MLME-ADDTS.response primitive than were present in the request; when the SME's response includes a single TCLAS element, it shall not include a TCLAS processing element. If the SME changes a TCLAS element's Classifier Type field in the MLME-ADDTS.response primitive but is unable to suggest a value for the Classifier Mask field, it shall set that field to 0. If the SME changes a TCLAS element's Classifier Type field or Classifier Mask field in the MLME-ADDTS.response primitive but is unable to suggest values for one or more Classifier Parameter subfields, it shall set those subfields to 0.

When the HC in an AP that has dot11SSPNInterfaceActivated equal to true receives a TSPEC, the AP shall inspect it to determine the requested access policy, user priority, and mean data rate as follows:

a) The access category shall be determined from the user priority according to Table 9-1. For a TS to be admitted when the requested access policy is EDCA, both of the following shall be true:

1) The field corresponding to this access category in dot11NonAPStationAuthAccessCategories from the non-AP STA's dot11InterworkingEntry is equal to 1.
2) The sum of the mean data rate of all the requesting STA's active TSs in this access category plus the mean data rate in the TSPEC is less than or equal to the non-AP STA's dot11InterworkingEntry for dot11NonAPStationAuthMaxVoiceRate, dot11NonAPStationAuthMaxVideoRate, dot11NonAPStationAuthMaxBestEffortRate, or dot11NonAPStationAuthMaxBackgroundRate depending on whether the derived access category is AC_VO, AC_VI, AC_BE or AC_BK, respectively.

b) For a TS to be admitted when the requested access policy is HCCA, all of the following shall be true:
1) The dot11NonAPStationAuthHCCAHEMM value is true.
2) The sum of the mean data rate of all the requesting STA's active TSs having access policy set to HCCA plus the mean data rate in the TSPEC is less than or equal to dot11NonAPStationAuthMaxHCCAHEMMRate in the non-AP STA's dot11InterworkingEntry.
3) The delay bound that will be provided by the HC in the TSPEC response is less than or equal to dot11NonAPStationAuthHCCAHEMMDelay in the non-AP STA's dot11InterworkingEntry.

The HC's MLME transmits an ADDTS Response frame containing this TSPEC and status. The encoding of the ResultCode values to Status Code field values is defined in Table 8-37.

The STA's MLME receives this management frame and cancels its ADDTS timer. It generates an MLME-ADDTS.confirm primitive to its SME containing the TSPEC and status.

The SME decides whether the response meets its needs. How it does this is beyond the scope of this standard. A SME receiving a modified TCLAS element having a Classifier Mask field equal to 0 or Classifier Parameter subfields equal to 0 should regard these values as meaning that no suggested value has been provided by the HC.

— If the result code is SUCCESS, the TS enters into an active state.
— If the result code is REJECTED_WITH_SUGGESTED_BSS_TRANSITION, the non-AP STA may try to transition to other BSSs. In case that the non-AP STA is recommended to transition to other BSSs, it should do so according to the process defined in 10.23.6. Once the transition is completed, it should proceed with a TS setup process with the new HC.
— Otherwise, the whole process can be repeated using the same TSID and direction, a modified TSPEC, optional TCLAS element(s), and an optional TCLAS processing element until the SME decides that the granted medium time is adequate or inadequate and cannot be improved.

The parameters that are set for a TS may be renegotiated in a similar manner, when such a request is generated by the SME through ADDTS.request primitive. When a request for the modification of the TS parameters is accepted by the HC, it shall reset both the suspension interval and the inactivity interval timers.

If the HC grants medium time for an ADDTS Request frame with the Ack Policy subfield equal to Block Ack and the Direction field equal to either downlink or bidirectional, then it shall initiate a Block Ack negotiation by sending an ADDBA Request frame to the STA that originated the ADDTS Request frame. If a STA is granted medium time for an ADDTS Request frame with the Ack Policy subfield equal to Block Ack and the Direction field equal to other than downlink, then it shall initiate a Block Ack negotiation by sending an ADDBA Request frame to the recipient of the TS.

The combination of the TSID and Direction subfields identify the TS, in the context of the STA, to which the TSPEC applies. A bidirectional link request is equivalent to a downlink TS and an uplink TS, each with the same TSID and parameters.

The same TSID may be used for multiple TSs at different STAs. A STA can use the same TSID subfield value for a downlink TSPEC and either an uplink or a direct-link TSPEC at the same time. A non-AP STA shall not use the same TSID for both uplink and direct-link TS.

If the TS Info Ack Policy subfield of a TSPEC element is Block Ack and the type of Block Ack policy is unknown to the HC, the HC assumes, for TXOP scheduling, that the immediate Block Ack policy is being used (see 9.21).

An HC shall be capable of receiving an ADDTS request frame that contains a TCLAS element and capable of generating an indication that contains this as a parameter.

When a STA requests service at a higher priority than authorized by its dot11InterworkingTableEntry, the HC may optionally provide a suggested TSPEC with a data rate and lower priority that would be authorized. Usage of the TSPEC in an Interworking environment is described in Annex N.

10.4.5 TS setup by resource request during a fast BSS transition

A QoS STA may transmit a TSPEC as part of a RIC-Request in a resource request message. The SME in the hybrid coordinator (HC) decides whether to accept the TSPEC as specified, or refuse the TSPEC, or not accept but suggest an alternative TSPEC. It then generates a RIC-Response, according to the procedures given in 12.11.

Each TS established by this resource request is placed in the accepted state. This state is an intermediate state between inactive and active. In the accepted state, the inactivity and suspension timers shall not be started for the TS. For a TS based on hybrid coordination function (HCF) controlled channel access (HCCA), the HC shall not generate CF-Poll for the TS.

The SME may take the resource/timing requirements of the TS in the accepted state into consideration before assigning any further resources to any other admitted or accepted TS, and in calculating the available admission capacity for the BSS Load element.

The TS is moved to the active state once the STA performs a reassociation to the AP (see 12.11.3). Once the TS becomes active, the inactivity and suspension timers are started.

If the reassociation timer times out and the TS is not yet in the active state, the TS goes back to the inactive state.

10.4.6 PSMP management

A STA may attempt to create a scheduled PSMP session with its AP only if the AP has the S-PSMP Support field in the Extended Capabilities element equal to 1.

The TSPEC reserves resources within the AP and modifies the AP's scheduling behavior. The parameters in the TSPEC can be grouped into two categories: PSMP scheduling and PSMP reservation. The scheduling parameters are used by the AP to schedule a suitable SI. The reservation parameters are used by the AP to reserve time in the PSMP-UTT and PSMP-DTT.

The service start time and SI specify the PSMP schedule in the response's Schedule element. All other parameters result in a reservation for the PSMP-UTT and PSMP-DTT within the scheduled PSMP sequence.

An AP shall terminate the PSMP session only when the last TS associated with the particular PSMP session is terminated.

Once created, a PSMP session can be extended by another TSPEC setup. A STA that has an established PSMP session may issue an additional TSPEC request with the following:

- The Aggregation field set to 1
- The Scheduled field set to 1 and APSD field set to 0 (S-PSMP)
- The Minimum Service Interval and Maximum Service Interval fields both set to the Service Interval field value from the Schedule element specified when the PSMP session was established

The AP shall return an identical Schedule element for all TSPEC response frames related to the same PSMP session.

NOTE—A STA that does not have an established PSMP session might send a TSPEC request specifying S-PSMP session with the same SI. The AP is free to choose between aggregating this request with an existing PSMP session of the same SI or creating a new PSMP session.

The parameters of a TS already associated with the PSMP session may be changed; however, the SI shall not be changed. The start time of existing STAs in the PSMP schedule shall not be changed by the addition of a TSPEC from a new STA.

A TSPEC that reserves resources for a STA under scheduled PSMP shall have the APSD and Scheduled fields set to indicate Scheduled PSMP as defined in Table 8-110.

The non-AP STA's SME decides that a PSMP session needs to be established. How it makes this decision and how it selects the related TSPEC parameters are beyond the scope of this standard.

The Minimum Service Interval field of the TSPEC element of an ADDTS request frame shall be a multiple of the SI granularity indicated by the AP in its Extended Capabilities element.

10.4.7 Failed TS setup

There are two possible types of failed TS setup:

a) The transmission of ADDTS Request frame failed.
b) No ADDTS Response frame is received from the HC (e.g., because of delay due to congestion or because the response frame cannot be transmitted).

Figure 10-9 summarizes the remaining two cases. The MLME shall issue an MLME-ADDTS.confirm primitive, with a result code of TIMEOUT. In either case, if the request is not for an existing TS, the MLME shall send a DELTS to the HC specifying the TSID and direction of the failed request just in case the HC had received the request and it was the response that was lost.

10.4.8 Data transfer

After the setup of a TSPEC, MSDUs are classified above the MAC and are sent to the MAC through the MAC_SAP using the MA-UNITDATA.request primitive with the priority parameter encoded to the TID.

The generation of the associated TID is done by a classifier above the MAC, and it may use the associated TCLAS elements if any are present. When there are multiple TCLASs and if the Processing subfield of TCLAS Processing element is 0, the priority parameter in the associated MA-UNITDATA.request primitive is set to TID if the classifier matches the parameters in all the TCLAS elements associated with the TS. When there are multiple TCLASs and if the Processing subfield of the TCLAS Processing element is 1, the priority parameter in the associated MA-UNITDATA.request primitive is set to TID if the classifier matches the parameters in at least one of the TCLAS elements associated with the TS. When there is no TCLAS element and if the Processing subfield of the TCLAS Processing element is 2, the priority parameter in the associated MA-UNITDATA.request primitive is set to TID if the classifier cannot match the parameters to

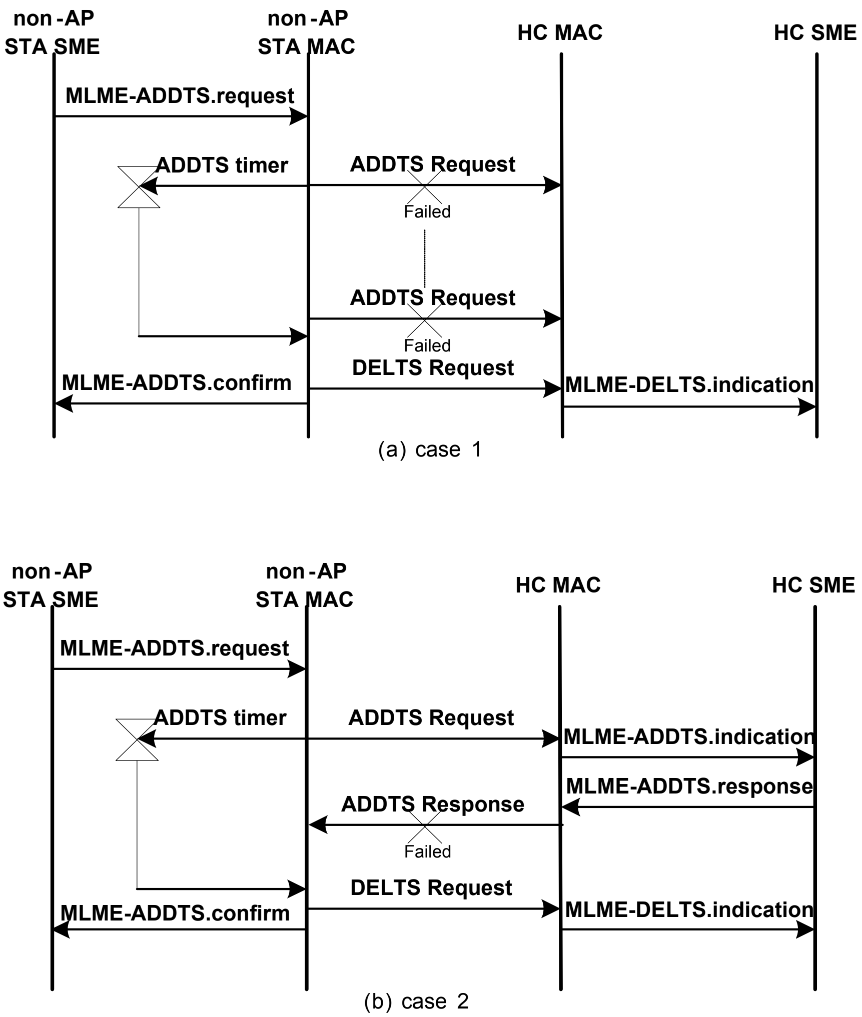


Figure 10-9—Failed TS setup detected within non-AP STA's MLME

any of the TCLAS elements. See 5.1.1.4 for the treatment of an MSDU with a TSID for which there is no associated TSPEC.

When an MSDU is classified using a TCLAS element, the original UP cannot be recovered by the receiver.

10.4.9 TS deletion

There are two types of TS deletion: non-AP STA-initiated and HC-initiated. In both cases, the SME entity above the MAC generates an MLME-DELTS.request primitive specifying the TSID and direction of the TS to be deleted and the reason for deleting the TS. This causes the MAC to send a DELTS frame. The encoding of ReasonCode values to Reason Code field (see 8.4.1.7) values is defined in Table 8-36.

The TS is considered inactive within the initiating MAC when the ACK frame to the Action frame is received. No Action frame response is generated.

Figure 10-10 shows the case of TS deletion initiated by the non-AP STA and the case of TS deletion initiated by the HC.

An HC should not delete a TSPEC without a request from the SME except due to inactivity (see 10.4.10) or an HC service change that precludes the HC from continuing to meet the requirements of the TSPEC.

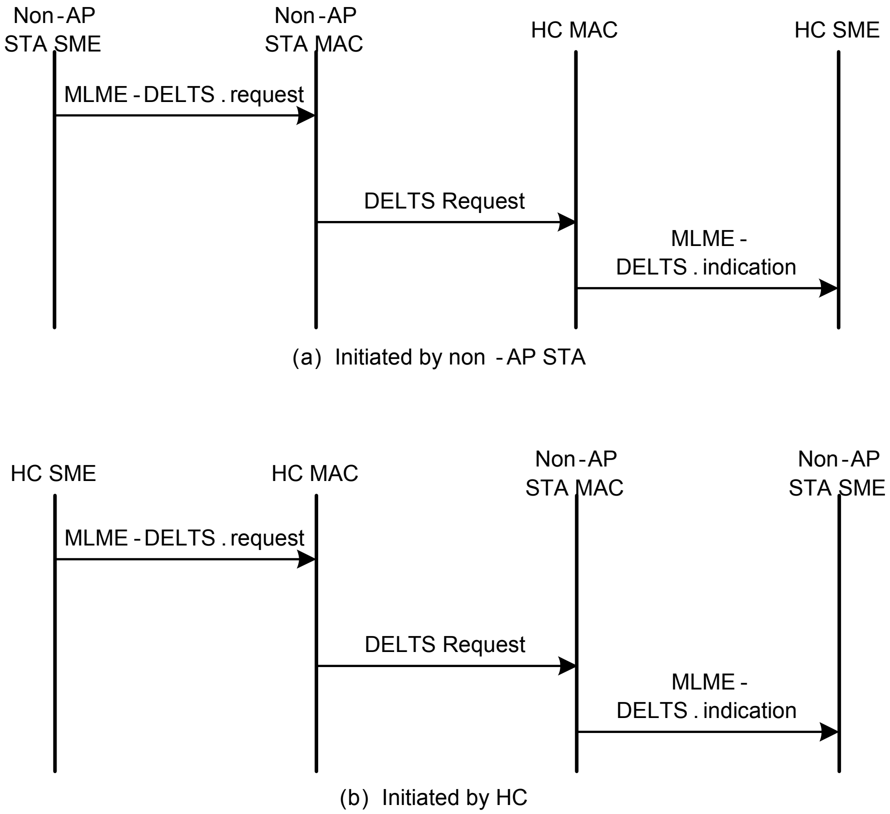


Figure 10-10—TS deletion

All TSPECs that have been set up shall be deleted upon disassociation and reassociation. Reassociation causes the non-AP STA and AP to clear their state, and the non-AP STA has to reinitiate the setup.

10.4.10 TS timeout

TS timeout is detected within the HC's MAC when no traffic is detected on the TS within the inactivity timeout specified when the TS was created.

For an uplink TS, the timeout is based on the arrival of correctly received MSDUs that belong to the TS, after any decryption and reassembly.

For a downlink TS, the timeout is based on the following:

- Arrival of valid MA-UNITDATA.request primitives using this TS at the MAC_SAP when the QoS data frames are sent with the Ack Policy subfield equal to No Ack.
- Confirmation of correctly sent MSDUs that belong to the TS within the MAC when the QoS data frames are sent with the Ack Policy subfield set other than to No Ack.

For a direct-link TS, inactivity is considered to have happened if one of the two following events occurs:

- The HC transmits a QoS CF-Poll frame and the polled STA returns a QoS Null immediately after a SIFS interval that contains a QoS Control field in which the Queue Size subfield contains 0.
- The HC transmits a QoS CF-Poll frame, and no QoS Null frame is received within the granted TXOP duration that indicates the queue size for the related TSID. This is to ensure that the STA is actually using the assigned TXOP for the given TSID.

Any other use of a polled TXOP delivered to the STA is considered to be activity on all direct-link TS associated with that STA. Detection of inactivity of this type is optional.

In response to an inactivity timeout, the HC shall send a DELTS frame to the STA with the result code set to TIMEOUT and inform its SME using the MLME-DELTS.indication primitive.

The case of uplink TS timeout is shown in Figure 10-11.

10.4.11 TS suspension

TS suspension occurs within the HC's MAC when no traffic is detected on the TS within the suspension interval specified when the TS was created. In the suspended state, the generation of QoS (+)CF-Poll frames is stopped for the related TS.

10.4.12 TS Reinstatement

A suspended TS may be reinstated following activity for that TS.

For uplink and direct link, a suspended TS may be reinstated by a STA by sending a QoS data or QoS Null frame. This frame may be sent at the highest priority using EDCA. The generation of successive QoS (+)CF-Poll frames shall then be resumed by the HC.

For downlink, a suspended TS is reinstated by the HC when the AP receives an MSDU from a higher layer.

10.5 Block Ack operation

10.5.1 Introduction

Block Ack may be set up at the MAC (see 9.21.2) or by the initiation of SME. The setup and deletion of Block Ack at the initiation of the SME is described in this subclause.

10.5.2 Setup and modification of the Block Ack parameters

10.5.2.1 General

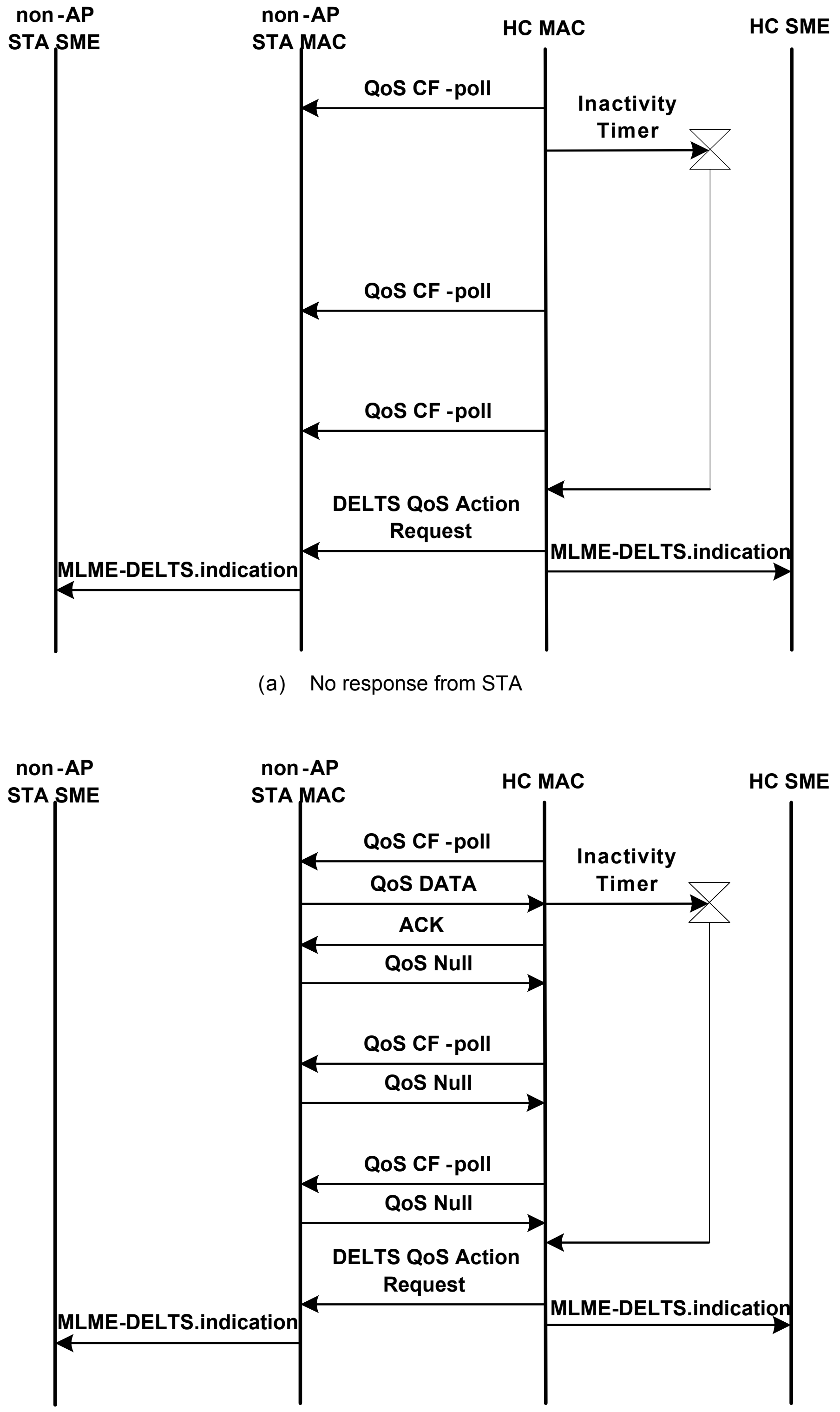
The procedures for setting up and modifying the Block Ack parameters are described in 10.5.2.2 and 10.5.2.3, respectively, and illustrated in Figure 10-12.

10.5.2.2 Procedure at the originator

Upon receipt of an MLME-ADDBA.request primitive, an initiating STA that intends to send QoS data frames under the Block Ack mechanism shall set up the Block Ack using the following procedure:

a) If the initiating STA is an HT STA, is a member of an IBSS, and has no other existing Block Ack agreement with the recipient STA, then the initiating STA shall transmit a Probe Request frame to the recipient STA and shall not transmit an ADDBA Request frame unless it receives a Probe Response frame from the recipient within dot11ADDBAFailureTimeout.

   NOTE—When the Block Ack agreement is being established between a non-AP STA and its AP, then the originator and the recipient have exchanged capability information during the association exchange that allows them to determine whether the other STA is an HT STA. If the STA is establishing a Block Ack agreement with another STA through DLS, then the DLS setup procedure includes the exchange of capability information that allows both STAs to determine whether the other STA is an HT STA.



(a) No response from STA

(b) No Data frames from STA

Figure 10-11—TS timeout

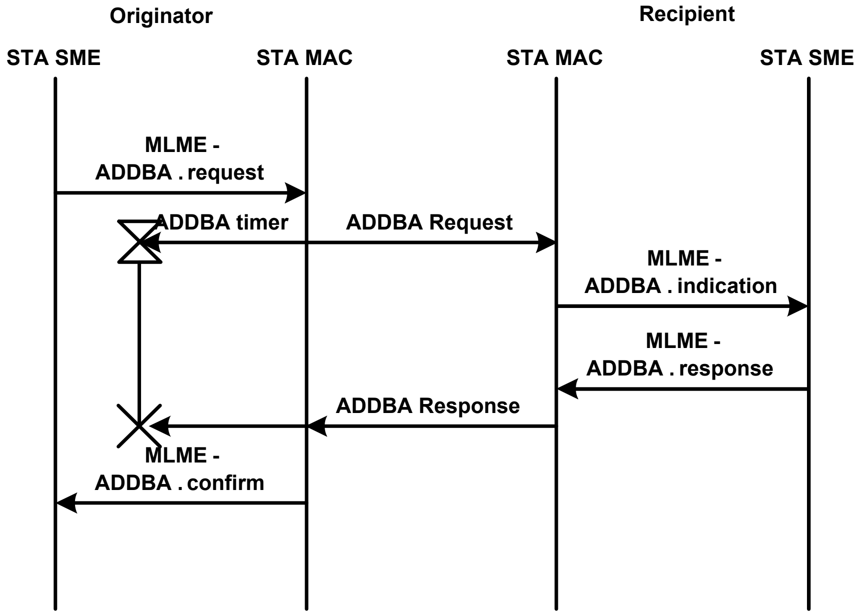


Figure 10-12—Block Ack setup

b) Check whether the intended peer STA is capable of participating in the Block Ack mechanism by discovering and examining its "Delayed Block Ack" and "Immediate Block Ack" capability bits. If the recipient is capable of participating, the originator sends an ADDBA frame indicating the TID and the buffer size.

c) If an ADDBA Response frame is received with the matching dialog token and the TID and with a status code equal to 0, the STA has established a Block Ack mechanism with the recipient STA; and the MLME shall issue an MLME-ADDBA.confirm primitive indicating the successful completion of the Block Ack setup.

d) If an ADDBA Response frame is received with the matching dialog token and the TID and with a status code not equal to 0, the STA has not established a Block Ack mechanism with the recipient STA; and the MLME shall issue an MLME-ADDBA.confirm primitive indicating the failure of the Block Ack setup.

e) If there is no response from the recipient within dot11ADDBAFailureTimeout, the STA has not established a Block Ack mechanism with the recipient STA; and the MLME shall issue an MLME-ADDBA.confirm primitive with a result code of TIMEOUT.

10.5.2.3 Procedure at the recipient

A recipient shall operate as follows in order to support Block Ack initialization and modification:

a) When an ADDBA Request frame is received from another STA, the MLME shall issue an MLME-ADDBA.indication primitive.

b) Upon receipt of the MLME-ADDBA.response primitive, the STA shall respond by an ADDBA Response frame with a result code as defined in 8.5.5.3.

   1) If the result code is SUCCESS, the Block Ack is considered to be established with the originator. Contained in the frame are the type of Block Ack and the number of buffers that have been allocated for the support of this block.

   2) If the result code is REFUSED, the Block Ack is not considered to have been established.

The encoding of ResultCode values to Status Code field values is defined in Table 8-37.

10.5.2.4 Procedure common to both originator and recipient

Once a Block Ack agreement has been successfully established between two STAs, the type of agreement thus established is dependent on the capabilities of the STAs and the contents of the ADDBA frames used to establish this agreement as defined in Table 10-2.

Table 10-2—Types of Block Ack agreement based on capabilities and ADDBA conditions

Capabilities condition	ADDBA condition	Type of Block Ack agreement
One or both of the STA are non-HT.	Block Ack Policy subfield equal to 1	Immediate
	Block Ack Policy subfield equal to 0	Delayed
Both STAs are HT STAs.	Block Ack Policy subfield equal to 1	HT-Immediate
Both STAs are HT STAs, and both of the STAs set the HT-Delayed Block Ack subfield of the HT Capabilities element to 1.	Block Ack Policy subfield equal to 0	HT-Delayed
Both STAs are HT STAs, and at least one of the STAs sets the HT-Delayed Block Ack subfield of the HT Capabilities element to 0.	Block Ack Policy subfield equal to 0	Delayed

10.5.3 Teardown of the Block Ack mechanism

10.5.3.1 General

The procedure at the two STAs is described in 10.5.3.2 and 10.5.3.3 and illustrated in Figure 10-13.

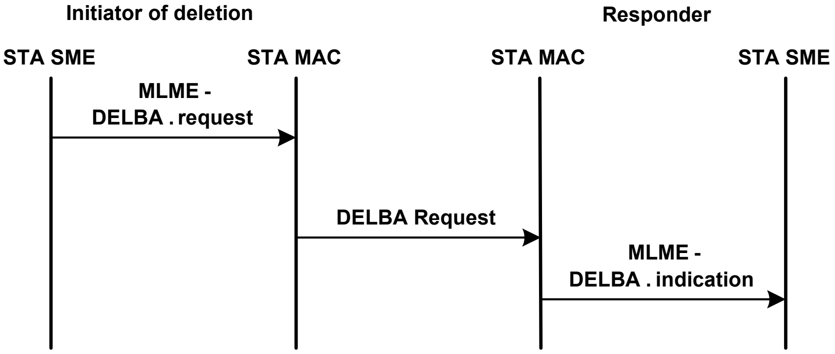


Figure 10-13—Block Ack deletion

10.5.3.2 Procedure at the initiator of the Block Ack teardown

Upon receipt of an MLME-DELBA.request primitive, the MLME shall tear down the Block Ack by transmitting a DELBA frame.

The encoding of ReasonCode values to Reason Code field (see 8.4.1.7) values is defined in Table 8-36.

10.5.3.3 Procedure at the recipient of the DELBA frame

A STA shall issue a MLME-DELBA.indication primitive with the parameter ReasonCode having a value of REQUESTED when a DELBA frame is received.

10.5.4 Error recovery upon a peer failure

Every STA shall maintain an inactivity timer for every negotiated Block Ack setup. The inactivity timer at a recipient is reset when MPDUs corresponding to the TID for which the Block Ack policy is set are received and the Ack Policy subfield in the QoS Control field of that MPDU header is Block Ack or Implicit Block Ack Request. The inactivity timer is not reset when MPDUs corresponding to other TIDs are received. The inactivity timer at the recipient is also reset when a BlockAckReq frame corresponding to the TID for which the Block Ack policy is set is received. The inactivity timer at the originator is reset when a BlockAck frame corresponding to the TID for which the Block Ack policy is set is received. When a timeout of BlockAckTimeout is detected, the STA shall send a DELBA frame to the peer STA with the Reason Code field set to TIMEOUT and shall issue a MLME-DELBA.indication primitive with the ReasonCode parameter having a value of TIMEOUT. The procedure is illustrated in Figure 10-14.

When a recipient does not have an active Block ack for a TID, but receives data MPDUs with the Ack Policy subfield equal to Block Ack, it shall discard them and shall send a DELBA frame within its own TXOP. If such a STA receives a BlockAckReq frame, it may respond with an ACK frame and shall respond with a DELBA frame within its own TXOP. The originator may attempt to set up the use of Block Ack or may send the MPDUs using an alternative acknowledgment mechanism. When the recipient transmits a DELBA frame, it shall set the last sequence number received value to the sequence number of the last received MPDU, regardless of the acknowledgment policy used in that frame. When the originator receives a DELBA frame, it shall

a) Discard any MPDU that has been transmitted and not acknowledged, with the possible exception if it was the last MPDU to be sent and it was not a retransmission, and
b) Set the sequence number to either that of the last MPDU that is sent if it intends to retransmit or one beyond the last MPDU sent.

The originator STA may send an ADDBA Request frame in order to update Block ACK Timeout Value. If the updated ADDBA Request frame is accepted, both STAs initialize the timer to detect Block ACK timeout. Even if the updated ADDBA Request frame is not accepted, the original Block ACK setup remains active.

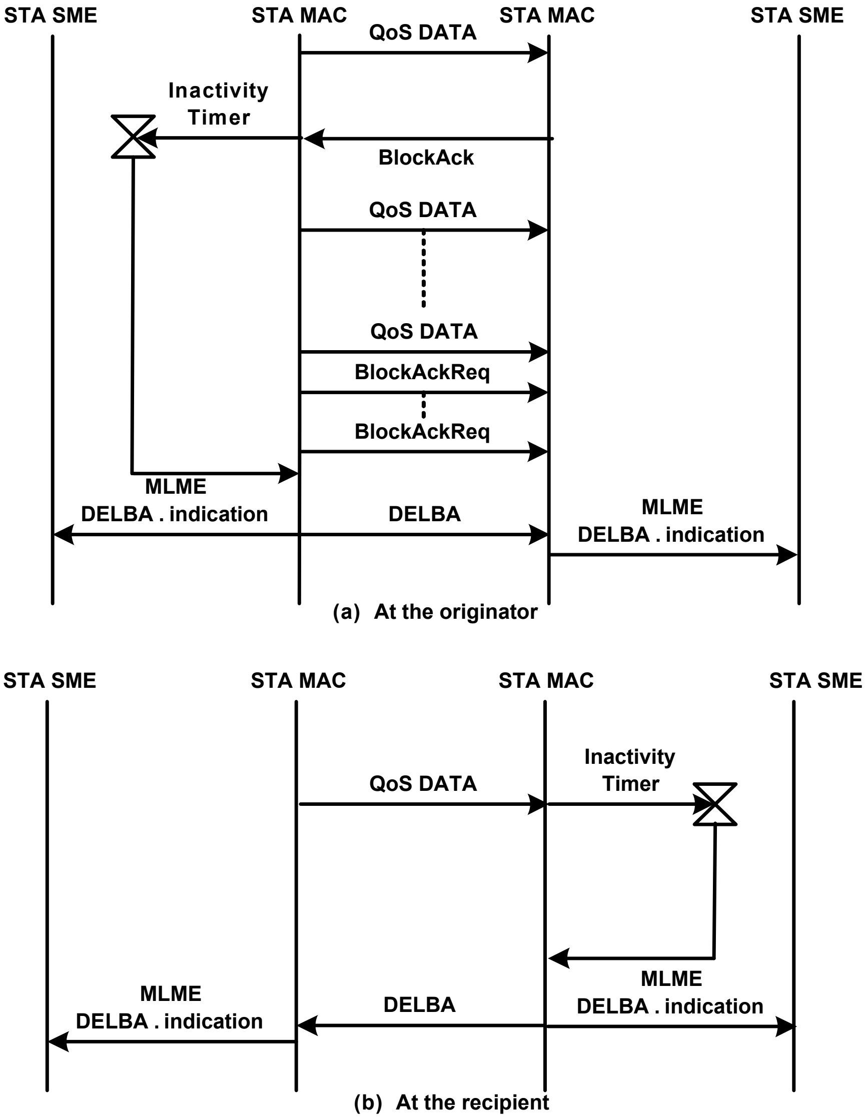
10.6 Higher layer timer synchronization

10.6.1 Introduction

Higher layer timer synchronization is beyond the scope of this standard. However, explanation of how the constructs in this standard might be used to support such capabilities might be useful to the designer.

One way to accomplish synchronization across a BSS is by multicasting synchronization (Sync) packets from the higher layers containing a time stamp and a sequence number. These packets would be opaque to the MAC and would be transported in the same way as any other MSDU (most likely addressed to the group address). Sync packets would be treated as a type of management packet by the higher layers. The time stamp in the Sync packet would contain the higher layer clock value at the time when the previous Sync packet was transmitted. The sequence number parameter has a value equal to the sequence number of the MSDU in which the time stamp is provided.

The reason the packet would contain the time stamp for the previous Sync packet (rather than the current packet) is that hardware and layering constraints would prohibit the ability to provide a time stamp for the



(a) At the originator

(b) At the recipient

Figure 10-14—Error recovery by the receiver upon a peer failure

exact instant the current packet is transmitted within that packet. However, a MLME-HL-SYNC.indication primitive allows the transmitting STA to know exactly when each Sync packet is transmitted. A receiving STA might note the time when each Sync packet is received as well as the sequence number for that frame. The receiving STA would save this receive time indication for each packet along with its sequence number and compare the indication of the previously received Sync packet to the time stamp in the most currently received packet. This comparison allows the STA to compute an offset, which can be used to adjust its time reference to match that of the synchronization source. A check of the sequence number verifies that the correct packet is being used for time stamp comparison. It is possible a packet is lost; in this case, the received time stamp for the lost packet should be discarded.

The last symbol of the Sync frame is indicated by the PHY using the PHY-TXEND.confirm and PHY-RXEND.indication primitives in the transmitter and receiver of the Sync frame, respectively. Practical limits on the coincidence of this indication and the last symbol of the Sync frame are implementation dependent. The accuracy of this technique also depends on the propagation delay between the source and receiving channel. However, both the time difference (between the PHY indication and the last symbol of the Sync frame) and the propagation delay can be considered as fixed-delay components. Therefore, they contribute only to the fixed time difference between the transmitter and receiver STAs' clocks and do not contribute to jitter. An implementation-dependent scheme might be used to cancel or minimize this fixed time difference.

The Sync frame may also be relayed through the AP. In this case, the STA that generates the time stamps notes the reception of the group addressed Sync frame from the AP as the indication to save the higher clock value for the next Sync frame. Receiving STAs would also similarly note the time when each Sync packet is received from the AP. The sequence number would include a value corresponding to the frame received from the AP.

This is an example implementation using the higher layer timer synchronization feature. Other implementations are possible.

10.6.2 Procedure at the STA

In order to determine whether to provide an MLME-HL-SYNC.indication primitive for a particular data frame, a MAC that supports MLME-HL-SYNC primitives compares the Address 1 field in a data frame's MAC header to a list of group addresses previously registered by an MLME-HL-SYNC.request primitive. If the MAC and the transmitter of the Sync frame are collocated within the same STA, the MLME-HL-SYNC.indication primitive shall occur when the last symbol of the PPDU carrying a matching data frame is transmitted. Otherwise, the indication shall occur when the last symbol of the PPDU carrying the matching data frame is received. In both cases, the MLME-HL-SYNC.indication primitive provided is simultaneous (within implementation-dependent delay bounds) with the indication provided to other STAs within the BSS for the same data frame.

10.7 DLS operation

10.7.1 General

DLS is a protocol that enables a STA in an infrastructure network to transmit frames directly to another STA in the same infrastructure network. The need for this protocol is motivated by the fact that the intended recipient may be in PS mode, in which case it can be awakened only by the AP. The second feature of DLS is to exchange rate set and other information between the sender and the receiver.

This protocol prohibits the STAs going into PS mode for the duration of the direct stream as long as there is an active DLS between the two STAs.

DLS does not apply in an IBSS, where frames are always sent directly from one STA to another. DLS does not apply in an MBSS, because frames in an MBSS are always sent directly from one mesh STA to another.

The handshake involved in the setup is illustrated in Figure 10-15.

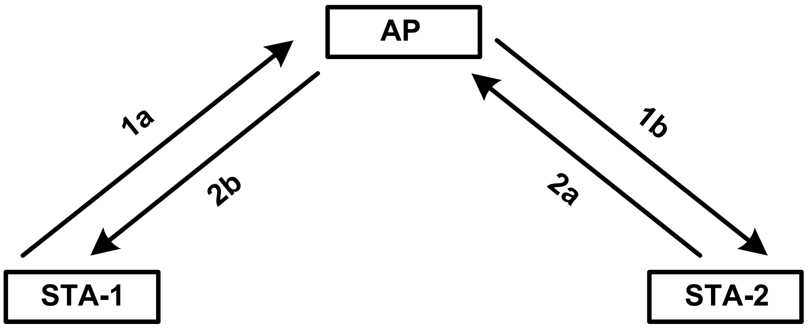


Figure 10-15—The four steps involved in direct-link handshake

The handshake involves the following steps:

a) A STA, STA-1, that intends to exchange frames directly with another non-AP STA and dot11DLSAllowed is true, STA-2, invokes DLS and sends a DLS Request frame to the AP (step 1a in Figure 10-15). This request contains the rate set, capabilities of STA-1, and the MAC addresses of STA-1 and STA-2. If STA-1 is an HT STA, this request also contains the HT capabilities of STA-1.

b) If STA-2 is associated in the BSS, direct streams are allowed in the policy of the BSS (as determined by dot11DLSAllowedInQBSS), and STA-2 is indeed a QoS STA, then the AP forwards the DLS Request frame, independently of whether the AP is capable of decoding all of the fields in the body of the frame, to the recipient, STA-2 (step 1b in Figure 10-15).

c) If STA-2 has dot11DLSAllowed true and accepts the direct stream, it sends a DLS Response frame to the AP (step 2a in Figure 10-15), which contains the rate set, (extended) capabilities of STA-2, and the MAC addresses of STA-1 and STA-2. If STA-2 is an HT STA, this response also contains an HT Capabilities element representing the HT capabilities of STA-2.

d) The AP forwards the DLS Response frame to STA-1 (step 2b in Figure 10-15), independently of whether the AP is capable of decoding all of the fields in the body of the frame.

e) If the DLS Response frame contained a status code of SUCCESSFUL, the direct link becomes active and frames may be sent from STA-1 to STA-2 and from STA-2 to STA-1.

When a STA transitions to a different AP after a DLS is set up, the DLS shall be torn down as described in 10.7.5.

10.7.2 DLS procedures

10.7.2.1 General

The DLS message flow is illustrated in Figure 10-16.

10.7.2.2 Setup procedure at the QoS STA

A STA shall maintain a list of all STAs with which a direct link has been established.

Upon receipt of an MLME-DLS.request primitive from the SME, the STA shall do one of the following actions:

— Issue an MLME-DLS.confirm primitive with a result code of SUCCESS if the peer MAC address of MLME-DLS.request primitive is in the list of STAs with which direct link has been established; or

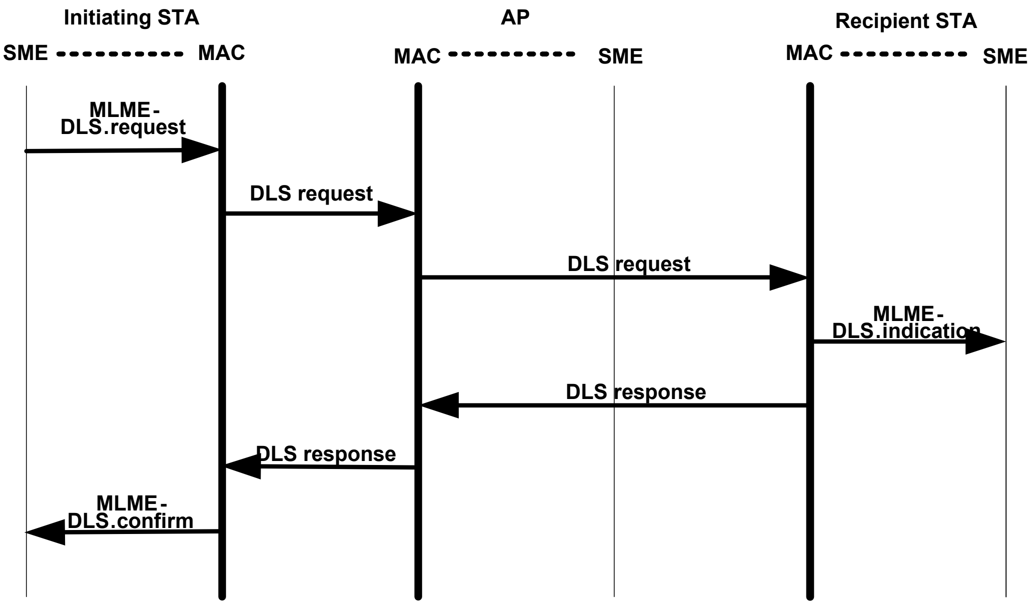


Figure 10-16—DLS message flow

- Initiate the setup of the direct link by sending the DLS Request frame to the AP (step 1a in Figure 10-15). If the STA does not receive a DLS Response frame within DLSResponseTimeout after sending the DLS Request frame, the STA shall issue an MLME-DLS.confirm primitive with the result code of TIMEOUT.

Upon receipt of the DLS Request frame from the AP (step 1b in Figure 10-15), the MLME shall send to the AP a DLS Response frame (step 2a in Figure 10-15) with a result code of

- SUCCESS if the STA is willing to participate in the direct link with the source STA. The STA shall also issue an MLME-DLS.indication primitive to the SME and shall add the source STA to the list of the STAs, if not already present, with which direct link has been established.
- REFUSED if the STA is not willing to participate in the direct link.

Upon receipt of the DLS Response frame from the AP (step 2b in Figure 10-15), the STA shall issue an MLME-DLS.confirm primitive.

- If the result code is SUCCESS, the direct link is considered to be established with the destination STA in the DLS Response frame, and the MAC shall add the destination STA to the list of STAs with which direct link has been established.
- If the result code is REFUSED, the direct links is not considered to have been established.

10.7.2.3 Setup procedure at the AP

Upon receipt of the DLS Request frame (step 1a in Figure 10-15), the AP shall

- Send DLS Response frame to the STA that sent the DLS Request frame with a result code of Not Allowed in the BSS, if direct links are not allowed in the BSS (step 2b in Figure 10-15), or for the AP with dot11SSPNInterfaceActivated equal to true with a result code of Not Allowed by SSP if the dot11NonAPStationAuthDls MIB variable in either of the non-AP STA's dot11InterworkingTable is false.
- Send DLS Response frame to the STA that sent the DLS Request frame with a result code of Not Present, if the destination STA is not present in the BSS (step 2b in Figure 10-15).

- Send DLS Response frame to the STA that sent the DLS Request frame with a result code of Not a QoS STA, if the destination STA does not have QoS facility (step 2b in Figure 10-15).
- Send the DLS Request frame, with all fields having the same value as the DLS Request frame received by the AP, to the destination STA (step 1b in Figure 10-15), independently of whether the AP is capable of decoding all of the fields in the body of the frame.

Upon receipt of the DLS Response frame from a STA (step 2a in Figure 10-15), the AP shall send DLS Response frame to the source STA (step 2b in Figure 10-15).

The mapping of Status Code field values to ResultCode parameter values in an MLME-DLS.confirm primitive is defined in Table 8-37.

10.7.2.4 Operation of the DLS Timeout Value field

The DLS Timeout Value field within the DLS Request frame contains the duration, in seconds, after which the direct link is terminated, if there are no frame exchanges within this duration with the peer. A value of 0 implies that the direct link is never to be terminated based on a timeout.

10.7.3 Data transfer after setup

For each active direct link, a STA shall record the MAC and PHY features, rates, and MCSs that are supported by the other STA participating in the direct link, according to the Supported Rates, Extended Supported Rates, Capability Information, and HT Capabilities fields within the DLS Request and DLS Response frames that were used to establish the direct link.

A STA transmitting frames within a direct link shall not transmit frames using features, rates, or MCSs that are not supported by the other STA in the direct link. After establishing protection as required by 9.23 or 9.3.2.7, STAs may use features, rates, or MCSs that are supported by both of the STAs in the direct link, even when the AP does not support those features, except for transmission of a 40 MHz mask PPDU, which is governed by the rules found in 10.15.4.

STAs participating in a direct link may set up a Block Ack agreement, if needed. The STAs may set up TSs with the HC to provide enough bandwidth or use polled TXOPs for data transfer. A protective mechanism (such as transmitting using HCCA, RTS/CTS, or the mechanism described in 9.23) should be used to reduce the probability of other STAs interfering with the direct-link transmissions.

10.7.4 DLS teardown

10.7.4.1 General

The DLS teardown procedure is divided into STA-initiated and AP-initiated DLS teardown. The STA-initiated DLS teardown message flow is illustrated in Figure 10-17.

10.7.4.2 STA-initiated DLS teardown procedure

Upon receipt of MLME-DLSTeardown.request primitive from the SME, the STA shall initiate the teardown of the direct link by sending the DLS Teardown frame to the AP. The applicable values of ReasonCode are defined in Table 10-3. The encoding of the Reason Code field is defined in Table 8-36.

Upon receipt of the DLS Teardown frame (from the AP), the STA shall issue an MLME-DLSTeardown.indication primitive to the SME and shall delete the STA from the list of the STAs with which direct link has been established.

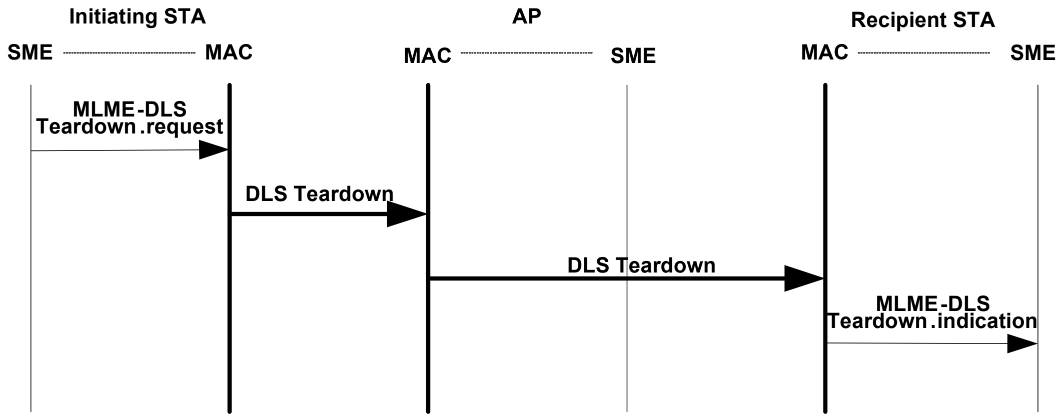


Figure 10-17—STA-initiated DLS teardown message flow

Table 10-3—ReasonCode values for DLS teardown

ReasonCode	Applicable at
STA_LEAVING	Non-AP STA
END_DLS	Non-AP STA
UNKNOWN_DLS	Non-AP STA
TIMEOUT	Non-AP STA
PEERKEY_MISMATCH	AP
PEER_INITIATED	AP and Non-AP STA
AP_INITIATED	Non-AP STA

The DLS teardown procedure shall apply to a specific DLS session, as each STA may have multiple simultaneous DLS sessions with other STAs.

Prior to disassociation/deauthentication from the AP, the STA (STA1) shall initiate the teardown of any direct links it has by sending a DLS Teardown frame to the AP. If the DLS Teardown frame's Max Retry Limit was reached with no response from the AP, the STA shall send a DLS Teardown frame to its peer DLS STA (STA2).

A recipient STA (STA2), either on expiry of its DLSResponseTimeout or on receipt of a DLS Teardown frame with ReasonCode equal to PEER_INITIATED (from STA1), shall send a DLS Teardown frame to the AP with ReasonCode set to PEER_INITIATED.

An AP that receives a DLS Teardown frame with ReasonCode equal to PEER_INITIATED should send a DLS Teardown frame to any STAs that have a DLS link established with STA1.

NOTE—The failed STA can reestablish its DLS link according to 10.7.

10.7.4.3 Teardown procedure at the AP

Upon receipt of the DLS Teardown frame from a STA, the AP shall send a DLS Teardown frame to the destination STA.

Upon receipt of MLME-DLSTeardown.request primitive from the SME, the AP shall announce the tearing down of the direct link by sending the DLS Teardown frame to the two STAs using the direct link. The only applicable values of the ReasonCode are PeerKey_MISMATCH and STA_LEAVING. The encoding to Reason Code field values is defined in Table 10-3.

10.7.4.4 AP-initiated DLS teardown procedure

The AP-initiated DLS teardown procedure may be used in cases where the STA is unable to initiate the DLS teardown. The AP should initiate a DLS teardown procedure when the AP detects that either end of a DLS link has left the BSS without teardown of the DLS link, e.g., through receipt of a deauthentication frame or receipt of a deassociation frame.

If there are one or more STAs with open DLS connections with the STA being removed, the AP shall send a DLS Teardown frame to each such STA with ReasonCode AP_INITIATED.

NOTE—The AP can also initiate DLS teardown for implementation-dependent reasons.

10.7.5 Error recovery upon a peer failure

Every STA shall maintain an inactivity timer for every negotiated direct link (i.e., STAs on both sides of the link maintain these timers). The DLS inactivity timer shall be reset for every successful frame transmission or reception for that direct link. The direct link becomes inactive when no frames have been exchanged as part of the direct link for the duration of DLS timeout value, if the DLS Timeout Value field is a nonzero value during the DLS. When the direct link becomes inactive due to the timeout, the MLME issues an MLME-DLSTeardown.indication primitive to the SME and sends a DLS Teardown frame to the AP, with the peer MAC address as the destination MAC address and the reason code set to TIMEOUT. All frames shall henceforth be sent via the AP. The procedure is illustrated in Figure 14-9.

If there has been a TS setup for the data transfer, it is the responsibility of the STAs to renegotiate the parameters of the TSPEC with the HC.

When two STAs have set up a direct link, either STA may send DLS Request frames in order to update the DLS timeout value. If the updated DLS Request frame is accepted, both STAs initialize the timer to detect DLS timeout. Even if the updated DLS Request frame is not accepted, the original direct link remains active.

10.7.6 Secure DLS operation

PeerKey Handshake, defined in 11.6.8, is used to establish the keys needed to enable secure DLS. The PeerKey message exchange shall start after the DLS establishment and completed prior to initiation of the DLS data frame exchange.

The STKSA remains valid even if the STA disassociates from the originating AP, but the STKSA shall be deleted before a STA attempts another association or reassociation. If an AP transmits a Deauthenticate or Disassociate message to a STA, the AP shall also initiate teardowns for any existing DLS. The DLS STKs shall be deleted when the DLS teardown messages is sent or received.

10.8 TPC procedures

10.8.1 General

Regulations that apply to the 5GHz band in most regulatory domains require RLANs operating in the 5 GHz band to use transmitter power control, involving specification of a regulatory maximum transmit power and a mitigation requirement for each allowed channel, to reduce interference with satellite services. This standard describes such a mechanism, referred to as transmit power control (TPC).

This subclause describes TPC procedures that may satisfy needs in many regulatory domains and other frequency bands and may be useful for other purposes (e.g., reduction of interference, range control, reduction of power consumption).

STAs shall use the TPC procedures defined in Clause 10.8 if dot11SpectrumManagementRequired is true. dot11SpectrumManagementRequired shall be set to true when regulatory authorities require TPC. It may also be set to true in other circumstances. The TPC procedures provide for the following:

- Association of STAs with an AP in a BSS based on the STAs' power capability (see 10.8.2).
- Peering of mesh STAs based on the mesh STAs' power capability (see 10.8.3).
- Specification of regulatory and local maximum transmit power levels for the current channel (see 10.8.4).
- Selection of a transmit power for each transmission in a channel within constraints imposed by regulatory and local requirements (see 10.8.5).
- Adaptation of transmit power based on a range of information, including path loss and link margin estimates (see 10.8.6).

If dot11SpectrumManagementRequired is true, a STA shall not join an infrastructure BSS, MBSS or IBSS unless the Spectrum Management bit is 1 in the Capability Information field in Beacon frames and Probe Response frames or in the Condensed Capability Information field in Measurement Pilot frames received from other STAs in the BSS, with the following exceptions:

- A STA may operate when the Spectrum Management bit is 0 if the STA determines that it is in a regulatory domain that does not require TPC or determines that it meets regulatory requirements even if TPC is not employed. Potential methods for determining the regulatory domain include receiving a country indication in the Beacon frame, Measurement Pilot frame, user confirmation, or configuration information within the device. Potential methods to meet regulations even if TPC is not employed include using a transmit power that is below the legal maximum (including any mitigation factor).
- A STA shall set dot11SpectrumManagementRequired to true before associating with a BSS in which the Spectrum Management bit is 1 in the Capability Information field in Beacon frames and Probe Response frames or in the Condensed Capability Information field in Measurement Pilot frames received from the BSS.
- APs may allow association of devices that do not have the Spectrum Management bit equal to 1 in the Capability Information field in Association Request frames and Reassociation Request frames received from the STA to account for the existence of legacy devices that do not support TPC, but do meet regulatory requirements.

A mesh STA shall set dot11SpectrumManagementRequired to true before becoming a member of an MBSS in which the Spectrum Management bit is equal to 1 in the Capability Information field in Beacon frames and Probe Response frames received from the MBSS.

10.8.2 Association based on transmit power capability

A STA shall provide an AP with its minimum and maximum transmit power capability for the current channel when associating or reassociating, using a Power Capability element in Association Request frames or Reassociation Request frames.

An AP may use the minimum and maximum transmit power capability of associated STAs as an input into the algorithm used to determine the local transmit power constraint for any BSS it maintains. The specification of the algorithm is beyond the scope of this standard.

An AP may reject an association or reassociation request from a STA if it considers the STA’s minimum or maximum transmit power capability to be unacceptable. For example, a STA’s power capability might be unacceptable if it violates local regulatory constraints or increases the probability of hidden STAs by a significant degree. The criteria for accepting or rejecting an association or reassociation on the basis of transmit power capability are beyond the scope of this standard.

10.8.3 Peering based on transmit power capability

A mesh STA shall provide a candidate peer mesh STA with its minimum and maximum transmit power capability for the current channel when becoming a member of an MBSS, using a Power Capability element in Mesh Peering Open frames.

A mesh STA may use the minimum and maximum transmit power capability of a neighbor peer mesh STA as an input into the algorithm used to determine the local transmit power constraint. The specification of the algorithm is beyond the scope of this standard.

A mesh STA may reject a Mesh Peering Open request from a candidate mesh STA if it considers the candidate mesh STA’s minimum or maximum transmit power capability to be unacceptable. For example, a candidate mesh STA’s power capability might be unacceptable if it violates local regulatory constraints. The criteria for establishing or rejecting a Mesh Peering Open request on the basis of transmit power capability are beyond the scope of this standard.

10.8.4 Specification of regulatory and local maximum transmit power levels

A STA shall determine a regulatory maximum transmit power for the current channel. The STA shall use the minimum of the following:

— Any regulatory maximum transmit power received in a Country element from the AP in its BSS, another STA in its IBSS, or a neighbor peer mesh STA in its MBSS and
— Any regulatory maximum transmit power for the channel in the current regulatory domain known by the STA from other sources.

A STA shall determine a local maximum transmit power for the current channel by selecting the minimum of the following:

— Any local maximum transmit power received in the combination of a Country element and a Power Constraint element from the AP in its BSS, another STA in its IBSS, or a neighbor peer mesh STA in its MBSS and
— Any local maximum transmit power for the channel regulatory domain known by the STA from other sources.

The Local Power Constraint field of any transmitted Power Constraint element shall be set to a value that allows the mitigation requirements to be satisfied in the current channel.

Any calculation of the local maximum transmit power for the channel shall cause the mitigation requirements for the channel in the current regulatory domain to be satisfied. The conservative approach is to set the local maximum transmit power level equal to the regulatory maximum transmit power level minus the mitigation requirement. However, it may be possible to satisfy the mitigation requirement using a higher local maximum transmit power level. A lower local maximum transmit power level may be used for other purposes (e.g., range control, reduction of interference).

The regulatory and local maximum transmit powers may change in a STA during the life of an infrastructure BSS and an MBSS. However, network stability should be considered when deciding how often or by how much these maximums are changed. The regulatory and local maximum transmit powers shall not change during the life of an IBSS.

An AP in a BSS, a STA in an IBSS, and a mesh STA in an MBSS shall advertise the regulatory maximum transmit power for that STA's operating channel in Beacon frames and Probe Response frames using a Country element. An AP in a BSS, a STA in an IBSS, and a mesh STA in an MBSS shall advertise the local maximum transmit power for that STA's operating channel in Beacon frames and Probe Response frames using the combination of a Country element and a Power Constraint element.

Where TPC is being used for radio measurement without spectrum management, the inclusion of a Power Constraint element in Beacon and Probe Response frames shall be optional.

10.8.5 Selection of a transmit power

A STA may select any transmit power for transmissions in a channel within the following constraints:

- A STA shall determine a regulatory maximum transmit power and a local maximum transmit power for a channel in the current regulatory domain before transmitting in the channel.
- An AP shall use a transmit power less than or equal to the regulatory maximum transmit power level for the channel. The AP shall also meet any regulatory mitigation requirement.
- A STA that is not an AP shall use a transmit power less than or equal to the local maximum transmit power level for the channel.

10.8.6 Adaptation of the transmit power

A STA may use any criteria, and in particular any path loss and link margin estimates, to dynamically adapt the transmit power for transmissions of an MPDU to another STA. The adaptation methods or criteria are beyond the scope of this standard.

A STA may use a TPC Request frame to request another STA to respond with a TPC Report frame containing link margin and transmit power information. A STA receiving a TPC Request frame shall respond with a TPC Report frame containing the power used to transmit the response in the Transmit Power field and the estimated link margin in a Link Margin field.

An AP in a BSS or a STA in an IBSS shall autonomously include a TPC Report element with the Link Margin field set to 0 and containing transmit power information in the Transmit Power field in any Beacon frame or Probe Response frame it transmits.

10.9 DFS procedures

10.9.1 General

Regulations that apply to the 5GHz band in most regulatory domains require RLANs operating in the 5 GHz band to implement a mechanism to avoid co-channel operation with radar systems and to ensure uniform utilization of available channels. This standard describes such a mechanism, referred to as dynamic frequency selection (DFS).

This subclause describes DFS procedures that can be used to satisfy these and similar future regulatory requirements. The procedures might also satisfy comparable needs in other frequency bands and may be useful for other purposes.

STAs shall use the DFS procedures defined in 10.9.1 to 10.9.9 if dot11SpectrumManagementRequired is true. The MIB variable dot11SpectrumManagementRequired shall be set to true when regulatory authorities require DFS. It may also be set to true in other circumstances. The DFS procedures provide for the following:

- Associating STAs with an AP in a BSS based on the STAs' supported channels (see 10.9.2).
- Quieting the current channel so it can be tested for the presence of radar with less interference from other STAs (see 10.9.3).
- Testing channels for radar before using a channel and while operating in a channel (see 10.9.4).
- Discontinuing operations after detecting radar in the current channel to avoid further interfering with the radar (see 10.9.5).
- Detecting radar in the current and other channels based on regulatory requirements (see 10.9.6).
- Requesting and reporting measurements in the current and other channels (see 10.9.7).
- Selecting and advertising a new channel to assist the migration of a BSS after radar is detected (see 10.9.8).

For the purposes of DFS, the following statements apply:

- A STA with dot11SpectrumManagementRequired true shall not operate in a BSS unless the Spectrum Management bit is 1 in the Capability Information field in Beacon frames and Probe Response frames received from other STAs in the BSS, *with the following exception*.
  - A STA may operate when the Spectrum Management bit is 0 if the STA determines that it is in a regulatory domain that does not require DFS or determines that it meets regulatory requirements even if DFS is not employed. Potential methods for determining the regulatory domain include receiving a country indication in the Beacon frame, user confirmation, or configuration information within the device. Potential methods to enable regulations to be met even if DFS is not employed include independently detecting radar and ceasing operation on channels on which radar is detected.
- A STA shall set dot11SpectrumManagementRequired to true before associating with an infrastructure BSS, IBSS, or MBSS in which the Spectrum Management bit is 1 in the Capability Information field in Beacon frames and Probe Response frames received from the infrastructure BSS, IBSS, or MBSS.
- APs may allow association of devices that do not have the Spectrum Management bit equal to 1 in the Capability Information field in Association Request frames and Reassociation Request frames received from a STA to account for the existence of legacy devices that do not support DFS, but do meet regulatory requirements.

10.9.2 Association based on supported channels

A STA shall provide an AP with a list of the channels in which it can operate when associating or reassociating using a Supported Channels element in Association Request frames or Reassociation Request frames.

An AP may use the supported channels list for associated STAs as an input into an algorithm used to select a new channel for the BSS. The specification of the algorithm is beyond the scope of this standard.

An AP may reject an association or reassociation request from a STA if it considers the STA's supported channel list to be unacceptable. For example, a STA's supported channel list might be unacceptable if it can operate only in a limited number of channels. The criteria for accepting or rejecting associations or reassociations are beyond the scope of this standard.

10.9.3 Quieting channels for testing

An AP in a BSS or a mesh STA in an MBSS may schedule quiet intervals by transmitting one or more Quiet elements in Beacon frames and Probe Response frames. The AP or mesh STA may stop scheduling quiet intervals or change the value of the Quiet Period field, the Quiet Duration field, and the Quiet Offset field in Quiet elements as required. Only the most recently received Beacon frame or Probe Response frame defines all future quiet intervals; therefore, all schedules for quiet intervals based on older Beacon frames or Probe Response frames shall be discarded.

A STA in an IBSS may schedule quiet intervals only if it is the DFS owner. It shall set a quiet interval schedule by transmitting one or more Quiet elements in the first Beacon frame establishing the IBSS. All STAs in an IBSS shall continue these quiet interval schedules by including appropriate Quiet elements in any transmitted Beacon frames or Probe Response frames.

Multiple independent quiet intervals may be scheduled, so that not all quiet intervals have the same timing relationship to TBTT, by including multiple Quiet elements in Beacon frames or Probe Response frames.

Control of the channel is lost at the start of a quiet interval, and the NAV is set by all the STAs in the BSS for the length of the quiet interval. Transmission by any STA in the BSS of any MPDU and any associated acknowledgment within either the primary channel or the secondary channel (if present) of the BSS shall be complete before the start of the quiet interval. If, before starting transmission of an MPDU, there is not enough time remaining to allow the transmission to complete before the quiet interval starts, the STA shall defer the transmission by selecting a random backoff time, using the present CW (without advancing to the next value in the series). The short retry counter and long retry counter for the MSDU or A-MSDU are not affected.

10.9.4 Testing channels for radars

A STA does not transmit in a channel unless the channel has been tested for the presence of radar transmissions according to regulatory requirements.

10.9.5 Discontinuing operations after detecting radars

If a STA is operating in a channel and detects radar operating in the channel or accepts that another STA has detected radar operating in the channel, then the STA discontinues transmissions according to regulatory requirements.

10.9.6 Detecting radars

The methods that satisfy regulatory requirements to detect radar transmissions are beyond the scope of this standard.

10.9.7 Requesting and reporting of measurements

The response to a basic request is a basic report. It is mandatory for a STA in an infrastructure BSS to generate a basic report in response to a basic request if the request is received from the AP with which it is associated, except as specified in this subclause.

A STA may measure one or more channels itself or a STA may request other STAs in the same BSS to measure one or more channels on its behalf. These measurements may occur either during a quiet interval or during normal operation.

In order to request other STAs to measure one or more channels, a STA shall use a Measurement Request frame containing one or more Measurement Request elements. The measurement request may be sent to an individual or group destination address. Addressing requests to multiple STAs should be used with care to avoid a reply storm.

The measurement requests effectively allowed by these rules are shown in Table 10-4.

Table 10-4—Allowed measurement requests

Service set	Source of request	Destination of request	Type of measurement request allowed
BSS	AP	STA	Individual or group
	STA	AP	Individual only
	STA	STA	None
IBSS, MBSS	STA	STA	Individual or group

A STA that successfully requests another STA to perform a measurement on another channel should not transmit MSDUs, A-MSDUs, or MMPDUs to that STA during the interval defined for the measurement plus any required channel switch intervals. In determining this period, a STA shall assume that any required channel switches take dot11ChannelSwitchTime per switch.

A STA that receives a Measurement Request frame from a STA in its BSS shall parse the frame's Measurement Request elements in order, with measurements starting at the times specified by the Measurement Request elements. A STA may ignore any group addressed Measurement Request frames.

Any result of a measurement request shall be returned without undue delay to the requesting STA in Measurement Report elements using one or more Measurement Report frames. The result may be the completed measurement or an indication that the STA is unable to complete the measurement request.

A STA shall report it is too late to undertake a measurement request if it receives the request after the specified starting time for the measurement.

A STA shall report it is refusing a measurement request if all of the following conditions exist:

- The STA is capable of undertaking a measurement request,
- The STA does not want to undertake the measurement request at this time, and
- The measurement request is not mandatory

NOTE—Measurements are specified as mandatory or optional in 8.4.2.23).

A STA shall report it is incapable of performing a measurement request if any of the following conditions exists:

- The STA is incapable of undertaking an optional measurement request, or
- The STA does not support the channel specified in a mandatory measurement request, or
- The STA does not support any requested parallel measurements in the same or different channels.

The Measurement Report frames shall contain the same Dialog Token field as the corresponding Measurement Request frame, and each Measurement Report element shall contain the same Measurement Token field as the corresponding Measurement Request element.

A STA may autonomously report measurements to another STA in its BSS using a Measurement Report frame with a Dialog Token field set to 0 with one or more Measurement Report elements. A STA in an IBSS may also autonomously report measurements to other STAs in the IBSS using the Channel Map field in the IBSS DFS element in a Beacon frame or Probe Response frame.

A STA may enable or disable measurement requests or autonomous measurement reports from another STA by transmitting Measurement Request elements with the Enable bit set to 1 and the Request bit and Report bit set to 0 or 1, as appropriate. These elements do not require a corresponding Measurement Report element in a Measurement Report frame. All measurement requests and reports are enabled by default. An AP may ignore a request to disable a mandatory measurement request. All others requests shall be honored.

10.9.8 Selecting and advertising a new channel

10.9.8.1 General

An attempt may be made to move a BSS to a new operating channel. It is an objective that disruption to the BSS is minimized in this process, although it should be recognized that a channel switch might not successfully move all STAs. It should also be stressed that the channel switch process is distinct from the regulatory requirement to cease transmission on a particular channel in the presence of radar.

10.9.8.2 Selecting and advertising a new channel in an infrastructure BSS

The decision to switch to a new operating channel in an infrastructure BSS shall be made only by the AP. An AP may make use of the information in Supported Channel elements and the results of measurements undertaken by the AP and other STAs in the BSS to assist the selection of the new channel. The algorithm to choose a new channel is beyond the scope of this standard, but shall satisfy applicable regulatory requirements, including uniform spreading rules and channel testing rules. The AP shall attempt to select a new channel that is supported by all associated STAs, although it should be noted that this might not always be possible.

An AP shall inform associated STAs that the AP is moving to a new channel and maintain the association by advertising the switch using Channel Switch Announcement elements in Beacon frames, Probe Response frames, and Channel Switch Announcement frames until the intended channel switch time. The AP may force STAs in the BSS to stop transmissions until the channel switch takes place by setting the Channel Switch Mode field in the Channel Switch Announcement element to 1. The channel switch should be scheduled so that all STAs in the BSS, including STAs in power save mode, have the opportunity to receive at least one Channel Switch Announcement element before the switch. The AP may send the Channel Switch Announcement frame in a BSS without performing a backoff, after determining the WM is idle for one PIFS period.

A STA that receives a Channel Switch Announcement element may choose not to perform the specified switch, but to take alternative action. For example, it may choose to move to a different BSS.

A STA in a BSS that is not the AP shall not transmit the Channel Switch Announcement element.

10.9.8.3 Selecting and advertising a new channel in an IBSS

DFS in an IBSS is complicated by the following:

- There is no central AP function for collating measurements or coordinating a channel switch. If STAs make independent decisions to switch channel in the presence of radar, there is a danger that all STAs announce switches to different channels if several of them detect the radar.
- There is no association protocol that can be used to
  - Exchange supported channel information and
  - Determine membership of the IBSS at a given instant for requesting measurements.
- Beaconing is a shared process; therefore, there is no guarantee that a STA that has something to send (e.g., a channel switch message) will be the next STA to transmit a Beacon frame.
- A 20/40 MHz IBSS cannot be changed to a 20 MHz IBSS, and a 20 MHz IBSS cannot be changed to a 20/40 MHz IBSS.

The DFS owner service, IBSS DFS element, and Channel Switch Announcement frame address these complications.

- The DFS owner service provides a central point of coordination for a channel switch. It attempts to minimize the probability that multiple STAs concurrently decide to switch to different channels. The DFS Owner field and DFS Recovery Interval field within the IBSS DFS element support the DFS owner service.
- Each STA shall include a Channel Map field within the IBSS DFS elements that it transmits. The channel map communicates the STA-supported channel set and basic measurement reports for that STA.
- The ability to send a Channel Switch Announcement element within a management frame other than a Beacon frame or Probe Response frame is essential.

The potential for hidden nodes within an IBSS means that the IBSS channel switch protocol is best effort. All members of an IBSS shall have an individual responsibility to cease transmission on a particular channel in the presence of radar.

A STA at which an IBSS is started shall be a DFS owner in that IBSS. That STA shall include its MAC address in the DFS Owner field of the IBSS DFS element and DFS Recovery Interval field from the MLME.START.request primitive parameter. The purpose of the DFS owner is to coordinate a channel switch when required. All STAs within a spectrum-managed IBSS shall have the ability to become DFS owner.

Each STA in an IBSS shall adopt the DFS owner and the DFS owner recovery interval from any valid IBSS DFS element when the frame contained a matching SSID and the value of the timestamp is later than the STA’s TSF timer. The STA shall include the adopted values within the IBSS DFS elements it transmits. Because all STAs in an IBSS participate in sending Beacon frames, this process always, over a number of beacon intervals, results in a unified view of one DFS owner throughout the IBSS.

In order to attempt a channel switch using the DFS owner, a STA that detects radar shall broadcast one or more Measurement Report frames indicating the presence of the radar.

A DFS owner receiving a Measurement Report frame indicating the presence of radar in the current channel shall select and advertise a new operating channel (including the possibility of no change). The DFS owner may make use of information received in Channel Map fields and from measurements undertaken by other members of the IBSS to assist the selection of the new channel. The algorithm to choose a new channel is beyond the scope of this standard, but shall satisfy any regulatory requirements, including uniform spreading rules and channel testing rules. The DFS owner shall attempt to select a new channel that is supported by all members of the IBSS. It should be noted that this process might be imperfect in that the DFS owner may have incomplete knowledge and there may be no suitable channel.

The DFS owner shall attempt to make the members of the IBSS aware of the new operating channel by broadcasting at least one Channel Switch Announcement frame. The DFS owner shall also include the Channel Switch Announcement element in all Beacon frames, Probe Response frames, or Channel Switch Announcement frames until the intended channel switch time. A STA that receives a valid Channel Switch Announcement element shall repeat this element in all Beacon frames and Probe Response frames that it transmits. The DFS owner may attempt to silence STAs in the IBSS until the channel switch takes place using the Channel Switch Mode field in the Channel Switch Announcement element. If possible, the channel switch should be scheduled so that all STAs in the IBSS, including STAs in power save mode, have the opportunity to receive at least one Channel Switch Announcement element before the switch.

If a STA does not receive a valid Channel Switch Announcement element from the DFS owner within DFS recovery time measured from the end of the frame within which radar notification was first transmitted by the STA or received from another STA, then it shall enter a DFS owner recovery mode. In DFS owner recovery mode, the STA shall assume the role of DFS owner, shall select a new operating channel, and shall advertise the new channel by transmitting a Channel Switch Announcement frame using the contention resolution algorithm defined for beacon transmission at TBTT in 10.1.3.3. The STA shall also include the Channel Switch Announcement element in all Beacon frames and Probe Response frames until the intended channel switch time. A STA that is not the DFS owner shall not initiate a channel switch.

If the STA receives a valid Channel Switch Announcement element from another member of the IBSS, the STA shall leave DFS owner recovery mode prior to the channel switch and adopt the received channel switch information. If the Channel Switch Announcement element was within a Beacon frame or Probe Response frame, the STA shall also adopt the DFS owner address from the IBSS DFS element. If the Channel Switch Announcement element was within a Channel Switch Announcement frame, the STA shall adopt the DFS owner from the transmitter address of the received frame.

There are several circumstances when DFS owner recovery is required (e.g., if the original DFS owner has left the network or if the original measurement report was not received by the initial DFS owner). It should be noted that DFS owner recovery might temporarily give rise to more than one DFS owner within the IBSS. This risk is mitigated by the random nature of the IBSS DFS recovery mechanism. However, because all STAs in an IBSS participate in sending Beacon frames, over a number of beacon periods, there will be convergence from multiple DFS owners to one DFS owner.

10.9.8.4 MBSS channel switching

10.9.8.4.1 General

The mesh channel switch may be triggered by the need to avoid interference to a detected radar signal, or to reassign mesh STA channels to ensure the MBSS connectivity.

A mesh STA may make use of the information in Supported Channel elements, Supported Operating Classes elements, and the results of measurements undertaken by the mesh STAs in the MBSS to assist the selection of the new channel. The algorithm to choose a new channel is beyond the scope of this standard, but shall satisfy applicable regulatory requirements, including uniform spreading rules and channel testing rules.

A 20/40 MHz MBSS may be changed to a 20 MHz MBSS and a 20 MHz MBSS may be changed to a 20/40 MHz MBSS.

When an MBSS switches from a 20 MHz MBSS to a 20/40 MHz MBSS or switches from a 20/40 MHz MBSS to a 20 MHz MBSS, a mesh STA may need to do path maintenance to find an optimized path.

In the following subclauses, Mesh Channel Switch Announcement refers to Mesh Channel Switch Parameters element together with Channel Switch Announcement element or Extended Channel Switch Announcement element.

10.9.8.4.2 Initiating MBSS channel switch

A mesh STA shall not initiate a new channel switch attempt if there is an ongoing channel switch attempt by this mesh STA.

A mesh shall inform each of the peer mesh STAs that the mesh STA is moving to a new channel while maintaining mesh peerings by advertising the switch using Channel Switch Announcement elements together with Mesh Channel Switch Parameters element in Beacon frames, Probe Response frames, and Channel Switch Announcement frames until the intended channel switch time. The channel switch should be scheduled so that all mesh STAs in the MBSS, including mesh STAs in power save mode, have the opportunity to receive at least one Channel Switch Announcement element before the switch.

The fields in the Channel Switch Announcement element shall be set as follows. The Channel Switch Count field shall be set to the time period until the mesh STA sending the Channel Switch Announcement element switches to the new channel so that the channel switch attempt is propagated throughout the MBSS before the mesh STA leaves the channel. The Channel Switch Mode field is reserved. The New Channel Number field shall be set to the number of the channel to which the mesh STA is moving.

The fields in the Mesh Channel Switch Parameters element shall be set as follows:

- The Precedence Value field shall be set to a random value selected from a uniform distribution in the range from 0 to 65535.
- The mesh STA may force mesh STAs in the MBSS to stop transmissions of frames except frames containing Channel Switch Announcement element until the channel switch takes place by setting the Transmit Restrict subfield of the Flags field to 1.
- The Reason subfield in the Flags field shall be set to 1 to indicate that the content of the Reason Code field as defined in Table 8-36 of 8.4.1.7 is valid. The Reason Code field shall be set to MESH-CHANNEL-SWITCH-REGULATORY-REQUIREMENTS when channel switch is initiated to meet regulatory requirement; otherwise, the Reason Code field shall be set to MESH-CHANNEL-SWITCH-UNSPECIFIED.
- The Initiator subfield of the Flag field shall be set to 1.
- The Time To Live field should be set to the maximum number of hops (e.g., dot11MeshHWMPnetDiameter) for which this Channel Switch attempt is intended.

10.9.8.4.3 Processing channel switch announcement

Upon receipt of a Channel Switch Announcement, a mesh STA shall not accept and shall not process the received Channel Switch Announcement element or Extended Channel Switch Announcement element if any of the following is true:

- The Mesh Channel Switch Parameters element is not present in the received frame containing Channel Switch Announcement element or Extended Channel Switch Announcement element.
- The Time To Live field in the received Mesh Channel Switch Parameters element is 0.
- A mesh Channel switch is already running and mesh STA has not yet moved into the new channel and/or operating class and the Current Precedence value is greater than or equal to the received Precedence Value.

A mesh STA that receives a Channel Switch Announcement element may choose not to perform the specified switch, but to take alternative action. For example, it may choose to move to a different MBSS.

When mesh STA accepts a channel switch, it shall adopt information received in Channel Switch Announcement element and Mesh Channel Switch Parameters element. The mesh STA shall schedule the Channel Switch as per this information. If the Time To Live field value in the received Mesh Channel Switch Parameters element is greater than one, the mesh STA shall transmit Channel Switch Announcement frame and shall include Channel Switch Announcement element together with Mesh Channel Switch Parameters element in the Beacon and Probe Response frames until the intended channel switch time. The fields in the Channel Switch Announcement shall be set to the values identical to those in the received Channel Switch Announcement frame. The fields in the Mesh Channel Switch Parameters element shall be set to the values identical to those in the received Mesh Channel Switch Parameters element, except for the Time To Live field, Initiator field and the Transmit Restrict subfield of the Flags field. The Time To Live field shall be set to the received Time To Live field minus 1. The Initiator field shall be set to 0. The Transmit Restrict field shall be set to 1 when the mesh STA requires neighboring mesh STAs not to transmit further frames not containing Channel Switch Announcement element on the current channel until the scheduled channel switch. The Transmit Restrict subfield shall be set to 0 otherwise.

It is possible that a channel switch is not successful in moving all the mesh STAs in MBSS to the new operating channel. Transitioning to a new channel does not tear down mesh peerings and existing mesh peerings may be maintained in the new operating channel.

After moving into a new operating channel, the mesh STA shall perform CCA until a frame sequence is detected by which it can correctly set its NAV, or until a period of time equal to the ProbeDelay has transpired.

10.9.8.4.4 Channel switch across an operating class

When dot11OperatingClassesImplemented is true and the mesh STA is capable of operating in multiple operating classes, the mesh STA shall include the Supported Operating Classes element within its Mesh Peering Open frames. The Supported Operating Classes element announces the operating classes that the mesh STA supports.

When dot11OperatingClassesImplemented is true, mesh STAs may switch from the operating channel to a channel in a different operating class.

10.9.8.5 HT-greenfield transmissions in operating classes with behavior limits set of 16

The requirements described in this subclause apply only when an HT STA is operating in an operating class for which the behavior limits set listed in Annex E includes the value 16; i.e., the operating class is subject to DFS with 50–100 µs radar pulses.

A non-HT OBSS scan operation is a passive or active scan of the primary channel and of the secondary channel if it is within a 20/40 MHz BSS that a STA currently uses or intends to use. During a non-HT OBSS scan operation, the channel scan duration is a minimum of dot11OBSSScanPassiveTotalPerChannel TU when scanning passively and a minimum of dot11OBSSScanActiveTotalPerChannel TU when scanning actively.

Before an HT STA starts a BSS with the OBSS Non-HT STAs Present field of the HT Operation element equal to 0, the HT STA shall perform a non-HT OBSS scan in order to search for any existing non-HT OBSSs.

When an HT STA detects there are one or more non-HT OBSSs and if the HT STA starts a BSS on that channel, then the HT STA shall set the OBSS Non-HT STAs Present field of the HT Operation element to 1; otherwise, the HT STA may set the OBSS Non-HT STAs Present field of the HT Operation element to 0.

NOTE—Detection of a non-HT OBSS can be achieved by the reception of a Beacon or Probe Response frame that does not contain an HT Capabilities element or HT Operation element.

An HT AP shall not transmit a PPDU with the FORMAT parameter of the TXVECTOR set to HT_GF if the OBSS Non-HT STAs Present field of the HT Operation element is equal to 1 in the most recently transmitted management frame that contained that element.

An HT non-AP STA shall not transmit a PPDU with the FORMAT parameter of the TXVECTOR set to HT_GF if the most recent frame received from its AP containing an HT Operation element has the OBSS Non-HT STAs Present field equal to 1.

NOTE—This requirement applies also to PPDUs transmitted on a direct link between two non-AP STAs.

When moving the BSS to a new channel or set of channels and before completing a non-HT OBSS scan of the new channel or set of channels, an HT AP shall set the OBSS Non-HT STAs Present field of the HT Operation element to 1. After the HT AP completes one non-HT OBSS scan of the new channel or set of channels and if the HT AP has detected that there are zero non-HT OBSSs, then the HT AP may set the OBSS Non-HT STAs Present field of the HT Operation element to 0.

10.9.9 Channel Switch Announcement element operation

A Channel Switch Mode equal to 1 means that the STA in a BSS to which the frame containing the element is addressed shall transmit no further frames within the BSS until the scheduled channel switch. A STA in an IBSS may treat a Channel Switch Mode field equal to 1 as advisory. A Channel Switch Mode equal to 0 does not impose any requirement on the receiving STA.

10.10 Extended channel switching (ECS)

10.10.1 General

This subclause describes ECS procedures that change BSS operation in channel frequency and channel bandwidth. Enabling STAs (see 10.12), APs, DFS owners, and mesh STAs are each STAs that may construct and transmit frames containing Extended Channel Switch Announcement elements when dot11ExtendedChannelSwitchActivated is true.

A STA shall use the ECS procedures defined in this subclause if dot11ExtendedChannelSwitchActivated is true. When dot11ExtendedChannelSwitchActivated is true, dot11MultiDomainCapabilityActivated and dot11OperatingClassesRequired shall be true, and the Extended Channel Switching field of the Extended Capabilities element shall be set to 1 to indicate that the STA can perform ECS procedures. When dot11ExtendedChannelSwitchActivated is false, the STA shall not use ECS procedures. When an AP is switching to a different channel and one or more of its associated STAs do not support Extended Channel Switch, then both the Extended Channel Switch Announcement and the Channel Switch Announcement elements and frames may be used.

If dot11ExtendedChannelSwitchActivated and dot11LCIDSERequired are true, frames containing Channel Switch Announcement elements shall not be transmitted.

An enabling STA may send frames containing Extended Channel Switch Announcement elements to dependent STAs (see 10.12.5). If dot11DSERequired is true, a STA shall perform ECS procedures to switch at the time indicated by the channel switch count, or the STA shall change its enablement state for the enabling STA to unenabled.

10.10.2 Advertising supported operating classes

When dot11ExtendedChannelSwitchActivated is true, the Current Operating Class field in the Supported Operating Classes element shall indicate the operating class in use for transmission and reception. The Operating Classes field shall list all operating classes with which the STA is capable of operating for the country that is specified in the Country element (8.4.2.10).

10.10.3 Selecting and advertising a new channel and/or operating class

10.10.3.1 General

An attempt may be made to move a BSS to a new operating channel and/or new operating class using extended channel switching. An objective during this process is to minimize disruption to the BSS. It is possible, however, that an extended channel switch is not successful in moving all STAs to the new channel and/or operating class.

10.10.3.2 Selecting and advertising a new channel in an infrastructure BSS

When an AP with dot11DSERequired true receives frames containing Extended Channel Switch Announcement elements from the enabling STA, it shall advertise an extended channel switch with the same channel switch mode, new operating class, new channel number, and channel switch count as received in the Extended Channel Switch Announcement elements.

The decision to switch to a new operating channel and/or operating class in an infrastructure BSS is made by the AP when dot11DSERequired is false. An AP may make use of the information in the Supported Channels element, Supported Operating Classes element, and the results of measurements undertaken by the AP and other STAs in the BSS to assist the selection of the new channel and/or operating class. A method to make the decision and to select a new channel is defined in 10.9.8.2.

When an AP is switching to a different operating class and dot11ExtendedChannelSwitchActivated is true, then the AP shall use the Extended Channel Switch Announcement element and frame. In addition, the AP may also send Channel Switch Announcement elements and frames when the requirements signified by the new operating class are met by all associated STAs.

When an AP is switching to a new channel within the same operating class and dot11ExtendedChannelSwitchActivated is true, then the AP shall send the Extended Channel Switch Announcement element and frame, or both the Extended Channel Switch Announcement and the Channel Switch Announcement elements and frames. If dot11ExtendedChannelSwitchActivated is false, the AP shall send the Channel Switch Announcement element and frame, or both the Extended Channel Switch Announcement and the Channel Switch Announcement elements and frames.

When dot11ExtendedChannelSwitchActivated is true, an AP shall inform associated STAs that the AP is moving to a new channel and/or operating class and maintain the association by advertising the switch using Extended Channel Switch Announcement elements in any transmitted Beacon frames, Probe Response frames, and Extended Channel Switch Announcement frames until the intended channel switch time. The AP may request STAs in the BSS to stop transmissions until the channel switch takes place by setting the Extended Channel Switch Mode field to 1 in the Extended Channel Switch Announcement element. If possible, the channel switch should be scheduled so that all STAs in the BSS, including STAs in power save mode, have the opportunity to receive at least one Extended Channel Switch Announcement element before the switch. The AP may send the Extended Channel Switch Announcement frame without performing a backoff, after determining the WM is idle for one PIFS period. When both the Extended Channel Switch Announcement and the Channel Switch Announcement elements are transmitted in Public Action frames, they shall be sent in separate frames.

When a STA with dot11DSERequired equal to false receives an Extended Channel Switch Announcement element, it may choose not to perform the specified switch, but to take alternative action. For example, it might choose to move to a different BSS.

A non-AP STA in an infrastructure BSS shall not transmit the Extended Channel Switch Announcement element.

10.10.3.3 Selecting and advertising a new channel in an IBSS

When a DFS owner with dot11DSERequired true receives frames containing Extended Channel Switch Announcement elements from the enabling STA, it shall advertise an extended channel switch with the same channel switch mode, new operating class, new channel number, and channel switch count as received in the Extended Channel Switch Announcement elements.

The DFS owner that advertises a channel switch shall follow the rules defined in 10.9.8.3 with the following extensions:

a) If a DFS owner is switching to a new channel or to the same channel in a different operating class and dot11ExtendedChannelSwitchActivated is true, then the DFS owner shall use the Extended Channel Switch Announcement element and frame. Alternatively, both the Extended Channel Switch Announcement and the Channel Switch Announcement elements and frames may be used when Channel Switch Announcement elements and frames are permitted for operation in the band signified by the new operating class.

b) If a DFS owner is switching to a new channel within the same operating class and dot11ExtendedChannelSwitchActivated is true, then the DFS owner shall send the Extended Channel Switch Announcement element and frame, or both the Extended Channel Switch Announcement and the Channel Switch Announcement elements and frames. If dot11ExtendedChannelSwitchActivated is false, the DFS owner shall send the Channel Switch Announcement element and frame, or both the Extended Channel Switch Announcement and the Channel Switch Announcement elements and frames.

c) If both the Extended Channel Switch Announcement and the Channel Switch Announcement elements are transmitted in Public Action frames, they shall be sent in separate frames.

10.10.3.4 Selecting and advertising a new channel in an MBSS

A mesh STA may make use of the information in the Supported Channels element, Supported Operating Classes element, and the results of measurements undertaken by this mesh STA and other mesh STAs in the MBSS to assist the selection of the new channel and/or operating class.

The mesh STA that advertises a channel switch shall follow the rules defined in 10.9.8.4 with the following extensions:

a) If a mesh STA is switching to a different operating class, then the mesh STA shall use the Extended Channel Switch Announcement element and frame. Alternatively, both the Extended Channel Switch Announcement and the Channel Switch Announcement elements and frames may be used when Channel Switch Announcement elements and frames are permitted for operation in the band signified by the new operating class.

b) If a mesh STA is switching to a new channel within the same operating class, then the mesh STA shall send the Channel Switch Announcement element and frame, or both the Extended Channel Switch Announcement and the Channel Switch Announcement elements and frames.

c) If both the Extended Channel Switch Announcement and the Channel Switch Announcement elements are transmitted in Public Action frames, they shall be sent in separate frames.

d) The Extended Channel Switch Announcement element shall be included in the Beacon and Probe Response frames until the intended channel switch time.

10.11 Radio measurement procedures

10.11.1 General

This subclause describes the radio measurements and the procedures for requesting and reporting radio measurements between STAs. When a STA implements support for one or more of the procedures described in this subclause, it shall set dot11RadioMeasurementActivated to true. When dot11RadioMeasurementActivated is true, dot11MultiDomainCapabilityImplemented, dot11MultiDomainCapabilityActivated, dot11OperatingClassesImplemented, and dot11OperatingClassesRequired shall be true.

NOTE—A key issue in radio measurement is network operation and management, considering each STA's service load, power state, and operating conditions. Timely measurement reports might be more important than percentage of wireless capacity or STA capacity used by radio measurements. The measurement requester might consider traffic load and application requirements, regulatory requirements, and specific measurement states from every STA in support of wireless network management. There are no typical scenarios that describe IEEE 802.11 operation in all bands. Off-channel measurements are desirable to gather timely information about which channel to switch BSS operation to, and the noisier the operating environment, the more urgent the need for radio measurements off the serving channel. In any case, the measuring STA might refuse any measurement request.

10.11.2 Measurement on operating and nonoperating channels

If a STA supports measurements on nonoperating channels, it shall set dot11RMNonOperating-ChannelMeasurementActivated to true. Measurements on nonoperating channels may require the measuring STA to interrupt its data services on the operating channel, switch channels, and make measurements. Measurements on the operating channel may not require the STA to interrupt its data services.

All stations are responsible for maintaining data services and an association or membership with the BSS on the operating channel while performing measurements on nonoperating channels.

A STA shall determine the time between successive nonoperating channel measurements. This time may be a fixed length, or it may be determined by the STA using application-specific (or other) knowledge.

10.11.3 Measurement start time

A Radio Measurement Request frame may contain a single Measurement Request element or a sequence of Measurement Request elements. A STA that accepts the first or only measurement request within a Radio Measurement Request frame shall start the measurement as soon as practical after receiving the request. Subsequent measurement requests in the Radio Measurement Request frame that are accepted shall start as soon as practical after processing the previous request in the frame. Such measurement start times shall be subject to any specified Randomization Interval.

The Radio Measurement category permits a Randomization Interval to be specified for measurement start times. The intent of this is to avoid traffic storms that could arise with synchronized group addressed measurements. Prior to making each measurement in the requested sequence, the STA shall calculate a random delay distributed uniformly in the range 0 to the Randomization Interval specified in the measurement request. The STA shall not start the measurement until this delay has expired. Randomization Interval is specified in units of TUs. A Randomization Interval of 0 in a measurement request indicates that no random delay is to be used.

NOTE—It is important that designers recognize the need for statistical independence among the pseudo random number streams among STAs.

A number of repetitions may be specified in the Radio Measurement Request frame. In this case, the measurements in the frame are repeated as detailed further in 10.11.7. Each time a measurement is repeated, the STA shall recalculate the random delay as described above.

When a Measurement Start Time field is present in a measurement report, the measuring STA shall report the value of its TSF timer at the time the measurement started to an accuracy of ±1 TU.

10.11.4 Measurement Duration

The values of Request Measurement Duration and Duration Mandatory in the received measurement request and the dot11RMMaxMeasurementDuration setting in the receiving STA determine if the receiving STA accepts the measurement request and for how long the measurement is performed. dot11RMMaxMeasurementDuration indicates a measurement duration using the following formula:

$$\text{Maximum Measurement Duration in TUs} = 2^{(\text{dot11RMMaxMeasurementDuration} - 4)} \times \text{BeaconInterval}$$

Table 10-5 describes how a STA responds to a measurement request depending on the values of dot11RMMaxMeasurementDuration, Measurement Duration, and Duration Mandatory.

Table 10-5—Measurement Duration

dot11RMMax-MeasurementDuration	Measurement Duration in the Measurement Request	Duration Manda-tory	Notes
0	Any value	1	The STA shall perform measurements for the requested measurement duration.
0	Any value	0	The STA may perform measurements for a duration shorter than the requested measurement duration
Nonzero	Any value	0	The STA shall perform measurements for a maximum duration that is equal to the minimum of the requested measurement duration and the dot11RMMaxMeasurementDuration.
Nonzero	Requested measurement duration > Maximum-MeasurementDuration in TUs	1	The STA shall reject the measurement request with the Measurement Report Mode set to “refused.”
Nonzero	Requested measurement duration ≤ Maximum-MeasurementDuration in TUs	1	The STA shall perform the measurement for the requested duration.

Measurement duration on nonoperating channels is defined by dot11RMNonOperatingChannelMaxMeasurementDuration. If this attribute is 0, the STA does not support RM measurements on nonoperating channels; on receipt of a measurement request frame requesting a measurement on nonoperating channels, the STA shall reject the measurement request by returning a Measurement Report in which the Incapable bit in the Measurement Report Mode field is 1. The interpretation rules defined in Table 10-5 also apply for all nonzero values of dot11RMNonOperatingChannelMaxMeasurementDuration for measurements on nonoperating channels.

NOTE—Measurement duration on nonoperating channels is subject to further limitations due to maximum off-operating channel time.

If the Duration Mandatory bit is 1 in the Measurement Request mode field of a measurement request, the requested STA, if it accepts the request, shall perform the measurement over the Measurement Duration specified in the request. If the STA is unable to commit to making the measurement over the requested duration, it shall refuse the request by sending a measurement report in which the refused bit in the Measurement Report Mode field is set to 1. The measurement duration in the measurement report is equal to the requested measurement duration.

If the Duration Mandatory bit is 0 in the Measurement Request mode field of a measurement request, the requested STA, if it accepts the request, shall attempt a measurement using the requested duration as a maximum measurement duration, and may report results with an actual measurement duration less than the requested duration. The duration over which the measurement was made will be included in the measurement duration field of the measurement report.

Each separate measurement within the Radio Measurement Request frame shall be performed over a continuous measurement duration time period. In Measurement Request frames, the requested Measurement Duration value shall not be set to 0 except for Beacon Request with Measurement Mode set to Beacon Table Mode, Statistics Request, and requests for triggered autonomous measurements.

10.11.5 Station responsibility for conducting measurements

A Radio Measurement-capable STA shall decode and interpret each Radio Measurement Request frame that it receives and shall assess the contents against its capabilities and the impact on its own performance. A measurement request may be refused by the receiving STA by sending a Radio Measurement Report frame in which the refused bit in the Measurement Report Mode field is set to 1. The reasons for refusing a measurement request are outside the scope of this standard but may include reduced quality of service, unacceptable power consumption, measurement scheduling conflicts, or other significant factors.

In assessing the performance impact of each measurement request element, a STA may use application-specific knowledge or other knowledge to limit the time it spends away from the operating channel. In doing so, the STA may either:

- Reject any Measurement Request element in which the Duration Mandatory bit is 1 and that has a mandatory measurement duration exceeding the maximum allowed off-operating channel time, or
- Measure for a reduced duration if the Duration Mandatory bit is 0.

A STA shall cancel all in-process radio measurements and shall delete all pending, unprocessed radio measurement requests upon receipt of a Disassociation message or upon association or reassociation with a BSSID different from its most recent association.

10.11.6 Requesting and reporting of measurements

A STA may perform radio measurements on one or more channels itself or a STA may request STAs in the same BSS to perform measurements on its behalf.

A STA advertises its radio measurement capability using the RM Enabled Capabilities element. If a STA advertises that it is capable of a measurement, it shall not reject a request for the corresponding measurement by sending a Radio Measurement Report frame in which the Incapable bit in the Measurement Report Mode field is set to 1, except as specified in Clause 10.11.9.7. Measurement requests for radio measurements that the STA has advertised it is not capable of shall be rejected, and the corresponding report shall have the Incapable bit in the Measurement Report Mode field set to 1.

When requesting other STAs to measure one or more channels, a STA shall use a Radio Measurement Request frame containing one or more Measurement Request elements. The measurement request may be sent to an individual or group destination address. The permitted measurement requests are shown in Table 10-6.

Table 10-6—Allowed measurement requests

Service set	Source of request	Destination of request	Receiver address of radio measurement request frame
Infrastructure BSS	AP	Non-AP STA	Individual or group
	Non-AP STA	AP	Individual only
	Non-AP STA	Non-AP STA	Individual only for Direct Link within a BSS served by QoS AP, otherwise not allowed
IBSS	Non-AP STA	Non-AP STA	Individual or group

The source and destination of a measurement request shall both be a member of the same infrastructure BSS or a member of the same IBSS. Measurement requests with an individual Receiver Address shall be sent only to STAs that have indicated Radio Measurement capability.

The set of requested measurements received in the most recently received Radio Measurement Request frame of highest precedence is active at a STA. The precedence order for measurement requests shall be as follows (highest precedence first):

— Measurement requests received in individually addressed Radio Measurement Request frames

— Measurement requests received in Multicast-group addressed Radio Measurement Request frames

— Measurement requests received in Broadcast addressed Radio Measurement Request frames

The Measurement Request elements shall be processed in sequence by default, with certain Measurement Request elements processed in parallel according to the parallel bit field setting: see 8.4.2.23. A STA shall accept a Measurement Request with the parallel bit field enabled if dot11RMParallelMeasurenmentActivated is true; otherwise, the STA shall reject the Measurement Request by returning a Measurement Report with the Incapable bit in the Measurement Report Mode field set to 1.

If dot11RMParallelMeasurementActivated is true and if measurement resources are available, the STA processes each element by setting up and making the specified measurement. If measurement resources are not available to perform the requested parallel measurements, the STA shall return a Measurement Report with the Refused bit in the Measurement Report Mode field set to 1.

The Measurement Request elements within a Radio Measurement Request frame may specify multiple measurement types across multiple channels.

A STA may receive another Radio Measurement Request frame while the measurements requested in a previous Radio Measurement Request frame are pending or in progress. If this request is accepted, the set of measurement requests in the new frame supersedes any previous requests received in a Radio Measurement Request frame of the same or lower precedence. The measuring STA shall report the results of any completed measurements and terminate any pending or in-progress measurements. Results from a terminated in-progress measurement may be valid and reported if Duration Mandatory was not equal to 1 in the corresponding request. It is permissible for the superseding Radio Measurement Request frame to contain no new measurement requests. This has the effect of cancelling all pending or in-progress measurements of the same or lower priority. If a station receives a Radio Measurement Request frame with lower precedence than the currently active Radio Measurement Request frame, the station shall discard the measurement requests in the new Radio Measurement Request frame. Measurement Request elements that have the Enable bit equal to 1 shall be processed in all received Radio Measurement Request frames regardless of these precedence rules.

If a STA receives a Spectrum Management Measurement Request with Measurement Type equal to 0 (Basic Request), this shall take priority over any pending or in-progress radio measurements.

A STA that issues a radio measurement request to another STA to perform a measurement on the operating channel may continue to transmit MPDUs and MMPDUs to that STA while the measurement is being processed.

A STA that issues a radio measurement request to another STA to perform a measurement on a nonoperating channel is not required to take any special action to suspend traffic to that STA. All stations shall maintain state information such that data services and association or membership with the BSS continue when returning from a nonoperating channel measurement.

A single Measurement Request element may generate a large quantity of measurement report data. The measurement report data may be reported using multiple measurement report elements in multiple measurement report frames. The result of each measurement requested in a Measurement Request element shall be reported in one or more Measurement Report elements of type corresponding to the request. Each Measurement Report element returned shall have the same Measurement Token as in the corresponding Measurement Request element, and the same Actual Measurement Start Time field, if present, as in the first returned Measurement Report element. The results of each measurement should be returned without undue delay to the requesting STA.

Measurement Report elements shall be returned to the requesting STA in one or more Radio Measurement Report frames. Each Radio Measurement Report frame shall contain the same Dialog Token field value as the corresponding Radio Measurement Request frame, and the same Actual Measurement Start Time field, if present, as in the first returned Measurement Report element.

When a STA is permanently unable to make a requested measurement, the STA shall respond to such a measurement request received within an individually addressed Radio Measurement Request frame with a measurement report indicating that it is incapable of completing the measurement request. A STA shall not respond to requests received in group addressed frames in this manner. Examples of when a response of incapable is appropriate are:

- The requested measurement type is not supported.
- The measuring STA cannot support requested parallel measurements due to the requests relating to different channels.

A STA that receives a response with an incapable indication shall not make the same request to the responding STA during the lifetime of the current association, or IBSS membership. This is logically the same as the responding STA using the Enable and Request bits in a measurement request to indicate that it does not accept measurement requests of a certain type. A STA that has indicated an incapable response to a requesting STA may discard further requests of the same type from that STA without responding.

A STA may refuse to make any requested measurement. A STA refusing a measurement request within an individually addressed Radio Measurement Request frame shall respond with a measurement report indicating that it is refusing the measurement request. A STA shall not respond to measurement requests received in Radio Measurement Request frames in this manner.

By default, a STA may send a radio measurement request of any defined measurement type. A STA that receives a Measurement Request element with the Enable bit equal to 1 and the Request bit equal to 0 shall not issue measurement requests of the Measurement Type type in the request to the STA from which the element was received.

Since measurements on nonoperating channels interrupt normal operation on the operating channel, the requesting STA should consider each STA's service load, power state, and operating conditions. Since measurements on the operating channel execute concurrently with normal traffic processing, operating channel measurements can be requested more frequently and for longer durations.

10.11.7 Repeated measurement request frames

Radio Measurement Request frames contain a field specifying the number of repetitions for the Radio Measurement Request frame.

If the Radio Measurement Request frame includes a nonzero value for the Number of Repetitions and dot11RMRepeatedMeasurementsActivated is false, the STA shall reject the measurement request and return a Measurement Report with the Incapable bit in the Measurement Report Mode field set to 1.

If the Radio Measurement Request frame includes a nonzero value for the Number of Repetitions and dot11RMRepeatedMeasurementsActivated is true, the STA shall iterate (repeat) the processing of all the Measurement Request elements in the frame as specified by the value in the Number of Repetitions field. A value of 0 in the Number of Repetitions field indicates Measurement Request elements are executed once without repetition; a value of 1 in the Number of Repetitions field indicates Measurement Request elements are executed twice, one initial execution and one repetition; and so on. When completing the initial processing of the last Measurement Request element in the frame, the STA shall begin processing of the first Measurement Request element in the frame to repeat the frame until the number of iterations reaches the value in the Number of Repetitions field. Measurement Request elements with the Enable bit equal to 1 shall be processed once regardless of the value in the Number of Repetitions in the measurement request.

Each repeated measurement result shall include the Measurement Token value as in the corresponding Measurement request element and the Dialog Token value as in the corresponding Radio Measurement Request frame.

Measurement results shall be reported for each repetition of a repeated measurement request subject to any conditional reporting requirement.

STAs responding with incapable or refused indications to measurement requests within a Radio Measurement Request frame with a nonzero value for Number of Repetitions shall respond only once.

10.11.8 Triggered autonomous reporting

Autonomous reporting is defined for spectrum management measurements supporting DFS; see 10.9.7. It allows a STA to report the results of measurements to a peer STA for which there was no explicit measurement request. In this case, the transmission of autonomous reports shall be entirely the decision of the STA at which such reporting has been enabled. An example of this use would be to report a change in conditions at the STA observed as a result of background measurement, e.g., the presence of a radar signal.

In radio measurement, triggered autonomous reporting shall be subject to trigger conditions set by the enabling STA that determine when measurement reports are issued. Triggered autonomous reporting provides a method for conditional reporting during continuous background measurements. An example of the use of triggered autonomous measurement is for reporting problem conditions in continuous, noninvasive statistical monitoring.

Triggered autonomous reporting is defined for the Transmit Stream/Category Measurement measurement type; see 10.11.9.8. When dot11MgmtOptionMulticastDiagnosticsActivated is true, triggered autonomous reporting is used for Multicast Diagnostics (10.11.19). When dot11MgmtOptionTriggerSTAStatisticsActivated is true, triggered autonomous reporting is used for STA Statistics Reports (10.11.9.5).

If dot11RMTriggeredTransmitStreamCategoryMeasurementActivated is true, a STA indicates that it wishes to accept triggered autonomous reports by sending a Measurement Request element with the Enable and Report bits set to 1; see 8.4.2.23. The type of measurement is indicated in the Measurement Type field. Trigger conditions that determine when measurement reports are to be generated shall be specified in the Measurement Request field. A Measurement Request element that is being used to control triggered autonomous reporting shall be sent within a Radio Measurement Request frame. Measurement Request elements being used to request measurements may also appear in the same Radio Measurement Request frame. The Radio Measurement Request frame may be sent to a group receiver address to enable triggered autonomous reports at more than one STA.

A STA shall not send autonomous reports for radio measurement types having triggered autonomous reporting enabled without a requested trigger condition having been met.

If a request to enable triggered autonomous reporting is sent to an individual address and the recipient STA does not support measurements of the type indicated or the recipient STA has dot11RMTriggeredTransmitStreamCategoryMeasurementActivated equal to false, a Measurement Report element shall be returned to the requesting STA with the Incapable bit set to 1. A STA may also refuse to enable triggered autonomous reporting. In this case a Measurement Report element shall be returned to the requesting STA with the refused bit set to 1. Such responses shall not be issued if the request to enable triggered autonomous reporting was sent to a group address.

A STA receiving a request to enable triggered autonomous reporting from another STA may send reports of the appropriate type, addressed to the individual address of the STA that sent the enable request. Autonomous reports shall be sent only to the individual addresses of STAs from which a valid enable request has been received and shall be issued only when a requested trigger condition has been met. The Measurement Token in each Measurement Report element and the Dialog Token value in the Measurement Report frame shall both be set to 0 in a triggered autonomous report.

A STA may update the trigger conditions set for triggered autonomous reports by issuing a new Measurement Request element with the Enable and Report bits both set to 1, the Measurement Type field set to the appropriate type and the Measurement Request field indicating the new trigger conditions. A STA disables all triggered autonomous measurement reports by sending a Measurement Request element with the Enable bit set to 1 and the Report bit set to 0; see 8.4.2.23.

A STA in an infrastructure BSS shall cease all triggered autonomous reporting if it disassociates, or reassociates to a different BSS (reassociation to the same BSS shall not affect triggered reporting). A STA in an independent BSS shall cease all triggered autonomous reporting if it leaves the BSS.

Triggered autonomous reporting and requested measurements are independent: a STA may request measurements from another STA even if it has enabled triggered autonomous reporting from that STA. All Measurement Request elements received in Radio Measurement Request frames that have the Enable bit equal to 1 shall be processed without regard for the measurement precedence rules for requested measurements in 10.11.6.

A number of triggered measurements may run concurrently at a non-AP STA. The number of simultaneous triggered measurements supported is outside the scope of the standard. Each triggered measurement is logically separate; reporting conditions such as Trigger Timeout periods shall only apply to the measurement for which they are established.

10.11.9 Specific measurement usage

10.11.9.1 Beacon Report

If a STA accepts a Beacon Request it shall respond with a Radio Measurement Report frame containing Beacon Measurement Reports for all observed BSSs matching the BSSID and SSID in the Beacon Measurement Request, at the level of detail requested in the Reporting Detail. If the Reporting Detail is 1 and the optional Request information subelement is included in the Beacon Measurement Request, the corresponding Beacon Measurement Report shall include the list of elements listed in the Request information subelement. The RCPI in the Beacon Report indicates the power level of the received Beacon, Measurement Pilot, or Probe Response frame. For repeated measurements (when the Measurement Request frame contains a nonzero value for the Number of Repetitions field), the transmission of the Beacon Report may be conditional on the measured RCPI or RSNI value. If the Measurement Request frame contains a 0 value for the Number of Repetitions field, the Beacon Reporting Information subelement shall not be included in the Beacon Request. If the Measurement Request frame contains a nonzero value for the Number of Repetitions field, and if both dot11RMBeaconMeasurementReportingConditionsActivated and dot11RMRepeatedMeasurementsActivated are true, and if a Beacon Reporting Information subelement is included in a Beacon Request, the STA shall respond with a Beacon Report if the indicated Beacon Reporting Condition is true. Otherwise, the STA shall not respond with a Beacon Report. Table 8-66 lists the reporting conditions that are based on the measured RCPI or RSNI levels.

If the requested Beacon Measurement Report includes the Supported Operating Classes element, then the channel number, operating class, and/or reported frame information for that measurement may be included in the beacon report; otherwise these fields shall be set to 255 in the beacon report. The STA shall use the Reporting Detail specified in the measurement request to determine the data to be included in the measurement report. If the STA has no beacon information available then the STA may either refuse the request or send an empty Beacon Report.

If dot11RMBeaconPassiveMeasurementActivated is true and the Measurement Mode in the measurement request is Passive, the measuring STA shall perform the following procedure (or an equivalent procedure) on the requested channel:

- Set a measurement duration timer.
- At the end of the measurement duration, process all received Beacons or Probe Response management frames with the requested SSID and BSSID to compile the measurement report. The STA shall use the Reporting Detail specified in the measurement request to determine the data to be included in the measurement report. If no Beacons or Probe Responses with the requested SSID and BSSID were received in the measurement duration, then process all Measurement Pilot Frames with the requested BSSID to compile the measurement report. Otherwise, compile an empty Beacon measurement report.

If dot11RMBeaconPassiveMeasurementActivated is false and the Measurement Mode in the measurement request is Passive, the measuring STA shall reject the measurement request and return a Beacon Measurement Report with the Incapable bit in the Measurement Report Mode field set to 1.

If dot11RMBeaconActiveMeasurementActivated is true and the Measurement Mode in the measurement request is Active, the measuring STA shall perform the following procedure (or an equivalent procedure) on the requested channel:

- If the channel is not the operating channel, wait for dot11RMMeasurementProbeDelay, or until a PHY-RXSTART.indication primitive has been received.
- Using the basic access protocol in 9.3.4.2, send a Probe Request management frame to the broadcast destination address (DA). The BSSID field in the Probe Request shall be set to the BSSID field in the measurement request. The SSID element in the Probe Request shall be set to the SSID element in the measurement request.

- Set a measurement duration timer.
- At the end of the measurement duration, process all received Probe Response and Beacon management frames with the requested SSID and BSSID to compile the measurement report. The STA shall use the Reporting Detail specified in the measurement request to determine the data to be included in the measurement report. If no Beacons or Probe Response frames were received in the measurement duration and Measurement Pilot frames with the requested BSSID were received in the measurement duration, then process all these Measurement Pilot Frames to compile the measurement report. Otherwise, compile an empty Beacon measurement report.

If dot11RMBeaconActiveMeasurementActivated is false and the Measurement Mode in the measurement request is Active, the measuring STA shall reject the measurement request and return a Beacon Measurement Report with the Incapable bit set in the Measurement Report Mode field.

When more than one Beacon or Probe Response from a BSS is received in the measurement duration, the contents of the Beacon Report shall be based on the latest received. If only Measurement Pilot frames were received in the measurement duration, the contents of the Beacon Report shall be based on the latest Measurement Pilot frame received.

If the BSSID field in the Measurement Request contains a wildcard BSSID, all observed BSSs with the requested SSID shall be reported in a separate Beacon Report for each BSSID. If the SSID subelement is not included in the Beacon Request, all observed BSSs shall be reported in a separate Beacon Report for each BSSID. In Active mode, Probe Response frames shall be evaluated regardless of whether the Probe Response frame was triggered by the measuring STA’s Probe Request.

On accepting an active or passive mode Beacon measurement request, a STA shall conduct measurements as follows:

- If the Channel Number is 0, a STA shall conduct iterative measurements on all supported channels in the specified Operating Class where the measurement is permitted on the channel and the channel is valid for the current regulatory domain.
- If the Channel Number is 255 and includes AP Channel Report subelements, a STA shall conduct iterative measurements on all supported channels listed in the AP Channel Report subelements that are valid for the current regulatory domain. If there is no AP Channel Report subelement included in the Beacon Report request, a STA shall conduct iterative measurements on all supported channels listed in the latest AP Channel Report received from the serving AP that are valid for the current regulatory domain. If there are no AP Channel Report subelements included in the Beacon Request, and no AP Channel Report included in last received AP Beacon frame, the STA shall reject the Beacon Report request.
- If the Channel Number is a value other than 0 or 255, a STA shall conduct iterative measurements on that Channel Number, where the measurement is permitted on the channel and the channel is valid for the current regulatory domain.

Measurements shall be made using the specified Measurement Duration with the time between each consecutive measurement as defined in 10.11.2. Iterative measurements shall cease when all channels have been measured. While the STA is processing a Beacon measurement request for iterative channel measurements, the STA shall not begin processing the next measurement request in the measurement request frame.

If dot11RMBeaconTableMeasurementActivated is true and the Measurement Mode in the measurement request is Beacon Table, the measuring STA shall return a Beacon Report containing the current contents of any stored beacon information for any supported channel with the requested SSID and BSSID without performing additional measurements. The receiving STA shall ignore the channel and measurement duration specified in the Beacon Request when Beacon Table mode is selected. The beacon information accumulated may be the result of any operation that caused the STA to acquire these results. If the stored beacon information is based on a measurement made by the reporting STA, and if the actual measurement start time, measurement duration, and Parent TSF are available for this measurement, then the beacon report shall include the actual measurement start time, measurement duration, and Parent TSF; otherwise the actual measurement start time, measurement duration, and Parent TSF shall be set to 0. The RCPI and RSNI for that stored beacon measurement may be included in the beacon report; otherwise the beacon report shall indicate that RCPI and RSNI measurements are not available.

If dot11RMBeaconTableMeasurementActivated is false and the Measurement Mode in the measurement request is Beacon Table, the measuring STA shall reject the measurement request and return a Beacon Measurement Report with the Incapable bit set in the Measurement Report Mode field.

For repeated measurements, the Beacon Request element may include a Beacon Reporting Information subelement that determines when the measuring STA is to send a Beacon Report for a measured Beacon, Measurement Pilot, or Probe Response frame with the requested SSID and BSSID. When the requested Reporting Condition value is nonzero, and dot11RMBeaconMeasurementReportingConditionsActivated is true, the STA shall create and transmit a Beacon Report for that measured frame if the condition indicated in Table 8-66 is true. Otherwise, the STA shall not transmit a Beacon Report for that measured frame. If multiple Beacons, Measurement Pilots, or Probe Response frames with the requested specific BSSID are received during the measurement duration, the reporting condition shall be applied only to the latest received Beacon, Measurement Pilot, or Probe Response. If multiple Beacons, Measurement Pilots, or Probe Response frames are received during the measurement duration when a wildcard BSSID is requested, the STA shall generate one Beacon Report for each BSSID occurring in frames that satisfy the reporting condition; the Beacon Report shall be based on the latest received Beacon, Measurement Pilot, or Probe Response for that specific BSSID. For reporting conditions 5–10, the serving AP's reference RCPI level and the serving AP's reference RSNI level referred to in Table 8-66 are average values of the RCPI or RSNI of the 16 most recent Beacon frames received from the measuring STA's serving AP. The serving AP's reference RCPI level and the serving AP's reference RSNI level are so averaged to provide a more accurate and stable indication of the signal level from the serving AP. For reporting conditions 5–10, the STA shall use the serving AP's reference RCPI level or reference RSNI level (with offset, if any) to test the measured RCPI or RSNI to determine whether to create and send a Beacon Report for this measured Beacon, Measurement Pilot, or Probe Response frame.

The STA shall return a Beacon Report with the Incapable bit set in the Measurement Report Mode field in the following cases:

- Reporting Condition in the Beacon Request is nonzero and dot11ReportingConditionsActivated is false,
- Reporting Condition in the Beacon Request is nonzero and the Number of Repetitions in the Measurement Request frame is 0.

NOTE—Reporting conditions described here for repeated Beacon Request measurements are distinct from the conditions defined elsewhere for triggered measurements.

10.11.9.2 Frame Report

If dot11RMFrameMeasurementActivated is true, and a station accepts a frame request, it shall respond with a Radio Measurement Report frame containing one or more Measurement (Frame) Report elements. (See 8.4.2.24.8.)

If the MAC Address field was included in the frame request, a Frame Report Entry where Transmitter Address (TA) matches the MAC address in the frame request shall be included in the Frame Report if at least one data or management frame was received with this Transmitter Address during the measurement duration. If the MAC address field was not included in the frame request in response to which this Frame Report is being generated, the measuring station shall report all frames correctly received during the measurement duration in one or more Frame Report elements.

If the Frame Request Type of the corresponding frame request equals 1, then each Frame Report element contains one Frame Count subelement that contains in turn one or more Frame Report Entries.The measuring station shall count the number of individually addressed data and management frames received from one transmit address during the measurement duration and shall summarize this traffic in a Frame Report Entry.

Each Frame Report Entry contains the Transmit Address, BSSID, PHY Type, Average RCPI, Last RSNI, Last RCPI, Antenna ID, and Frame Count for the frames counted in this Frame Report Entry.

The reported Average RCPI shall be an average of the RCPI values of frames received and counted in the Frame Report Entry. If there are up to 32 frames, then the Average RCPI indicates the mean of the RCPI of each of the frames. If there are more than 32 frames, then the Average RCPI indicates an exponentially weighted average, initialized by the mean RCPI of the first 32 frames and exponentially updated by new RCPI values. The averaging is calculated as depicted as follows.

If the number of frames correctly received is less than or equal to 32:

$$\text{Average RCPI} = \text{Sum of RCPI values} / \text{Number of frames}$$

For 33 correctly received frames and above:

$$\text{Average RCPI} = (\text{Last Average RCPI} \times 31 / 32) + (\text{Current frame RCPI} / 32)$$

The Last RCPI shall be the RCPI value of the most recently received frame counted in the Frame Report Entry. The Antenna ID field contains the identifying number for the antenna(s) used to receive the most recently received frame included in this Frame Report Entry as defined in 8.4.2.42. If different antennas are used to receive the frame preamble and the frame body, this Antenna ID shall contain the identifying number for the antenna(s) used to receive the frame body.

If dot11RMFrameMeasurementActivated is false, a station shall reject the received frame request and shall respond with a Frame Report in which the Incapable bit in the Measurement Report Mode field is 1.

10.11.9.3 Channel Load Report

If dot11RMChannelLoadMeasurementActivated is true and a station accepts a Channel Load Request, it shall respond with a Radio Measurement Report frame containing one Measurement (Channel Load) Report element. The Channel Load field is defined as the percentage of time, linearly scaled with 255 representing 100%, the STA sensed the medium was busy, as indicated by either the physical or virtual carrier sense (CS) mechanism. This percentage is computed using the following formula:

$$\text{Channel Load} = \text{Integer}((\text{channel busy time}/(\text{MeasurementDuration} \times 1024)) \times 255)$$

where channel busy time is defined to be the number of microseconds during which the CS mechanism, as defined in 9.3.2.1, has indicated a channel busy indication.

If dot11RMChannelLoadMeasurementActivated is false, a station shall reject the received Channel Load Request and shall respond with a Channel Load Report with the Incapable bit in the Measurement Report Mode field set to 1.

If dot11RMChannelLoadMeasurementActivated is true and if a Channel Load Reporting Information subelement is included in a Channel Load Request, the STA shall respond with a Channel Load Report if the indicated Channel Load Reporting Condition is true. Otherwise, the STA shall not respond with a Channel Load Report.

10.11.9.4 Noise Histogram Report

If dot11RMNoiseHistogramMeasurementActivated is true and a station accepts a Noise Histogram Request, it shall respond with a Radio Measurement Report frame containing one Measurement (Noise Histogram) Report element. The Noise Histogram Report shall contain the IPI densities observed in the channel for the IPI levels defined in Table 8-84.

To compute the IPI densities, the STA shall measure the IPI in the specified channel as a function of time over the measurement duration when NAV is equal to 0 (when virtual CS mechanism indicates idle channel) except during frame transmission or reception. The time resolution of the IPI measurements shall be in microseconds. The IPI densities are then computed for each of the nine possible IPI values using:

$$\text{IPI Density} = \text{Integer}\left[\frac{255 \times D_{IPI}}{(1024 \times D_M) - T_{NAV} - T_{TX} - T_{RX}}\right]$$

where

D_{IPI} is the duration receiving at the specified IPI value (μs)
D_M is the measurement duration (TU)
T_{NAV} is the total time that NAV is nonzero during the Measurement Duration (μs)
T_{TX} is the frame transmission time during the Measurement Duration (μs)
T_{RX} is the frame reception time during the Measurement Duration (μs)

The sum of the IPI densities is approximately 255. If either the NAV is nonzero, or if there is frame transmission, or if there is frame reception throughout the entire measurement duration period, no reportable IPI values are measured, and all IPI Densities shall be set to 0 in the Measurement Report element.

A STA shall include in the Noise Histogram Report an average noise power indicator (ANPI) value representing the average noise plus interference power on the measured channel at the antenna connector during the measurement duration. The STA may use Noise Histogram IPI density values to calculate ANPI. The IPI densities in the Noise Histogram Report may be used to calculate an average noise power for the channel during the measurement duration. This calculated average IPI power value may be reported as the value for ANPI. Any equivalent method to measure ANPI may also be used. ANPI power is defined in dBm using the same accuracy as defined for RCPI.

ANPI may be calculated over any period and for any received frame. ANPI may be calculated in any period and at any time by filtering all PHY IPI values in a MAC filter to exclude IPI values received when NAV is nonzero. These filtered IPI values represent idle channel noise and may be stored in a first-in-first-out (FIFO) buffer to facilitate ANPI calculation over a fixed number of IPI samples. ANPI may be so calculated upon receipt of any frame and may be used with RCPI to calculate RSNI for any received frame. Any equivalent method to measure ANPI may also be used to calculate RSNI for any received frame.

If dot11RMNoiseHistogramMeasurementActivated is false, a station shall reject the received Noise Histogram Measurement Request and shall respond with a Noise Histogram Measurement Report with the Incapable bit in the Measurement Report Mode field set to 1.

If dot11RMNoiseHistogramMeasurementActivated is true and if a Noise Histogram Reporting Information subelement is included in a Noise Histogram Request, the STA shall respond with a Noise Histogram Report if the indicated Noise Histogram Reporting Condition is true. Otherwise the STA shall not respond with a Noise Histogram Report.

10.11.9.5 STA Statistics Report

If dot11RMStatisticsMeasurementActivated is true and a station accepts a STA Statistics Request, it shall respond with a Radio Measurement Report frame including one STA Statistics Report element. If the Requested Measurement Duration value is 0, the STA shall report the current values for the requested Statistics Group Data. If the Requested Measurement Duration value is greater than 0, the STA Statistics Report reports the change in the requested Statistics Group Data measured within that nonzero Measurement Duration. The reported change in data value shall be the value of the data at the end of the actual Measurement Duration minus the value of the data at the beginning of the actual Measurement Duration. If a STA accepts a Statistics Request measurement with nonzero, positive Measurement Duration, the STA shall perform the measurement over the requested Measurement Duration without regard to the Duration Mandatory bit in the Measurement Request Mode field. If a STA cannot measure over the requested Measurement Duration, the STA shall refuse the Statistics Request measurement.

If dot11RMStatisticsMeasurementActivated is false, a station shall reject the received Statistics Measurement Request and shall respond with a Statistics Measurement Report with the Incapable bit in the Measurement Report Mode field set to 1.

A STA may request that a STA Statistics report be sent when statistics of interest reach a threshold as defined in the Measurement Request element of the STA Statistics Request frame (see 8.4.2.23.9). This is termed a triggered STA Statistics measurement. Implementation of Triggered STA Statistics Reporting is optional for a WNM STA. A STA that implements Triggered STA Statistics Reporting has dot11MgmtOptionTriggerSTAStatisticsImplemented set to true. When dot11MgmtOptionTriggerSTAStatisticsImplemented is true, dot11WirelessManagementImplemented shall be true.

A triggered STA Statistic measurement shall be requested by setting the Enable and Report bits to 1 within a Measurement Request element containing the STA Statistics Measurement Type. The Measurement Request field shall contain a STA Statistics Request with the trigger conditions specified in the Triggered Reporting subelement, as defined in 8.4.2.23.9. One or more trigger conditions shall be set with specified thresholds. See 10.11.8. To prevent generation of too many triggered reports, the value of the Trigger Timeout field shall be set to a value greater or equal to the value of dot11MinTriggerTimeout. If the value of the Trigger Timeout field is less than the value of dot11MinTriggerTimeout, the STA shall reject the measurement request and respond with a report where the Measurement Report Mode field is “Incapable.”

A STA accepting a triggered STA Statistics measurement shall measure the requested statistics. If a trigger condition occurs, the measuring STA shall send a STA Statistics measurement report to the requesting STA. The measuring STA shall not send further triggered STA Statistics reports for that trigger condition to the requesting STA until the Trigger Timeout period specified in the request frame has expired. The STA Statistics measurement report is defined in 8.4.2.23.9. If the number of MPDUs or MSDUs indicated in the Measurement Count field are transmitted or received without any of the counted statistics meeting the corresponding trigger threshold then the measuring STA shall restart measuring for another measurement count window.

If a STA receives a STA Statistics measurement request from the same STA for which a triggered STA Statistics measurement is in progress, the in-progress triggered measurement shall be terminated.

STA Statistics reported in a triggered STA Statistics report shall be the values accumulated over the number of transmitted or received MSDUs or MPDUs before the trigger condition is met. Measurement duration shall not be specified when requesting a triggered STA statistics measurement and the Measurement Duration field in the corresponding Measurement Request shall be set to 0.

All triggered STA Statistics measurements shall be terminated at a measuring STA by receiving a STA Statistics measurement request with the Enable bit equal to 1 and the Report bit equal to 0. A STA requesting a triggered STA Statistics measurement may update the trigger conditions by sending a STA Statistics measurement request specifying the new trigger conditions.

Once accepted by a measuring STA, a triggered STA Statistics measurement continues to be active until the measurement request is superseded by a STA Statistics measurement request from the requesting STA or the measuring STA disassociates or reassociates.

10.11.9.6 Location Configuration Information Report

If dot11RMLCIMeasurementActivated is true, a STA shall reject any LCI Request for location information that is not available and shall respond with a Radio Measurement Report frame including a Radio Measurement Report element with the Refused bit set to 1. If dot11RMLCIMeasurementActivated is true and a STA accepts an LCI Request that does not include an Azimuth Request, it shall respond with a Radio Measurement Report frame including one LCI element (LCI Report). If both dot11RMLCIMeasurementActivated and dot11RMLCIAzimuthActivated are true, and the STA accepts an LCI request that includes an Azimuth Request, it shall respond with a Radio Measurement Report frame including one LCI element (LCI Report) that includes the requested Azimuth Report, if available. If dot11RMLCIAzimuthActivated is false, a STA shall reject any LCI Request that includes an Azimuth Request and shall respond with a Radio Measurement Report frame including an Radio Measurement Report element with the Incapable bit set to 1.

NOTE—Section 2.1 of IETF RFC 3825 (July 2004) defines formats and information fields for reporting physical location to sub-centimeter resolution. The fixed-point values have integer and fractional parts, which together represent Latitude, Longitude, and Altitude to a maximum resolution of 34 bits, 34 bits, and 30 bits, respectively. A Latitude report with 24-bit resolution would be reporting with a precision of about 3.18 m in Latitude at the equator. The physical location and azimuth MIB information of the STA might be set by administrative means.

The Datum value shall be 1 (World Geodetic System 1984), unless another datum is required for operation in the regulatory domain.

If the Altitude Type is 2 (Floors of Altitude), the value reported shall be as required for operation in the regulatory domain.

An LCI request shall indicate a location request for the requesting STA, the reporting STA, or a third STA with the MAC address specified in the Target MAC Address subelement, by setting the LCI request Location Subject field to indicate a Local, a Remote, or a third-party request, respectively. Local LCI Measurement Request is used by the requesting STA to obtain its own location by asking “Where am I?” Remote LCI Measurement Request is used by requesting STA to obtain location of reporting STA by asking “Where are you?” Third-party Location request is used by requesting STA to obtain location of a STA with the MAC address specified in the Target MAC Address subelement.

If the STA receiving an LCI request lacks the means to report the requested location to the requested resolution, then the LCI Report shall have that corresponding Latitude, Longitude, Altitude, or Azimuth resolution set to the known value; otherwise Latitude, Longitude, Altitude, and Azimuth fields shall be reported to their requested resolutions, with the remaining less significant bits set to 0.

If the STA receiving an LCI request has no location information about the requested LCI Subject physical location or requested Azimuth, it shall set the Incapable bit to 1 in the Measurement Report Mode field. The method by which the physical location and azimuth information in the LCI Report is generated is outside the scope of this standard.

NOTE—A STA that requested a “Local” LCI and received an LCI Report in which the Incapable bit is 1 can alternatively make a “Remote” LCI request to obtain the reporting STA’s physical location. A STA that requested an LCI including an Azimuth Request, and received an LCI Report in which the Incapable bit is 1 might alternatively request the LCI with no Azimuth requested.

If dot11RM3rdPartyMeasurementActivated is false, a STA shall reject any LCI Request that includes a LCI Request with the Location Subject field equal to 2 and shall respond with a Radio Measurement Report frame including an Radio Measurement Report element with the incapable bit set to 1.

It is optional for a STA to support an LCI Request and an LCI Report with the Location Subject field equal to 2. If dot11RM3rdPartyMeasurementActivated is true and a STA supports LCI Request and LCI Report, the following procedure shall be followed:

- When a non-AP STA requests the geospatial location of a STA with the MAC address specified in the Target MAC address field, it shall also include its own MAC address in the Originator Requesting STA MAC address field. When an AP receives an LCI Request with the Location Subject field value equal to 2, the AP shall generate an LCI Request to the STA with the MAC address specified in the Target MAC address field. If the AP does not have an association with the STA with the MAC address specified in the Target MAC address field, the AP shall reject the received LCI Request and shall respond with a LCI Report where the Incapable bit is set in the MeasurementReport Mode field. The AP shall copy the Originator Requesting STA MAC address and Target MAC address fields into the request from the received LCI request.
- When a STA receives an LCI Request with the Location Subject field value equal to 2, the STA shall only generate an LCI Report if the MAC address in the Target MAC address field is its own MAC address. When an LCI Report is generated, the reporting STA shall include its MAC address into the Target MAC address field and the MAC address present in the Originator Requesting STA MAC address field of the corresponding LCI Request into the Originator Requesting STA MAC address field. When an AP receives an LCI Report with an Originator Requesting STA MAC address field present, the AP shall generate an LCI Report to the associated STA with the MAC address specified in the Originator Requesting MAC address field. The AP shall copy the Originator Requesting STA MAC address and Target MAC address fields into the LCI report being transmitted to the originating requesting STA.

If dot11RMLCIMeasurementActivated is false, a station shall reject the received LCI Measurement Request and shall respond with a LCI Report with the Incapable bit in the Measurement Report Mode field set to 1.

If dot11RMLCIMeasurementActivated is true and a STA has its own location configured in LCI format, it shall set the Geospatial Location field to 1 in the Extended Capabilities element (see 8.4.2.29).

NOTE—It is recommended that User Applications not send location information to other stations without the express permission of the user. User agents acquire permission through a user interface, unless they have prearranged trust relationships with users. Those permissions that are acquired through the user interface and that are preserved beyond the current browsing session (i.e., beyond the time when the BSS connection is terminated) are revocable and receiving stations should respect revoked permissions. Some user applications might have prearranged trust relationships that do not require such user interfaces. For example, while a social networking application might present a user interface when a friend performs a location request, a VOIP telephone might not present any user interface when using location information to perform an E911 function.

10.11.9.7 Measurement pause

A measurement pause is used within a Measurement Request frame to provide a time delay between the processing of two other Measurement Request elements within the sequence of Measurement Request elements in that frame.

If a STA accepts a measurement pause request it shall delay processing of the next measurement request in the Measurement Request frame. If the measurement pause request is the last Request element in a repeated Measurement Request frame, the STA shall delay processing the first Request element in the Measurement Request frame for the next repeat. In each case the delay shall be no less than the Pause Time value specified in the measurement pause request.

NOTE—Measurement pause is always supported by a STA implementing Radio Measurements.

A measurement pause shall not be sent as the only Request element in a Measurement Request frame. A measurement pause shall not be included as the last Request element in a Measurement Request frame that has the Number of Repetitions field equal to 0.

A measurement pause cannot be processed in parallel to other measurements. If the Parallel bit is 1 in the Measurement Request element immediately prior to a measurement pause, an incapable response shall be returned even if dot11RMParallelMeasurementsActivated is true.

There is no measurement report associated with a measurement pause request.

10.11.9.8 Transmit Stream/Category Measurement Report

The Transmit Stream/Category Measurement applies to TIDs for Traffic Streams associated with TSPECs and also to TIDs for Traffic Categories for QoS traffic without TSPECs.

If dot11RMTransmitStreamCategoryMeasurementActivated is true and has no resource constraint that prevents it from being able to make the requested measurement, a QoS STA receiving a Transmit Stream/Category Measurement Request shall respond with a Radio Measurement Report frame containing one Measurement (Transmit Stream/Category Measurement) Report element. If the traffic stream (TS) that is corresponding to the Traffic Identifier is deleted, either by a DELTS Action frame or by disassociation, the STA shall cease sending Radio Measurement Reports.

If dot11RMTransmitStreamCategoryMeasurementActivated is false, a STA shall reject the received Transmit Stream/Category Measurement Request and shall respond with a Transmit Stream/Category Measurement Report with the Incapable bit in the Measurement Report Mode field set to 1.

The transmit stream/category measurement shall be made on traffic that is transmitted from the measuring QoS STA to the peer QoS STA and TID indicated in the request. The Peer STA Address may be the MAC address of the QoS STA from which the Measurement Request was sent, the MAC address of another QoS STA within the BSS, or the broadcast address. This enables a QoS AP to query Transmit Stream/Category Measurement metrics for DLS links. A broadcast address shall be used only with a TID corresponding to a TC. In the case of a broadcast address, measurement shall be made on all traffic for the specified TC. Depending on policy, a QoS AP may disallow transmit stream/category measurement requests for traffic to other QoS STAs in the BSS. In this case the QoS AP shall respond with a matching Measurement Report frame with the Incapable subfield of the Measurement Report Mode field set to 1.

If, during the course of a Transmit Stream/Category Measurement, any counter that is included in the Transmit Stream/Category Measurement Report increments to a value of $2^{32}–1$, the Transmit Stream/Category Measurement shall terminate, and the Transmit Stream/Category Measurement Report shall indicate the shortened, actual measurement duration.

If the measurement request included multiple transmit stream/category measurement requests for multiple TIDs, the corresponding measurement report shall include a transmit stream/category measurement report for each unique TID in the request that has been admitted. If the measurement request is for a TID that has not been admitted yet, a report is generated only after the TID becomes admitted.

The requesting and reporting STAs are QoS STAs. A non-QoS STA receiving a Transmit Stream/Category Measurement Request shall reject the request with indication of incapable.

A QoS STA may request that a measuring QoS STA send a transmit stream/category measurement report when the number of TID-specified MSDUs are discarded or delayed reaches a specified threshold. This is termed a triggered transmit stream/category measurement and shall be requested by setting the Enable and Report bits to 1 within a Measurement Request element containing the Transmit Stream/Category Measurement Type. The Measurement Request field shall contain a Transmit Stream/Category

Measurement Request with the trigger conditions specified in the Triggered Reporting subelement. One or more trigger conditions may be set with specified thresholds. See 8.4.2.23.11.

Depending on policy, a QoS AP may not permit the establishment of triggered transmit stream/category measurement. Such a QoS AP receiving a triggered transmit stream/category measurement request shall give an incapable indication. The number of simultaneous triggered transmit stream/category measurements supported at a QoS STA is outside the scope of the standard. A STA shall respond to further requests with a refused indication if the number of simultaneous triggered QoS measurements supported by the STA is reached.

If dot11RMTriggeredTransmitStreamCategoryMeasurementActivated is true, a QoS STA shall accept a triggered Transmit Stream/Category Measurement and shall reject it otherwise. A QoS STA accepting a triggered QoS measurement shall measure the requested TC or TS. If a trigger condition occurs, the measuring QoS STA shall send a Transmit Stream/Category Measurement Report to the requesting QoS STA. The measuring QoS STA shall not send further triggered QoS reports until the Trigger Timeout period specified in the request has expired or new trigger conditions have been requested. Measurement of transmit stream/category metrics shall continue during the reporting timeout period. Reporting shall resume following the Trigger Timeout period, or immediately following the acceptance of new trigger conditions.

If a QoS STA receives a Transmit Stream/Category Measurement Request for a TC, or TS that is already being measured using a triggered transmit stream/category measurement, the triggered traffic stream measurement shall be suspended for the duration of the requested traffic stream measurement. When triggered measurement resumes, the traffic stream metrics shall be reset.

Traffic stream metrics reported in a triggered transmit stream/category measurement report shall be the values accumulated over the number of successfully and unsuccessfully transmitted MSDUs prior to the trigger event given in the Measurement Count field of the transmit stream/category measurement request that established the trigger condition. It is possible that a consecutive or delay trigger event occurs after acceptance of a triggered transmit stream/category measurement but before the number of MSDUs in Measurement Count has been transmitted. In this case the report shall be the values accumulated since measurement started. The measurement count value appears in the Transmitted MSDU Count field of a triggered transmit stream/category measurement report. Measurement duration shall not be used in triggered QoS measurement, and the Measurement Duration field in both the Measurement Request and any Measurement Report shall be set to 0.

The Measurement Start Time field of a triggered transmit stream/category measurement report shall contain the value of the QoS STA TSF timer at the time the trigger condition occurred to an accuracy of 1 TU.

Once accepted by a measuring QoS STA, a triggered QoS measurement continues to be active until

- The relevant TS is deleted,
- The measuring QoS STA or QoS STA that requested the measurement disassociates or successfully reassociates, or
- The measurement is terminated by the requesting QoS STA.

All triggered QoS measurements shall be terminated at a measuring QoS STA by receiving a triggered transmit stream/category measurement request with the Enable bit equal to 1 and the Report bit equal to 0. A triggered QoS measurement request with no trigger conditions specified in the Trigger Conditions field shall terminate a triggered QoS measurement for the TC or TS specified in the request. A QoS STA requesting a triggered QoS measurement may update the trigger conditions by sending a triggered transmit stream/category measurement request specifying the new trigger conditions.

10.11.9.9 Location Civic report

If dot11RMCivicMeasurementActivated is true and civic location information is not available, the STA shall reject a Location Civic Request and shall respond with a Measurement Report frame including a Measurement Report element with the incapable bit set to 1. If dot11RMCivicMeasurementActivated is true and civic location information is available, the STA shall respond with a Measurement Report frame including one Location Civic Report element.

A Location Civic Request shall indicate a location request for the requesting STA, the reporting STA, or a third STA with the MAC address specified in the Target MAC address field, by setting the Location Civic request Location Subject field to indicate a Local, a Remote, or a third-party request respectively. Local Location Civic Measurement Request is used by requesting STA to obtain its own location by asking "Where am I?" Remote Location Civic Measurement Request is used by requesting STA to obtain location of the reporting STA by asking "Where are you?" Third-party Location request is used by requesting STA to obtain location of a STA with the MAC address specified in the Target MAC address field.

If dot11RMCivicMeasurementActivated is false, a STA shall reject the received Location Civic Measurement Request and shall respond with a Location Civic Report where the Incapable bit is set in the Measurement Report Mode field.

If dot11RM3rdPartyMeasurementActivated is false, a STA shall reject any LCI Request that includes a LCI Request with the Location Subject field equal to 2 and shall respond with a Radio Measurement Report frame including an Radio Measurement Report element with the incapable bit set to 1.

It is optional for a STA to support a Location Civic Request and a Location Civic Report with the Location Subject field equal to 2. If dot11RM3rdPartyMeasurementActivated is true and a STA supports Location Civic Request and Location Civic Report, the following procedure shall be followed:

- When a non-AP STA requests the civic location of a STA with the MAC address specified in the Target MAC address field, it shall also include its own MAC address in the Originator Requesting STA MAC address field. When an AP receives a Location Civic Request with the Location Subject field value equal to 2, the AP shall generate a Location Civic Request to the STA with the MAC address specified in the Target MAC address field. If the AP does not have an association with the STA with the MAC address specified in the Target MAC address field, the AP shall reject the received Location Civic Measurement Request and shall respond with a Location Civic Report where the Incapable bit is set in the Measurement Report Mode field. The AP shall copy the Originator Requesting STA MAC address and Target MAC address fields into the request from the received Location Civic Request.
- When a STA receives a Location Civic Request with the Location Subject field value equal to 2, the STA shall only generate a Location Civic Report if the MAC address in the Target MAC address field is its own MAC address. When a Location Civic Report is generated, the reporting STA shall include its MAC address into the Target MAC address field and the MAC address present in the Originator Requesting STA MAC address field of the corresponding Location Civic Request into the Originator Requesting STA MAC address field. When an AP receives a Location Civic Report with an Originator Requesting STA MAC address field present, the AP shall generate a Location Civic Report to the associated STA with the MAC address specified in the Originator Requesting MAC address field. The AP shall copy the Originator Requesting STA MAC address and Target MAC address fields into the Location Civic Report being transmitted to the originating requesting STA.

When a STA receives a Location Civic Request with the subject field equal to 2, but does not support third-party location, it shall respond with a Radio Measurement Report frame including a Radio Measurement Report element with the incapable bit set to 1.

If dot11RMCivicMeasurementActivated is true and a STA has its own location configured in Civic format, it shall set the Civic Location field to 1 in the Extended Capabilities element.

If dot11RMCivicMeasurementActivated is true and a STA has its own location available in one or more location formats, as defined in Table 8-77, and includes Civic Location Type Value 0, indicating "IETF RFC4776-2006," it shall set the Civic Location field to 1 in the Extended Capabilities element.

NOTE—User Applications should not send location information to other stations without the express permission of the user. User agents acquire permission through a user interface, unless they have prearranged trust relationships with users. Those permissions that are acquired through the user interface and that are preserved beyond the current browsing session (i.e., beyond the time when the BSS connection is terminated) are revocable and receiving stations should respect revoked permissions. Some user applications may have prearranged trust relationships that do not require such user interfaces. For example, while a social networking application might present a user interface when a friend performs a location request, a VOIP telephone may not present any user interface when using location information to perform an E911 function.

If the Location Civic Report contains the Location Reference and Location Shape subelements, the receiving STA may use the information specified in those subelements in combination with the Civic Location field value for additional granularity on the position reported in the Civic Location field.

If the Location Civic Report contains the Map Image subelement, the receiving STA's SME may retrieve the floor plan specified by the Map URL field. The method to retrieve the floor plan specified by the Map URL field is out of scope of this document.

10.11.9.10 Location Identifier Report

The Location Identifier Report provides the ability for a STA to receive an indirect URI reference and forward that reference to an external agent for the purposes of that agent gathering the STA's location value. The protocol used to query for a location report based on the Public Identifier URI provided in the Location Identifier Report is beyond the scope of this standard.

If dot11RMIdentifierMeasurementActivated is true and location information is not available, the STA shall reject any Location Identifier Request and shall respond with a Measurement Report frame including a Measurement Report element with the incapable bit set to 1. If dot11RMIdentifierMeasurementActivated is true and location information is available, the STA shall respond with a Measurement Report frame including one Location Identifier Report element.

A Location Identifier Request shall indicate a location request for the requesting STA, the reporting STA, or a third-party STA with the MAC address specified in the Target MAC address field, by setting the Location Identifier request Location Subject field to indicate a Local, a Remote, or a Third-party request respectively. Local Location Identifier Request is used by requesting STA to obtain its own location by asking "Where am I?" Remote Location Civic Measurement Request is used by requesting STA to obtain location of the reporting STA by asking "Where are you?" Third-party Location request is used by requesting STA to obtain location of a STA with the MAC address specified in the Target MAC address field.

If dot11RM3rdPartyMeasurementActivated is false, a STA shall reject any LCI Request that includes a LCI Request with the Location Subject field equal to 2 and shall respond with a Radio Measurement Report frame including an Radio Measurement Report element with the incapable bit set to 1.

It is optional for a STA to support a Location Identifier Request and a Location Identifier Report with the Location Subject field equal to 2. If dot11RM3rdPartyMeasurementActivated is true and a STA supports Location Identifier Request and Location Identifier Report, the following procedure shall be followed:

— When a non-AP STA requests the location identifier of a STA with the MAC address specified in the Target MAC address field, it shall also include its own MAC address in the Originator Requesting STA MAC address field. When an AP receives a Location Identifier Request with the

Location Subject field value equal to 2, the AP shall generate a Location Identifier Request to the STA with the MAC address specified in the Target MAC address field. If the AP does not have an association with the STA with the MAC address specified in the Target MAC address field, the AP shall reject the received Location Identifier Request and shall respond with a Location Identifier Report where the Incapable bit is set in the Measurement Report Mode field. The AP shall copy the Originator Requesting STA MAC address and Target MAC address fields into the request from the received Location Identifier Request.

— When a STA receives a Location Identifier Request with the Location Subject field value equal to 2, the STA shall only generate a Location Identifier Report if the MAC address in the Target MAC address field is its own MAC address. When a Location Identifier Report is generated, the reporting STA shall include its MAC address into the Target MAC address field and the MAC address present in the Originator Requesting STA MAC address field of the corresponding Location Identifier Request into the Originator Requesting STA MAC address field. When an AP receives a Location Identifier Report with an Originator Requesting STA MAC address field present, the AP shall generate a Location Identifier Report to the associated STA with the MAC address specified in the Originator Requesting MAC address field. The AP shall copy the Originator Requesting STA MAC address and Target MAC address fields into the Location Identifier Report being transmitted to the originating requesting STA.

When a STA receives an Location Identifier Request with the subject field equal to 2, but does not support third-party location, it shall respond with a Radio Measurement Report frame including a Radio Measurement Report element with the incapable bit set to 1.

If dot11RMIdentifierMeasurementActivated is false, a STA shall reject the received Location Identifier Request and shall respond with a Location Identifier Report where the Incapable bit is set in the Measurement Report Mode field.

If dot11RMIdentifierMeasurementActivated is true and a STA has its own location available in one or more location formats, as defined in Table 8-77, and includes Civic Location Type Value 0, indicating “IETF RFC4776-2006” or Geospatial format that can be referenced by a location identifier, it shall set the Identifier Location field to 1 in the Extended Capabilities element.

NOTE—User Applications should not send location information to other stations without the express permission of the user. User agents acquire permission through a user interface, unless they have prearranged trust relationships with users. Those permissions that are acquired through the user interface and that are preserved beyond the current browsing session (i.e., beyond the time when the BSS connection is terminated) are revocable and receiving stations should respect revoked permissions. Some user applications may have prearranged trust relationships that do not require such user interfaces. For example, while a social networking application might present a user interface when a friend performs a location request, a VOIP telephone may not present any user interface when using location information to perform an E911 function.

10.11.10 Usage of the neighbor report

10.11.10.1 General

A neighbor report is sent by an AP and it contains information on neighboring APs that are members of ESSs requested in the neighbor report request. A neighbor report may not be exhaustive either by choice, or due to the fact that there may be neighbor APs not known to the AP. The neighbor report contents are derived from the NeighborListSet parameter of the MLME-NEIGHBORREPRESP.request primitive. The mechanism by which the contents of this table are determined is outside the scope of this standard, but it may include information from measurement reports received from the STAs within the BSS, information obtained via a management interface, or the DS.

NOTE—The purpose of the neighbor report is to enable the STA to optimize aspects of neighbor service set transition and ESS operation. A Neighbor Report element contains information on APs that the STA might use as candidates for a service set transition. Since the information in the neighbor report might be stale, it is advisory; information obtained by the report recipient through a scan or other sources might also be considered, possibly overriding information in the neighbor report. For example, where information contained within a neighbor report is contradicted by information in the Measurement Pilot, Beacon, or Probe Response, that response information needs to take precedence.

10.11.10.2 Requesting a neighbor report

A STA requesting a neighbor report from an AP shall send a Neighbor Report Request frame to its associated AP.

10.11.10.3 Receiving a neighbor report

If dot11RMNeighborReportActivated is true, an AP receiving a neighbor report request shall respond with a Neighbor Report Response frame containing zero or more Neighbor Report elements. If an SSID element is specified in the corresponding Neighbor Report Request frame, the Neighbor Report element(s) shall contain information only concerning neighbor APs that are members of the current ESS identified by the SSID element contained within the neighbor report request. If the SSID element is omitted, the Neighbor Report element(s) shall contain information concerning neighbor APs that belong to the same ESS as the requesting STA. If the wildcard SSID element is specified in the corresponding Neighbor Request frame, the Neighbor Report element(s) shall contain information concerning all neighbor APs. If there are no neighbor APs available, the AP shall send a Neighbor Report Response frame with no Neighbor Report elements.

If dot11RMNeighborReportActivated is false in an AP receiving a neighbor report request, it shall ignore the request and return a Neighbor Report frame with the Incapable bit in the Measurement Report Mode field set to 1.

A STA receiving a neighbor report element with an unknown subelement identifier shall ignore the unknown subelement and continue to process remaining subelements. A STA receiving a neighbor report element containing a Vendor Specific subelement with an unknown Organization Identifier should ignore this vendor-specific subelement and shall continue to process any remaining Vendor Specific subelements.

A serving AP shall include a TSF Information subelement in the Neighbor Report element if it is able to guarantee an accumulated error of 1.5 TU or better on the TSF Offset subfield. Otherwise, the AP shall not include a TSF Information subelement in the Neighbor Report element.

10.11.11 Link Measurement

A STA may use a Link Measurement Request frame to request another STA to respond with a Link Measurement Report frame containing a TPC report element. If dot11RMLinkMeasurementActivated is true, a STA receiving a Link Measurement Request frame shall respond with a Link Measurement Report frame containing a TPC Report element indicating the power used to transmit the Link Measurement Report. The Link Measurement Report also contains antenna ID and signal quality (RCPI and RSNI).

If dot11RMLinkMeasurementActivated is false in an AP receiving a Link Measurement Request, it shall ignore the request.

10.11.12 Measurement of the RPI histogram

To compute RPI densities for an RPI Histogram report (see 8.4.2.24.4), the STA shall measure the received power level on the specified channel, as detected at the antenna connector, as a function of time over the measurement duration. The maximum tolerance of the received power measurements shall be ± 5 dB. Furthermore, the received signal power measurement should be a monotonic function of the actual power at the antenna. The time resolution of the received power measurements is in microseconds. The received power measurements are converted to a sequence of RPI values by quantizing the measurements according to Table 8-82. The RPI densities are then computed for each of the eight possible RPI values using

$$\text{RPI Density} = \text{Ceiling}\left\lceil\frac{255 \times D_{RPI}}{1024 \times D_M}\right\rceil$$

where
D_{RPI} is the duration receiving at the specified RPI value (µs)
D_M is the measurement duration (TU).

The sum of the RPI densities is approximately 255, but could be up to 262 because of rounding effects.

10.11.13 Operation of the Max Transmit Power field

The maximum tolerance for the value reported in Max Transmit Power field shall be 5 dB. The value of the Max Transmit Power field shall be less than or equal to the Max Regulatory Power value for the current channel.

10.11.14 Multiple BSSID Set

A Multiple BSSID Set is characterized as follows:

- All members of the set use a common operating class, channel, Channel Access Functions, and antenna connector.
- The set has a maximum range of 2^n for at least one n, where $1 \le n \le 46$.
- Members of the set have the same 48-n MSBs in their BSSIDs.
- All BSSIDs within the Multiple BSSID Set are assigned in a way that they are not available as MAC addresses for STAs using a different operating class, channel or antenna connector.

NOTE—For example, if the APs within BSSs with BSSIDs 16, 17, and 27 share the operating class, channel and antenna connector, and the range of MAC addresses from 16–31 inclusive are not assigned to other STAs using a different antenna connector, then the BSSIDs 16, 17, and 27 are members of a Multiple BSSID set. The set is described by n = 4 ($2^n = 16$) with BSSIDs in the range 0x00000000001X. The set cannot be described by n = 8 for instance since at least one of the BSSIDs in the range 0x0000000000XX might be used as a BSSID by an AP that does not share the same operating class, channel, and antenna connector.

When the Multiple BSSID set contains two or more members, the transmission of Measurement Pilots is constrained as described in 10.11.15.

A Multiple BSSID element, with or without optional subelements, indicates that all APs within the indicated range of BSSIDs transmit using a common class, channel, and antenna connector.

A single Beacon frame may contain elements for the Multiple BSSID Set members; see 10.1.3.6.

10.11.15 Measurement Pilot generation and usage

10.11.15.1 General

The Measurement Pilot frame is a compact Action frame transmitted pseudo-periodically by an AP at a small interval relative to a Beacon Interval. The Measurement Pilot frame provides reduced information relative to a Beacon frame to allow for the required small interval. The purpose of the Measurement Pilot frame is to assist a STA with the following functions:

- Rapid discovery of the existence of a BSS via passive scanning
- Rapid collection of neighbor AP signal strength measurements via passive scanning
- Enable transmission of a Probe Request

The value of dot11RMMeasurementPilotActivated in a STA determines the level of support for Measurement Pilot at the STA. Table 10-7 describes permitted values for dot11RMMeasurementPilotActivated and what they signify.

Table 10-7—Measurement Pilot Activated definition

Device function	dot11RM-MeasurementPilot Activated	Notes
AP and non-AP STA	0	If the STA is contained within an AP, it does not generate MPs and if the device is a non-AP STA, it ignores the MPs it receives
AP and non-AP STA	1	If the STA is contained within an AP, it can transmit MPs, and if the device is a non-AP STA, it can receive the MPs and can use the information contained in MPs.
Non-AP STA	2	The non-AP STA is making use of the MPs it receives or would receive if they were being transmitted.
Non-AP STA	3–7	Reserved
AP		The AP is transmitting MPs and using the information contained in them, and the AP is actively transmitting MPs with MP interval set to a value within the range as shown below. MP Interval with respect to Beacon Interval:
	2	> 3% and < 5% of Beacon Interval
	3	≥ 5% and < 10%
	4	≥ 10% and < 15%
	5	≥ 15% and < 20%
	6	≥ 20% and < 25%
	7	≥ 25% and < 50%

10.11.15.2 Measurement Pilot generation by an AP

The AP shall determine if it is a member of a Multiple BSSID Set with two or more members. If so, at most one AP of the Multiple BSSID Set shall have dot11RMMeasurementPilotActivated set to a value between 2 and 7. How this occurs is out of scope of this standard.

If dot11RMMeasurementPilotActivated is between 2 and 7, the following statements apply:

— If the AP is a member of a Multiple BSSID Set with two or more members, then the BSSIDs of all members of the Multiple BSSID Set shall be indicated in the Beacon and Probe Response frames by the Multiple BSSID subelement.

— The AP shall maintain a Measurement Pilot generation function, which transmits Measurement Pilot frames at a basic rate according to dot11RMMeasurementPilotPeriod.

— The AP defines a series of TMPTTs exactly dot11RMMeasurementPilotPeriod apart. A TMPTT arrives when the AP's local TSF timer (in µs) modulo the Measurement Pilot Interval equals 0.

— At each TMPTT, the AP shall schedule a Measurement Pilot as the next frame for transmission ahead of other queued frames using the AC_VO EDCA parameters unless the TMPTT satisfies:

$$\mathrm{TBTT} - \frac{T_{MPP}}{2} \le \mathrm{TMPTT} < \mathrm{TBTT} + \frac{T_{MPP}}{2}$$

where T_{MPP} is dot11RMMeasurementPilotPeriod, for any TBTT of members of the Multiple BSSID Set, in which case the AP shall not generate the Measurement Pilot. This is illustrated in Figure 10-18. How the AP determines the TBTTs of members of the Multiple BSSID Set is out of scope of this standard.

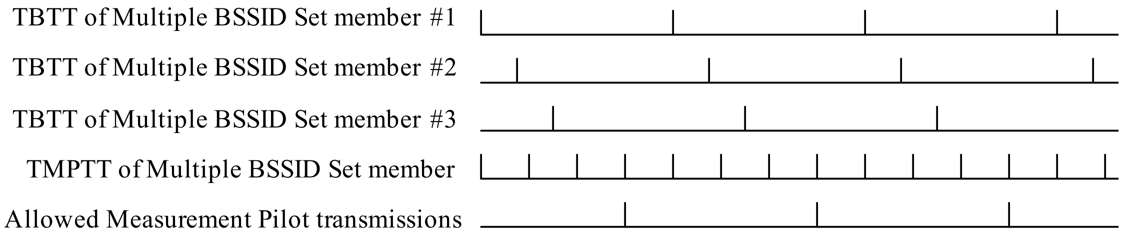

TBTT of Multiple BSSID Set member #1
TBTT of Multiple BSSID Set member #2
TBTT of Multiple BSSID Set member #3
TMPTT of Multiple BSSID Set member
Allowed Measurement Pilot transmissions

Figure 10-18—Example of Measurement Pilot Scheduling

In case the medium is determined by the carrier-sense mechanism (see 9.3.2.1) to be unavailable at the TMPTT, the AP shall delay the actual transmission of a Measurement Pilot according to the basic medium access rules specified in Clause 9 for a maximum period of one dot11RMMeasurementPilotPeriod and drop the delayed Measurement Pilot at the next TMPTT. In this way, a continuously busy medium causes multiple successive Measurement Pilots to be delayed, then dropped. An AP shall transmit Measurement Pilots to the broadcast address. An AP shall not retransmit or buffer Measurement Pilots as part of the PSP mechanisms.

— If the AP is a member of a Multiple BSSID Set with two or more members, then the BSSIDs of all members of the Multiple BSSID Set shall be indicated in the Measurement Pilot using the Multiple BSSID subelement.

NOTE 1—APs are advised to enable Measurement Pilots judiciously due to the possibility of excessive medium time being consumed by Measurement Pilots from multiple overlapping APs. For instance dot11RMMeasurementPilotTransmissionInformationActivated might be set to false:

a) When enabling Measurement Pilots would cause:
   1) More than 10% of the medium time at the AP to be consumed by beacons and Measurement Pilots transmitted by any source, or
   2) More than 5% of the medium time at the AP to be consumed by Measurement Pilots transmitted by any source.
b) When STAs are not expected to be using Measurement Pilots. How this is determined is out of the scope of this standard, but may depend upon many STAs setting the Measurement Pilot Capability field in the Supported RM Capabilities Enabled bitmask element to 0 or 1 upon association at any member of the Multiple BSSID Set recently or at similar times in the past.
c) When all members of the Multiple BSSID Set are within ESSs that contain one BSS only.
d) When the AP's operating regulatory domain is not subject to DFS regulations.
e) When the AP's operating regulatory domain is subject to DFS regulations but compliance with the regulations is impaired by Measurement Pilots.
f) When the number of channels valid for the AP's operating regulatory domain or frequency band is small.
g) When no members of the Multiple BSSID Set are located at ingress or egress points of an ESS, so are less useful for roaming between an IEEE 802.11 ESS and other networks.

NOTE 2—For efficient use of the medium, it is recommended that Measurement Pilots not be sent using a PHY specified in Clause 16 or Clause 17.

10.11.15.3 Measurement Pilot usage by a STA

Whenever testing a requested BSSID for equality against the BSSID of a Measurement Pilot, the following statements apply:

- If the Measurement Pilot does not contain the Multiple BSSID element, then equality shall be true if the requested BSSID equals the BSSID of the Measurement Pilot frame, and otherwise false.
- If the Measurement Pilot contains the Multiple BSSID element, and the requested BSSID is a non-wildcard BSSID, then equality shall be true if the requested BSSID equals any BSSID indicated by the Multiple BSSID element present in the Measurement Pilot, and otherwise false.
- If the Measurement Pilot contains the Multiple BSSID element, and the requested BSSID is the wildcard BSSID, then equality shall be true.

NOTE—STAs are advised that due to considerations such as those noted in the prior subclause, APs might not transmit Measurement Pilots at all times or in all bands.

10.11.16 Access Delay Measurement

Access delay is measured by the AP's MAC layer being the average medium access delay for transmitted frames measured from the time the MPDU is ready for transmission (i.e., begins CSMA/CA access) until the actual frame transmission start time. Access delay measurement results are included in the BSS Average Delay element and in the BSS AC Access Delay element.

For the BSS Average Delay measurement, the AP shall measure and average the medium access delay for all transmit frames using the DCF or EDCAF over a continuous 30 s measurement window. For the infrastructure BSS AC Access Delay measurement, the QoS AP shall measure and average the medium access delay for all transmit frames of the indicated AC (see Figure 8-227) using EDCA mechanism over a continuous 30 s measurement window. The accuracy for the average medium access delay shall be ± 100 μs or better when averaged over at least 200 frames. Accuracy is not defined for measurements averaged over less than 200 frames.

10.11.17 BSS Available Admission Capacity

BSS Available Admission Capacity provides a means for an AP to advertise admission capacity available for explicit admission control in any UP or AC. This information may assist STAs in making service set transition decisions.

The transmitted BSS Available Admission Capacity value represents a proportion of time on the wireless medium scaled linearly in units of 32 μs/s from 0 (0% available time) to 31 250 (100% available time). If an AP transmits a BSS Load element, the values for any transmitted BSS Available Admission Capacity values shall be less than or equal to the Available Admission Capacity field value of the BSS Load value. If an AP transmits a BSS Available Admission Capacity element, the transmitted values should be current or recently calculated. The AP recalculates Available Admission Capacity values according to local policy. An Available Admission Capacity value of 0 transmitted in the BSS Available Admission Capacity element indicates that no admission capacity is available at the calculation time and that no explicit admissions can be granted by the AP for that UP or AC unless additional capacity becomes available. An AP that receives a TSPEC admission request for total medium time (in both directions, if applicable) that is less than or equal to the current available admission capacity for the requested UP or AC local policy may apply additional local policy before admitting the requested TSPEC.

NOTE 1—Available Admission Capacity values are dynamic in a BSS and the transmitted values cannot always reflect the actual values currently used by the AP for explicit admission control. Thus an AP should recalculate the Available Admission Capacity values regularly or after changes in the environment or the admitted capacity.

NOTE 2—STAs are advised that requesting admission for any TSPEC at an UP or AC that requires more medium time than is reported as available for the requested UP or AC is possible yet unlikely to be successful.

10.11.18 AP Channel Report

The AP Channel Report element contains a list of channels in an operating class where a STA is likely to find an AP, excluding the AP transmitting the AP Channel Report. An AP Channel Report element only includes channels that are valid for the regulatory domain in which the AP transmitting the element is operating and consistent with the Country element in the frame in which it appears. One AP Channel Report element is included in the Beacon frame for each regulatory domain, which includes channels on which a STA is likely to find an AP.

The contents of the AP Channel Report elements may be compiled from the list of unique operating/channel pairs found in the neighbor report. The contents of the AP channel report may be configured or obtained by other means beyond the scope of this standard.

10.11.19 Multicast diagnostic reporting

Multicast diagnostic reporting enables an AP to collect statistics on group addressed traffic at associated STAs. The method an AP uses to determine the multicast groups to which an associated STA is a member of is outside the scope of the standard, and is typically performed by higher layer protocols. The Multicast Diagnostic Request and Multicast Diagnostic Report fields are defined in 8.4.2.23.13 and 8.4.2.24.12, respectively.

A STA that has a value of true for dot11MgmtOptionMulticastDiagnosticsActivated is defined as a STA that supports multicast diagnostics reporting. A STA for which dot11MgmtOptionMulticastDiagnosticsActivated is true shall set the Multicast Diagnostics field of the Extended Capabilities element to 1. When the Multicast Diagnostics field in the Extended Capabilities field is 1, the Incapable bit in the Measurement Report Mode field of a Multicast Diagnostic Report shall not be set to 1.

Multicast diagnostic reporting may use the triggered autonomous reporting capability described in 10.11.8.

The Measurement Start Time field of a triggered diagnostic report shall contain the value of the STA TSF Timer at the time the trigger condition started to occur to an accuracy of ±1 TU.

An AP may send a Multicast diagnostic request consisting of one or more Multicast Diagnostic Request fields in a Radio Measurement Request frame to a non-AP STA that has indicated support of the multicast diagnostic capability or to a multicast group address if all associated non-AP STAs support the multicast diagnostic capability. If the STA accepts the request it shall count the number of received MSDUs with the specified group address and the STA shall record the maximum observed PHY data rate of the frames that contained these MSDUs during the requested Measurement Duration. These two values shall be returned in a Multicast Diagnostic Report Measurement Report in a Radio Measurement Report frame, as defined in 8.4.2.24.12. A non-AP STA shall not transmit a Radio Measurement Request frame containing a Multicast Diagnostic Request. A STA shall not respond to a Radio Measurement Request frame containing a Multicast Diagnostic Request received from a STA other than the AP with which it is associated.

An AP may request that triggered Multicast Diagnostic reporting be enabled at associated non-AP STAs that have indicated support of the multicast diagnostic capability. To enable Multicast Diagnostic reporting, the AP shall send a Measurement Request element containing a Multicast Diagnostic Request Type and with the Enable and Report bits set to 1 within a Radio Measurement Request frame. See 10.11.8. For triggered Multicast Diagnostic reporting, the Multicast MAC Address and trigger conditions for diagnostic reporting shall be specified in the request.

Multicast Diagnostic reporting may be requested for broadcast traffic by setting the Multicast MAC Address field to the broadcast address.

A non-AP STA accepting a request for triggered multicast diagnostic reporting shall send a Multicast Diagnostic Report to the requesting STA if the specified trigger condition occurs. The measuring non-AP STA shall not send further triggered Multicast Diagnostic Reports until the period specified in the Reactivation Delay field in the Multicast Diagnostic Request has expired, or the non-AP STA receives a revised trigger condition from a Multicast Diagnostic Request. To prevent generation of too many triggered reports, the minimum value of the Reactivation Delay field shall be set to a value greater or equal to the value of dot11MinTriggerTimeout.

Once accepted, Multicast Diagnostic reporting continues to be active for the specified Multicast MAC address until one of the following occurs:

- The STA receives a Measurement Request element containing a Multicast Diagnostic Request Type and with the Enable bit equal to 1 and the Report bits equal to 0 within a Radio Measurement Request frame.
- Receipt of a Measurement Request element containing a Multicast Diagnostic Request Type, with the Enable and Report bits equal to 1, but with no trigger conditions.
- The STA leaves the Multicast Group or disassociates.
- The STA sends a Measurement Report element with the Measurement Result bit set to 1.

The STA shall maintain an inactivity timer for every multicast diagnostic request accepted by the STA in which the Inactivity Timeout Request field is 1. When a timeout of the inactivity timer is detected, the STA shall send a multicast diagnostic report with the inactivity Timeout Trigger field in the Multicast Reporting Reason field set to 1. The inactivity timer at a recipient is reset when MSDUs corresponding to the monitored group address are received.

A STA that declines a request for triggered multicast diagnostic reporting sends a Measurement Report element (as described in 8.4.2.24) in a Radio Measurement Report frame (as described in 8.5.2.3) with the Measurement Report Mode field set appropriately to indicate the reason for a failed or rejected request.

10.12 DSE procedures

10.12.1 General

Regulations that apply to the U.S. 3650 MHz band require enabling STAs to implement a mechanism to enable mobile and portable STA operation. Similar regulations exist in other regulatory domains. This standard describes such a mechanism, referred to as dependent STA enablement (DSE).

Subclause 10.12 describes DSE procedures that can be used to satisfy the U.S. 3650 MHz band and similar regulatory requirements. Regulations that apply to the U.S. 3650 MHz band require fixed STAs and enabling STAs to have their operating locations registered. Licensees with STAs suffering or causing harmful interference are expected to cooperate and resolve problems by mutually satisfactory arrangements. The DSE procedures provide the location of the enabling STA and unique identifiers to assist licensees in the resolution of interference issues. The DSE procedures might also satisfy needs in other frequency bands and be useful for other purposes.

STAs shall use the DSE procedures defined in this subclause if dot11LCIDSERequired is true. dot11DSERequired and dot11ExtendedChannelSwitchActivated shall be true when regulatory authorities require DSE, with the following exceptions: dot11DSERequired shall be set to false to configure STAs to operate as registered STAs, and dot11ExtendedChannelSwitchActivated may be set to false when operating as a fixed STA. A summary of STA attributes and these MIB attributes are shown in Table 10-8.

A fixed STA is a registered STA that broadcasts its registered location and is restricted from enabling other STAs (see 10.12.3). An enabling STA is a registered STA that broadcasts its registered location, and

Table 10-8—DSE STA attributes

Type of STA	Registered STA	dot11LCIDSERequired and dot11LCIDSEImplemented	dot11Extended ChannelSwitch Activated	dot11DSERe quired
Fixed STA	Yes	true	true or false	false
Enabling STA	Yes	true	true	false
Dependent STA	No	true	true	true

regulatory authorities permit it to enable operation of unregistered STAs (see 10.12.4). A dependent STA is an unregistered STA that operates under the control of an enabling STA (see 10.12.5). When management frame protection is negotiated, stations shall use Protected Dual of Public Action frames instead of individually addressed Public Action frames for DSE procedures.

The DSE procedures provide for the following:

- Registered STA operation
- Creation of a DSE service area for dependent STA operation
- Dependent STA operation with DSE

10.12.2 Enablement and deenablement

10.12.2.1 General

This subclause describes the procedures used for IEEE 802.11 enablement and deenablement. A STA keeps a state variable for each STA with which enablement communication is needed:

- Enablement state with a value of *unenabled* or
- Enablement state with a value of *enabled*

NOTE—Refer to 10.12.5 for description of dependent STA operation in either enablement state.

Enablement utilizes a two-message transaction sequence. The first message asserts identity and requests enablement. The second message returns the enablement result. In the description in 10.12.2.2 and 10.12.2.3, the STA initiating the enablement is referred to as *enablement requester*, and the STA to which the initial frame in the exchange is addressed is referred to as *enablement responder*. The specific items in each of the messages described in the following subclauses are defined in 8.5.8.4 and 8.5.8.5. An enabling STA may decline to enable a requesting STA. If the result is “successful,” the requesting STA shall be enabled.

Deenablement utilizes a one-message transaction sequence. In the description in 10.12.2.4 and 10.12.2.5, the STA initiating the deenablement is referred to as *deenablement requester*, and the STA to which the frame is addressed is referred to as *deenablement responder*.

10.12.2.2 Enablement requester STA

Upon receipt of an MLME-ENABLEMENT.request primitive, the enablement requester STA shall perform the following procedure:

a) If one or more request parameters are invalid, issue an MLME-ENABLEMENT.confirm primitive with ReasResultCode set to INVALID_PARAMETERS; else

b) Construct and transmit a DSE Enablement frame requesting enablement.

   1) Specific information items in the enablement message are as follows.

i) STA identity assertion (in RequesterSTAAddress)

ii) Enabling STA identity assertion (in ResponderSTAAddress)

iii) Reason result code = 2

iv) Enablement identifier = 0

2) Specific items in the enablement message sent by the enablement responder STA are described in 10.12.2.3.

c) On receipt of a matching DSE Enablement frame (i.e., the Requester STA Address and Responder STA Address mach the pending request) that acknowledges the DSE Enablement frame, issue an MLME-ENABLEMENT.confirm primitive to inform the SME of the result of the enablement.

1) The primitive may contain information from an enablement response message received from the enabling STA (see 10.12.2.3), or it may be issued for another reason (see 8.5.8.4).

2) The reason result code in the enablement confirmation message indicates when the enablement is successful.

3) If the enablement was successful, the enablement state variable for the enablement responder STA shall be set to Enabled.

d) If no matching DSE Enablement frame is received for the DSE Enablement frame within a period of EnablementTimeLimit TUs measured from the receipt of the MLME-ENABLEMENT.request primitive, issue an MLME-ENABLEMENT.confirm primitive with ResultCode set to TIMEOUT.

10.12.2.3 Enablement responder STA

Upon receipt of a Public Action DSE Enablement frame with a reason result code of 2, the enablement responder STA may enable the enablement requester STA using the following procedure:

a) Create and transmit a response frame with the enablement status as defined in 8.5.8.4 set in the Reason Result Code field and with a dependent enablement identifier chosen to be unique among all dependent enablement identifiers that have been assigned if enablement was successful. The enablement responder shall transmit the ResultCode parameter from the .confirm primitive mapped as specified in Table 8-213 of 8.5.8.4 to indicate the result of the enablement request.

1) Specific information items in the enablement message response are as follows:

i) STA identity assertion (in RequesterSTAAddress)

ii) Enabling STA identity assertion (in ResponderSTAAddress)

iii) The result of the requested enablement as defined in 8.5.8.4

iv) Enablement identifier

b) Issue an MLME-ENABLEMENT.indication primitive to inform the SME of the enablement.

1) If the enablement is successful, the enablement state variable for the enablement requester STA shall be set to Enabled.

10.12.2.4 Deenablement requester STA

Upon receipt of an MLME-DEENABLEMENT.request primitive, the deenablement requester STA shall create and transmit a DSE Deenablement frame to the deenablement responder STA. Specific information items in the deenablement message are as follows:

a) Enabling STA identity assertion (in RequesterSTAAddress)

b) STA identity assertion (in ResponderSTAAddress)

c) Reason result code = 2

10.12.2.5 Deenablement responder STA

Upon receipt of a DSE Deenablement frame, the deenablement responder STA shall deenable with the deenablement requester STA by issuing an MLME-DEENABLEMENT.indication primitive to inform the SME of the deenablement. The enablement state variable for the deenablement requester STA shall be set to Unenabled.

10.12.3 Registered STA operation

Registered STAs shall have dot11DSERequired set to false. They shall transmit the DSE Registered Location element in every Beacon frame and shall set the Dependent STA bit in the DSE Registered Location element to 0. If the registered STA is located within a national policy area, such as a Fixed Satellite Service exclusion zone, or within an international agreement area near a national border, the RegLoc Agreement bit in the DSE Registered Location element shall be set to 1, signifying to other STAs that additional restrictions on STAs with directional antennas may apply; otherwise, it shall be set to 0.

The Latitude, Longitude, and Altitude fields of the DSE Registered Location element shall be reported at their best known resolutions, which may exceed the resolutions required by regulatory authorities. The Altitude Type field value shall be 3 (i.e., height above ground is in meters or, in other words, the altitude is in meters above adjacent terrain), unless another altitude type is required for operation in the regulatory domain. The Datum field value shall be 1 (World Geodetic System 1984), unless another datum is required for operation in the regulatory domain.

An enabling STA is a registered STA that broadcasts its registered location, and regulatory authorities permit it to enable operation of unregistered STAs (see 10.12.4). A dependent STA is an unregistered STA that operates under the control of an enabling STA (see 10.12.5).

A fixed STA is a registered STA that broadcasts its registered location and is restricted from enabling other STAs. A fixed STA shall have dot11LCIDSERequired set to true, and dot11ExtendedChannelSwitchActivated may be set to true or false. A fixed STA shall set dot11RegLocDSE to false and the RegLoc DSE bit in the DSE Registered Location element to 0, signifying that it is not creating a DSE service area. A fixed STA may operate in an infrastructure BSS or IBSS. A registered STA that is not an enabling STA may operate as an AP in an infrastructure BSS and relay Public Action frames (specifically, DSE Enablement, DSE Deenablement, DSE Measurement Request, DSE Measurement Report, DSE Power Constraint) from a dependent STA to its enabling STA. The specification of the algorithm by which Public Action frames are relayed is beyond the scope of this standard. Note that the enabling signal is not a Public Action frame and is not relayed (see 10.12.4).

10.12.4 Enabling STA operation with DSE

A registered STA may create and manage a DSE service area for dependent STA operation where regulatory requirements permit. A registered STA operating in this manner is referred to as an *enabling STA*. An enabling STA sets dot11RegLocDSE to true and signifies the creation of a DSE service area by setting the RegLoc DSE bit in the DSE Registered Location element to 1. Dependent STA transmission of any frames is conditional on receiving directly over the air and decoding a DSE Registered Location element with RegLoc DSE bit equal to 1, sent from an enabling STA. Before attempting enablement with any one enabling STA, a dependent STA may have detected several enabling STAs, may attempt enablement with one and fail, and then attempt enablement with another. An enabling STA shall assign dependent enablement identifiers in a way that makes them unique among STAs enabled by this enabling STA to help identify sources of interference.

An enabling STA may issue a DSE measurement request to any of its dependent STAs in order to receive a DSE measurement report that may have reported DSE LCI fields received from nearby STAs. The licensed operator may be able to use these reports to identify interfering STAs or map overlapping coverage in a multiple BSS situation. Because reported DSE LCI fields may refer to any destination address, a single DSE report may contain elements received from STAs that are not yet associated with a BSS.

An enabling STA may issue an ECS announcement to any of its dependent STAs in order to have their radio operation changed in frequency, channel bandwidth, and other operational parameters. A dependent AP or DFS owner with dot11DSERequired true and receiving an ECS announcement from its enabling STA shall perform the extended channel switch procedure as specified in 10.10.

An enabling STA may issue a DSE power constraint announcement to any of its dependent STAs in order to have their transmit power reduced from the regulatory limit. A dependent AP or DFS owner with dot11DSERequired true and receiving a DSE power constraint announcement from its enabling STA shall perform the power constraint procedure as specified in 10.12.5.

10.12.5 Dependent STA operation with DSE

Dependent STAs shall set dot11DSERequired to true.

A typical state machine implementation of dependent STA operation with DSE is provided in Figure 10-19.

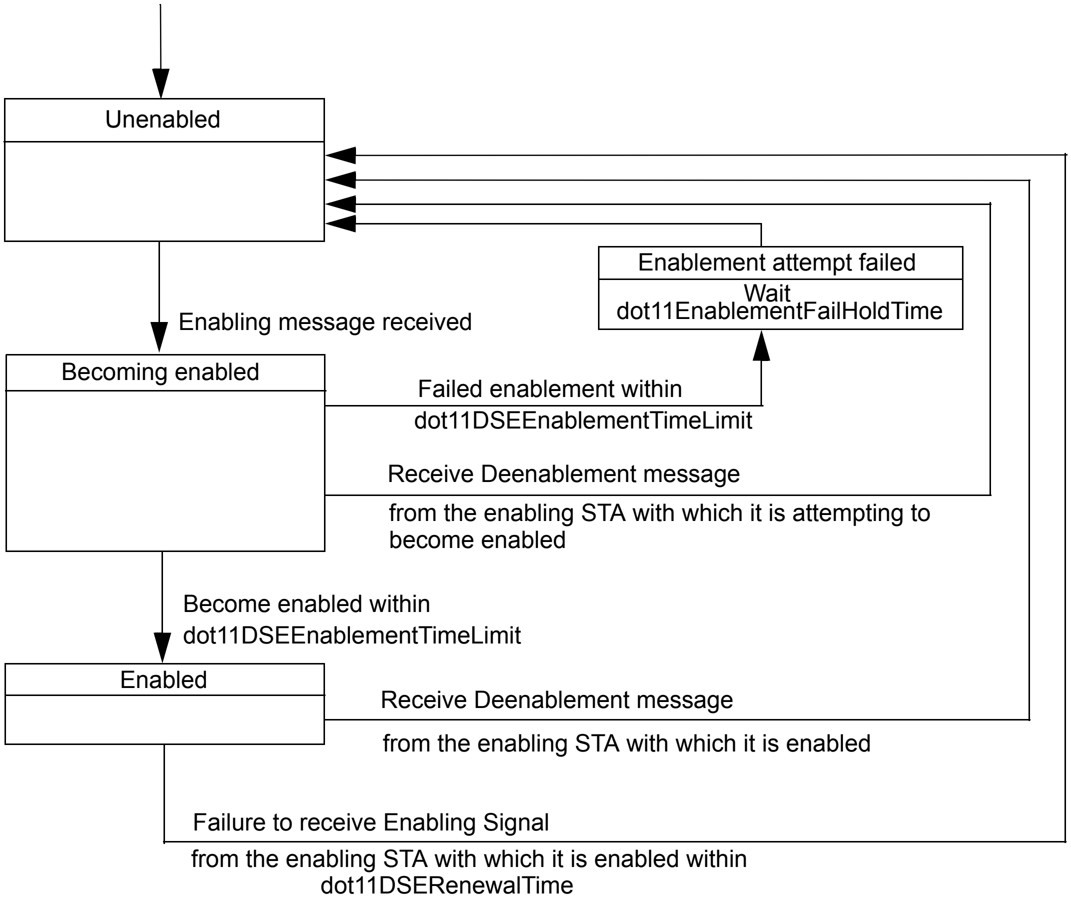

Unenabled
Enablement attempt failed
Wait
dot11EnablementFailHoldTime
Enabling message received
Becoming enabled
Failed enablement within
dot11DSEEnablementTimeLimit
Receive Deenablement message
from the enabling STA with which it is attempting to become enabled
Become enabled within
dot11DSEEnablementTimeLimit
Enabled
Receive Deenablement message
from the enabling STA with which it is enabled
Failure to receive Enabling Signal
from the enabling STA with which it is enabled within dot11DSERenewalTime

Figure 10-19—Dependent STA state machine

For DSE, the following statements apply:

- A STA with dot11DSERequired true shall not transmit any frames unless it has received a Beacon frame from an enabling STA with the Spectrum Management bit equal to 1 in the Capability Information field and with the RegLocDSE bit equal to 1 in the DSE Registered Location element. A dependent STA that is not enabled shall not transmit, except to attain enablement with an enabling STA, unless such action is mandated to be allowed in the regulatory domain (e.g., emergency services).
- A dependent STA shall not attempt enablement with an enabling STA unless the enabling STA is transmitting Beacon frames with the RegLoc DSE bit equal to 1 in the DSE Registered Location element.
- A dependent STA creates a dependent DSE Registered Location element containing the enabling STA's DSE Registered Location element and having the RegLoc DSE bit set to 0 and the Dependent STA bit set to 1. Before enablement, the Dependent Enablement Identifier field shall be set to 0. Upon attaining enablement, the Dependent Enablement Identifier field shall be set to the dependent enablement identifier value received by the dependent STA from the enabling STA in the MLME-ENABLEMENT.response primitive.
- The dependent STA shall send a Public Action DSE Registered Location Announcement frame to the broadcast address whenever the sum of dot11TransmittedFragmentCount, dot11GroupTransmittedFrameCount, and dot11ReceivedFragmentCount, modulo dot11DSETransmitDivider equals 0, possibly delayed by the completion of the sending of the frames of the current MSDU or MMPDU or by the completion of the current transmission opportunity (TXOP).
- A dependent STA that has not attained enablement shall not transmit beyond dot11DSEEnablement-TimeLimit (in seconds), measured from the time of the first PHY-TXSTART.request primitive, while attempting to attain enablement. Then if it is not enabled, it shall not transmit for dot11DSEEnablementFailHoldTime (in seconds), before it again attempts to attain enablement.
- A dependent STA receiving a DSE Deenablement frame with the RequesterSTAAddress field matching the enabling STA with which it last attempted enablement and the ResponderSTAAddress field matching its own IEEE MAC address shall change its enablement state with the enabling STA to unenabled and set all fields of its DSE Registered Location element body (see 8.4.2.54) to 0.
- A dependent STA shall return a DSE measurement report in response to a DSE measurement request if the request is received from the AP with which it is associated or the enabling STA with which it last attempted enablement and the ResponderSTAAddress field matches its own IEEE MAC address. The result may be the completed measurement or an indication that the STA is unable to complete the measurement request. A STA shall report it is too late to undertake a measurement request if it receives the request after the specified starting time for the measurement. The Measurement Report Mode field of a frame that is sent in response to a DSE Measurement Request frame shall not contain a value of 1 for the Incapable subfield and shall not contain a value of 1 for the Refused subfield of the Measurement Report Mode field of the DSE measurement report.
- A dependent STA receiving an ECS frame from the enabling STA with which it last attempted enablement or an ECS element from the AP with which it is associated shall perform the ECS procedure (see 10.10.3).
- A dependent STA receiving a DSE Power Constraint frame with the RequesterSTAAddress field matching the enabling STA with which it last attempted enablement and the ResponderSTAAddress field matching its own IEEE MAC address shall constrain its transmit power to be less than or equal to the maximum transmit power level specified for the current channel in the Country element minus the local power constraint specified in the DSE Power Constraint frame.
- An enabled dependent STA shall cease transmission within dot11DSERenewalTime (in seconds) if it has not received either a Beacon frame or a Probe Response frame from its enabling STA with the RegLoc DSE bit equal to 1 in the DSE Registered Location element. It shall then change its enablement state with the enabling STA to unenabled and set all fields of its DSE Registered Location element body (see 8.4.2.54) to 0.

10.13 Group addressed robust management frame procedures

When management frame protection is negotiated, the MLME shall provide an encapsulation service for group addressed robust management frames. All group addressed robust management frames shall be submitted to this service for encapsulation and transmission.

For group addressed management frames received in an MBSS for whom "Group Addressed Privacy" is indicated in Table 8-38, the group addressed frame protection service shall take the following actions:

— The frames shall be encapsulated and protected with the MGTK using the group cipher negotiated during the AMPE exchange.

For all other group addressed management frames, the group addressed frame protection service shall take the following actions:

— Management frame protection for multicast/broadcast shall be set using the MLME-SETPROTECTION.request primitive with the Protectlist including a Key Type value of IGTK. A non-AP STA shall also set the Protect Type value to Rx. In an IBSS, STAs shall set the ProtectType value to Rx_Tx. An AP shall set the Protect Type value to Tx.

— The IGTK shall be installed using the MLME-SETKEYS.request primitive with the value IGTK for the Key Type field in the Key Descriptor element.

— The frames shall be encapsulated and protected using BIP (see 11.4.4).

10.14 SA Query procedures

If dot11RSNAProtectedManagementFramesActivated is true, then the STA shall support the SA Query procedure.

To send an SA Query Request frame to a peer STA, the SME shall issue an MLME-SAQuery.request primitive. A STA that supports the SA Query procedure and receives an SA Query Request frame shall respond with an SA Query Response frame when all of the following are true: the receiving STA is currently associated to the sending STA, and no pending MLME-ASSOCIATE.request or MLME-REASSOCIATE.request primitives are outstanding for the STA that receives the SA Query indication.

If a non-AP STA that has an SA with its AP for an association that negotiated management frame protection receives an unprotected Deauthentication or Disassociation frame with reason code 6 or 7 from the AP, the non-AP STA may use this as an indication that there may be a mismatch in the association state between itself and the AP. In such a case, the non-AP STA may initiate the SA Query procedure with the AP to verify the validity of the SA by issuing one MLME-SAQuery.request primitive every dot11AssociationSAQueryRetryTimeout until a matching MLME-SAQuery.confirm primitive is received or dot11AssociationSAQueryMaximumTimeout TUs from the beginning of the SA Query procedure has passed. If the AP replies to the SA Query request with a valid SA Query response that has a matching transaction identifier, the non-AP STA may continue to use the SA. If no valid SA Query response is received, the non-AP STA may destroy the SA and move into State 1 with the AP.

10.15 20/40 MHz BSS operation

10.15.1 Rules for operation in 20/40 MHz BSS

The rules described in 10.15.1 through 10.15.12 are applicable to STAs that are either a STA 5G or a STA 2G4.

An FC STA shall support 20/40 BSS Coexistence Management.

NOTE—An FC HT STA that transmits a frame containing an Extended Capabilities element sets the 20/40 BSS Coexistence Management Support field of this element to 1.

An HT STA 2G4 that is a member of an IBSS and that transmits a frame containing an HT Operation element or Secondary Channel Offset element shall set the Secondary Channel Offset field of this element to SCN.

10.15.2 Basic 20/40 MHz BSS functionality

An HT AP declares its channel width capability (20 MHz only or 20/40 MHz) in the Supported Channel Width Set subfield of the HT Capabilities element.

An HT AP shall set the STA Channel Width field to 1 in frames in which it has set the Secondary Channel Offset field to SCA or SCB. An HT AP shall set the STA Channel Width field to 0 in frames in which it has set the Secondary Channel Offset field to SCN.

A non-AP HT STA declares its channel width capability (non-FC HT STA or FC HT STA) in the Supported Channel Width Set subfield in the HT Capabilities element.

NOTE 1—A 20/40 MHz BSS might include any mixture of FC HT STAs, non-FC HT STAs, and non-HT STAs. Protection requirements for mixed networks are defined in 9.23.

NOTE 2—A non-AP HT STA can switch between FC HT STA and non-FC HT STA operation by disassociation followed by association or reassociation.

An HT STA shall not indicate support for 40 MHz unless it supports reception and transmission of 40 MHz PPDUs using all MCSs within the BSSBasicMCSSet and all MCSs that are mandatory for the attached PHY.

An HT STA shall not transmit a 20 MHz PPDU containing one or more data MPDUs using the secondary channel of a 20/40 MHz BSSs. The Notify Channel Width frame may be used by a non-AP STA to notify another STA that the STA wishes to receive frames in the indicated channel width.

An HT STA that is a member of an IBSS adopts the value of the Secondary Channel Offset field in received frames according to the rules in 10.1.5 and shall not transmit a value for the Secondary Channel Offset field that differs from the most recently adopted value.

10.15.3 Channel selection methods for 20/40 MHz operation

10.15.3.1 General

For an HT STA, the following MIB attributes shall be set to true: dot11OperatingClassesImplemented, dot11OperatingClassesRequired, and dot11ExtendedChanneSwitchActivated.

An AP operating a 20/40 MHz BSS, on detecting an OBSS whose primary channel is the AP’s secondary channel, switches to 20 MHz BSS operation and may subsequently move to a different channel or pair of channels. A DFS owner (DO) STA operating a 20/40 MHz IBSS, on detecting an OBSS whose primary channel is the DO STA’s secondary channel, may choose to move to a different pair of channels.

NOTE—The setting up of a 40 MHz TDLS off-channel direct link is specified in 10.22.6.2.

10.15.3.2 Scanning requirements for a 20/40 MHz BSS

Before an AP or DO STA starts a 20/40 MHz BSS, it shall perform a minimum of dot11BSSWidthChannelTransitionDelayFactor OBSS scans (see 10.15.5) to search for existing BSSs.

If the AP or DO STA starts a 20/40 MHz BSS in the 5 GHz band and the BSS occupies the same two channels as any existing 20/40 MHz BSSs, then the AP or DO STA shall select a primary channel of the new BSS that is identical to the primary channel of the existing 20/40 MHz BSSs and a secondary channel of the new 20/40 MHz BSS that is identical to the secondary channel of the existing 20/40 MHz BSSs, unless the AP discovers that on these two channels are existing 20/40 MHz BSSs with different primary and secondary channels.

If an AP or DO STA starts a 20/40 MHz BSS in the 5 GHz band, the selected secondary channel should correspond to a channel on which no beacons are detected during the dot11BSSWidthChannelTransitionDelayFactor OBSS scan time performed by the AP or DO STA, unless there are beacons detected on both the selected primary and secondary channels.

NOTE—The 20/40 MHz channel sets and their corresponding behavior limits (i.e., choice of primary and secondary channels) permissible in each operating class are defined in Annex E and Annex D, respectively.

An HT AP or an DO STA that is also an HT STA should not start a 20 MHz BSS in the 5 GHz band on a channel that is the secondary channel of a 20/40 MHz BSS.

The AP or DO STA may continue to periodically scan after the BSS has been started. Information obtained during such scans is used as described within this subclause and within 10.15.2.

After starting a 20 MHz BSS, an FC HT AP 2G4 shall perform a minimum of dot11BSSWidthChannelTransitionDelayFactor OBSS scans, either by itself or through its associated STAs before making a transition from a 20 MHz BSS to a 20/40 MHz BSS. When the AP performs the scanning and the secondary channel for the 20/40 MHz BSS has been selected, then the scan shall be performed over the set of channels that are allowed operating channels within the current operational regulatory domain and whose center frequency falls within the *affected frequency range* given by Equation (10-1). When the AP performs the scanning without an intended secondary channel for the 20/40 MHz BSS or when the AP's associated STA(s) perform the scanning, then the scan shall be performed on all channels in the frequency band.

NOTE—An FC HT AP can change from operating a 20 MHz BSS to a 20/40 MHz BSS while maintaining associations by making a change to the transmitted value of the Secondary Channel Offset field.

$$\mathit{affected\ frequency\ range} = [\frac{f_P + f_S}{2} - 25\text{ MHz}, \frac{f_P + f_S}{2} + 25\text{ MHz}] \tag{10-1}$$

where
f_P is the center frequency of channel P
f_S is the center frequency of channel S

An FC HT AP 2G4 shall maintain a local boolean variable *20/40 Operation Permitted* that can have either the value true or false. The initial value of *20/40 Operation Permitted* shall be false. The value of *20/40 Operation Permitted* is recomputed according to Equation (10-2) whenever a BSS channel width trigger event is detected or whenever a period of time has elapsed with no BSS channel width triggers being detected and no overlap being reported, as defined in 10.15.12.

$$\begin{aligned}\mathit{20/40\ Operation\ Permitted} = {} & (P == OP_i \text{ for all values of } i) \text{ AND} \\ & (P == OT_i \text{ for all values of } i) \text{ AND} \\ & (S == OS_i \text{ for all values of } i)\end{aligned} \tag{10-2}$$

where
P is the operating or intended primary channel of the 20/40 MHz BSS
S is the operating or intended secondary channel of the 20/40 MHz BSS

OP_i is member *i* of the set of channels that are members of the channel set C and that are the primary operating channel of at least one 20/40 MHz BSS that was detected within the AP's BSA during the previous dot11BSSWidthChannelTransitionDelayFactor × dot11BSSWidthTriggerScanInterval seconds

OS_i is member *i* of the set of channels that are members of the channel set C and that are the secondary operating channel of at least one 20/40 MHz BSS that was detected within the AP's BSA during the previous dot11BSSWidthChannelTransitionDelayFactor × dot11BSSWidthTriggerScanInterval seconds

OT_i is member *i* of the set that comprises all channels that are members of the channel set C that were listed at least once in the Channel List fields of 20/40 BSS Intolerant Channel Report elements received during the previous dot11BSSWidthChannelTransitionDelayFactor × dot11BSSWidthTriggerScanInterval seconds and all channels that are members of the channel set C and that are the primary operating channel of at least one 20 MHz BSS that was detected within the AP's BSA during the previous dot11BSSWidthChannelTransitionDelayFactor × dot11BSSWidthTriggerScanInterval seconds

C is the set of all channels that are allowed operating channels within the current operational regulatory domain and whose center frequency falls within the *affected frequency range* given by Equation (10-1)

and where the use of "==" in the above expressions means that the value on the left side of the "==" is to be tested for equality with the value on the right side of the "==" yielding a boolean value of true if the two sides are equal and false if the two sides are unequal. If either side of the equality is the empty set or has a null value, then the expression is defined to have a boolean value of true.

An FC HT AP 2G4 shall not start a 20/40 MHz BSS in the 2.4 GHz band if the value of the local variable *20/40 Operation Permitted* is false (see Equation (10-2)).

An FC HT AP 2G4 may transmit a frame containing a Secondary Channel Offset field set to a value of SCA or SCB only if *20/40 Operation Permitted* is true.

In addition to information obtained by the FC HT AP 2G4 through its own scanning, an FC HT AP 2G4 shall use 20/40 BSS Intolerant Channel Report element information from received 20/40 BSS Coexistence Management frames with a value for the Address 1 field that matches the FC HT AP 2G4 using either individual or group addressing, but with no qualification of the Address 3 value, when determining if *20/40 Operation Permitted* is true or false. The information from the Channel List fields of received 20/40 BSS Intolerant Channel Report elements is used in generating the *OT* set for Equation (10-2).

After initial establishment of the 20/40 MHz BSS, if the value of *20/40 Operation Permitted* becomes false, the FC HT AP 2G4 reverts to 20 MHz BSS operation (see 10.15.12). The FC HT AP 2G4 might subsequently move the BSS to a pair of channels where the value of *20/40 Operation Permitted* evaluates to true.

10.15.3.3 Channel management at the AP and in an IBSS

While operating a 20/40 MHz BSS, an DO STA or an AP may decide to move its BSS, and an AP may decide to switch the BSS to 20 MHz operation either alone or in combination with a channel move. These channel move or BSS width switch operations might occur if, for example, another BSS starts to operate in either or both of the primary or secondary channels, or if radar is detected in either or both of the primary or secondary channels, or for other reasons that are beyond the scope of this standard. Specifically, the AP or DO STA may move its BSS to a different pair of channels, and the AP may separately, or in combination with the channel switch, change from a 20/40 MHz BSS to a 20 MHz BSS using either the primary channel of the previous channel pair or any other available 20 MHz channel. While operating a 20 MHz BSS, an DO STA or an AP may decide to move its BSS, and an AP may decide to switch the BSS to a 20/40 MHz BSS, either alone or in combination with a channel move.

If an AP or DO STA uses one or more Extended Channel Switch Announcement frames without also using Beacon and Probe Response frames to announce a change of operating class and/or a change in channel(s) and if the new operating class supports either of the behavior limits 13 or 14 as identified in the appropriate table of Annex E (i.e., Table E-1, Table E-2, or Table E-3), then the BSS width (20 MHz BSS or 20/40 MHz BSS) immediately after the switch shall be the same as the BSS width immediately before the transmission of the first Extended Channel Switch Announcement frame that announced the change. The AP or DO STA may subsequently perform a BSS width change.

NOTE—If an AP or DO STA uses one or more Extended Channel Switch Announcement frames without also using Beacon and Probe Response frames to announce a change of operating class and/or a change in channel(s), then the AP or DO STA cannot change from 20 MHz BSS operation to 20/40 MHz BSS operation as part of that change, even if the new operating class supports 20/40 MHz BSS operation, because Extended Channel Switch Announcement frames do not convey secondary channel information (i.e., information regarding whether a secondary channel, if permitted in the operating class, is to be used).

When switching a 20/40 MHz BSS to 20 MHz BSS mode, the AP may recalculate the TS bandwidth budget and may delete one or more active TSs by invoking the MLME-DELTS.request primitive with a ReasonCode value of SERVICE_CHANGE_PRECLUDES_TS.

An AP switches between 20/40 MHz BSS and 20 MHz BSS as follows:

- By changing the value of the Secondary Channel Offset field of the HT Operation element in the Beacon frame, and/or
- By changing the value of the Secondary Channel Offset field of the Secondary Channel Offset element, and/or
- Through the New Operating Class field of transmitted Extended Channel Switch Announcement elements.

In order to maintain existing associations and/or minimize disruption to communications with other STAs while making a channel width change or while performing a channel pair relocation, an AP may inform HT STAs within its BSS that it is making the change by including an Extended Channel Switch Announcement element in Beacon, Probe Response, and Extended Channel Switch Announcement frame transmissions until the intended channel switch time. An DO STA may inform HT STAs within its BSS that it is performing a channel pair relocation by including an Extended Channel Switch Announcement element in Beacon, Probe Response, and Extended Channel Switch Announcement frame transmissions until the intended channel switch time. The New Channel Number field of the Extended Channel Switch Announcement element represents the new channel (when the BSS after relocation/width change will be a 20 MHz BSS) or the primary channel of the new pair of channels (when the BSS after relocation/width change will be a 20/40 MHz BSS). When changing to a new pair of channels, the New Operating Class field specifies the position of the secondary channel relative to the new primary channel, i.e., either above or below.

When transmitting HT Operation elements, Channel Switch Announcement elements, and/or Extended Channel Switch Announcement elements, the AP moving the BSS or changing its channel width selects a combination of operating parameters from any single row of any one of the tables in Annex E that is appropriate for the current operating domain of the AP. Similarly, when transmitting HT Operation elements, Channel Switch Announcement elements, and/or Extended Channel Switch Announcement elements, the DO STA moving the BSS selects a combination of operating parameters from any single row of any one of the tables in Annex E that is appropriate for the current operating domain of the DO STA. The AP or DO STA selects one channel number from the “Channel set” column of the selected row. The AP or DO STA includes the selected information in subsequently transmitted frames that contain any combination of the following four elements:

- HT Operation element
- Channel Switch Announcement element
- Extended Channel Switch Announcement element

- Secondary Channel Offset element

The AP or DO STA shall set the Secondary Channel Offset field of transmitted HT Operation elements and transmitted Secondary Channel Offset elements to SCA if the Behavior Limit parameter of the selected row contains the value 13. The AP or DO STA shall set the Secondary Channel Offset field of transmitted HT Operation elements and transmitted Secondary Channel Offset elements to SCB if the Behavior Limit parameter of the selected row contains the value 14. The AP or DO STA shall set the Secondary Channel Offset field of transmitted HT Operation elements and transmitted Secondary Channel Offset elements to SCN if the Behavior Limit parameter of the selected row contains neither the value 13 nor the value 14.

The AP or DO STA shall set the New Channel Number field of transmitted Channel Switch Announcement elements and Extended Channel Switch Announcement elements to the value of the selected channel from the selected row.

The AP or DO STA shall set the New Operating Class field of transmitted Extended Channel Switch Announcement elements to the value of the "Operating class" column of the selected row.

Movement of a 20/40 MHz BSS from one channel pair to a different channel pair and changing between 20 MHz and 20/40 MHz operation should be scheduled so that all STAs in the BSS, including STAs in power save mode, have the opportunity to receive at least one Extended Channel Switch Announcement element or Channel Switch Announcement element before the switch.

When the Extended Channel Switch Announcement element and Extended Channel Switch Announcement frame are transmitted in bands where dot11SpectrumManagementRequired is true, the Channel Switch Announcement element and Channel Switch Announcement frame may also be transmitted. A STA that announces a channel switch using both the Extended Channel Switch Announcement element and the Channel Switch Announcement element shall set the New Channel Number field of both elements to the same value. An HT STA that receives a channel switch announcement through both the Extended Channel Switch Announcement element and the Channel Switch Announcement element shall ignore the received Channel Switch Announcement element.

For 20 MHz operation when the new operating class signifies 40 MHz channel spacing, the 20 MHz channel is the primary channel of the 40 MHz channel.

10.15.4 40 MHz PPDU transmission restrictions

10.15.4.1 Fields used to determine 40 MHz PPDU transmission restrictions

Several fields from various frames are used to convey information between STAs regarding the support for 40 MHz PPDU transmission and reception and regarding any current prohibition against the transmission and reception of 40 MHz PPDUs.

The rules defined in 10.15.4.2, 10.15.4.3, and 10.15.4.4 describe the behavior that accompanies those fields.

The fields that are used to determine the status of the transmission and reception of 40 MHz PPDUs are as follows:

- The Supported Channel Width Set subfield of the HT Capabilities element
- The Secondary Channel Offset field of the HT Operation element
- The STA Channel Width field of the HT Operation element
- The Channel Width field of the Notify Channel Width frame
- The Extended Channel Switch Announcement element

The Supported Channel Width Set subfield is used to indicate whether the transmitting STA is capable of transmitting and receiving 40 MHz PPDUs.

NOTE—The Supported Channel Width Set subfield transmitted by an AP is constant for the lifetime of its BSS as it is a parameter of the MLME-START.request primitive.

In addition to the restrictions on transmission of 40 MHz mask PPDUs found in 10.15.4.1 through 10.15.4.4, if a STA operating in the 2.4 GHz industrial, scientific, and medical (ISM) band has no means of determining the presence of non-IEEE 802.11 communication devices operating in the area, then the STA shall not transmit any 40 MHz mask PPDUs.

In addition to the restrictions on transmission of 40 MHz mask PPDUs found in 10.15.4.1 through 10.15.4.4, if a STA operating in the 2.4 GHz ISM band has a means of determining the presence of non-IEEE 802.11 communication devices operating in the area and determines either that no non-IEEE 802.11 communication device is operating in the area or that non-IEEE 802.11 communication devices are operating in the area but the STA implements a coexistence mechanism for these non-IEEE 802.11 communication devices, then the STA may transmit 40 MHz mask PPDUs; otherwise, the STA shall not transmit any 40 MHz mask PPDUs.

10.15.4.2 Infrastructure non-AP STA restrictions

A STA that is associated with an AP shall not transmit a value for the Supported Channel Width Set subfield that differs from a previously transmitted value during its current association.

The Secondary Channel Offset field is used to indicate whether the BSS is occupying a 40 MHz wide pair of channels and, when a secondary channel exists, whether it is above or below the primary channel in frequency. The Extended Channel Switch Announcement frame and the Extended Channel Switch Announcement element can each be used to indicate a transition from 20/40 MHz BSS operation to 20 MHz BSS operation and vice versa and to indicate whether a secondary channel, when it exists, is above or below the primary channel in frequency.

An FC HT STA shall maintain a local boolean variable *40MHzOperatingClass* as described here. The initial value of *40MHzOperatingClass* shall be false. The value of *40MHzOperatingClass* is recomputed according to the rules in this subclause at every TBTT and following the reception of a frame transmitted by the AP associated with the STA when that frame contains either of the following fields:

- Current Operating Class field
- New Operating Class field

The local boolean variable *40MHzOperatingClass* becomes true upon reception of a frame transmitted by the associated AP if the frame contained a Current Operating Class field with a value that corresponds to an operating class that corresponds to a channel spacing value of 40 MHz, as specified in Annex E.

The local boolean variable *40MHzOperatingClass* becomes false upon reception of a frame transmitted by the associated AP if the frame contained a Current Operating Class field with a value that corresponds to an operating class that does not correspond to a channel spacing value of 40 MHz.

The local boolean variable *40MHzOperatingClass* becomes true at the n^{th} TBTT following reception of a frame transmitted by the associated AP that contains an Extended Channel Switch Announcement element with a value of n in the Channel Switch Count field and a value in the New Operating Class field that corresponds to an operating class that corresponds to a channel spacing value of 40 MHz provided that the frame is the most recently received frame meeting the above conditions.

The local boolean variable *40MHzOperatingClass* becomes false at the n^{th} TBTT following reception of a frame transmitted by the associated AP that contains an Extended Channel Switch Announcement element with a value of n in the Channel Switch Count field and a value in the New Operating Class field that corresponds to an operating class that does not correspond to a channel spacing value of 40 MHz provided that the frame is the most recently received frame meeting the above conditions.

A STA can choose to dynamically constrain its operating channel width to 20 MHz while being a member of a 20/40 MHz BSS. Transitions to and from this constrained state are indicated using the transmission of a frame that carries the Channel Width field. A Channel Width field value of 0 indicates that the transmitting STA is not currently able to receive 40 MHz PPDUs, beginning at the end of the transmission of the frame that contained the Channel Width field.

A STA shall not transmit a frame containing a STA Channel Width field or a Channel Width field set to 1 if the value of its most recently transmitted Supported Channel Width Set subfield is 0.

A STA that is associated with an infrastructure BSS (STA1) shall not transmit a 40 MHz PPDU containing one or more frames addressed to another STA (STA2) unless the following three conditions are true:

- The Supported Channel Width Set subfield of the HT Capabilities element of both STAs is equal to 1
- The Secondary Channel Offset field of the most recently received HT Operation element sent by the AP of the BSS has a value of SCA or SCB
- The local boolean variable *40MHzOperatingClass* is true.

If the above three conditions are met, STA1 should not transmit a 40 MHz PPDU containing one or more frames addressed to STA2 unless the following two conditions are true:

- Either STA1 has not received a Notify Channel Width frame that was transmitted by STA2, or the Channel Width field of the most recently received Notify Channel Width frame at STA1 that was transmitted by STA2 is nonzero.
- If STA2 is an AP, the STA Channel Width field of the most recently received HT Operation element that was transmitted by STA2 is equal to 1.

10.15.4.3 AP restrictions

An AP shall not transmit a 40 MHz PPDU containing one or more frames addressed to another STA unless the following three conditions are true:

- The Supported Channel Width Set subfield of the HT Capabilities element of the AP and the STA are equal to a nonzero value.
- The Secondary Channel Offset field of the AP's most recently transmitted HT Operation element has a value of SCA or SCB
- The local boolean variable *40MHzOperatingClass* is true.

If the above three conditions are met, the AP should not transmit a 40 MHz PPDU containing frames addressed to another STA unless either the AP has not received a Notify Channel Width frame that was transmitted by the STA or the Channel Width field of the most recently received Notify Channel Width frame at the AP that was transmitted by the STA is nonzero.

An AP shall not transmit a 40 MHz PPDU containing one or more frames with a group address in the Address 1 field unless the following three conditions are true:

- The Supported Channel Width Set subfield of the HT Capabilities element of the AP is equal to 1
- The Secondary Channel Offset field of the AP's most recently transmitted HT Operation element has a value of SCA or SCB
- The local boolean variable *40MHzOperatingClass* is true.

If the above three conditions are met, the AP should not transmit a 40 MHz PPDU containing one or more frames with a group address in the Address 1 field if the most recently received Notify Channel Width frame for any of the STAs associated with the AP has the Channel Width field equal to 0.

10.15.4.4 Restrictions on non-AP STAs that are not infrastructure BSS members

An HT STA 2G4 that is not a member of an infrastructure BSS shall not transmit a 40 MHz mask PPDU.

An HT STA 5G that is not associated with an infrastructure BSS (STA1) shall not transmit a 40 MHz PPDU containing frames addressed to another STA (STA2) unless the following three conditions are true:

- The Supported Channel Width Set subfield of the HT Capabilities element of both STAs is equal to 1
- The Secondary Channel Offset field of the most recently received HT Operation element sent by STA2 has a value of SCA or SCB
- The Secondary Channel Offset field of the most recently transmitted HT Operation element sent by STA1 has a value of SCA or SCB

If the above three conditions are met, STA1 should not transmit a 40 MHz PPDU containing one or more frames addressed to STA2 unless STA1 has not received a STA Channel Width field that was transmitted by STA2 or the value of the most recently received STA Channel Width field at STA1 that was transmitted by STA2 is nonzero.

An HT STA 5G that is not associated with an infrastructure BSS (STA1) shall not transmit a 40 MHz PPDU containing one or more frames with a group address in the Address 1 field unless the following two conditions are true:

- The Supported Channel Width Set subfield of the HT Capabilities element most recently transmitted by STA1 is equal to 1.
- The Secondary Channel Offset field of the HT Operation element most recently transmitted by STA1 has a value of SCA or SCB.

If the above two conditions are met, STA1 should not transmit a 40 MHz PPDU containing one or more frames with a group address in the Address 1 field unless the most recently received STA Channel Width field for each other known member of the BSS of which STA1 is a member is equal to 1.

10.15.5 Scanning requirements for 40-MHz-capable STA

An OBSS scan operation is a passive or active scan of a set of channels that are potentially affected by 20/40 MHz BSS operation. Each channel in the set may be scanned more than once during a single OBSS scan operation. OBSS scans are performed by STAs that are FC HT STA 2G4. STAs that are FC HT STA 5G are not required to perform OBSS scan operations.

NOTE—STAs that perform OBSS scans report discovered BSSs and received 20/40 BSS coexistence information to their associated AP (see 10.15.12).

During an individual scan within an OBSS scan operation, the minimum per-channel scan duration is dot11OBSSScanPassiveDwell TU (when scanning passively) or dot11OBSSScanActiveDwell TU (when scanning actively). During an OBSS scan operation, each channel in the set is scanned at least once per dot11BSSWidthTriggerScanInterval seconds, and the minimum total scan time (i.e., the sum of the scan durations) per channel within a single OBSS scan operation is dot11OBSSScanPassiveTotalPerChannel TU for a passive scan and dot11OBSSScanActiveTotalPerChannel TU for an active scan.

NOTE—The values provided in the previous paragraph indicate the minimum requirements. For some combinations of parameter values, it is necessary to exceed the minimum values of some parameters in order to meet the minimum value constraints of all parameters.

When an AP transmits an Overlapping BSS Scan Parameters element, the value of each of the fields of the element shall be set to the value of the MIB attribute from the transmitting AP's MIB according to the mapping between the frame fields and MIB attributes as defined in 8.4.2.61.

Upon receipt of a frame containing an Overlapping BSS Scan Parameters element from the AP with which an FC HT STA 2G4 is associated, the MLME of the receiving FC HT STA 2G4 shall update each of the values of the MIB attributes used during OBSS scanning operations according to the mapping between the frame fields and MIB attributes as defined in 8.4.2.61.

An FC HT AP 2G4 may transmit frames containing an Overlapping BSS Scan Parameters element to any or all associated STAs in order to provide OBSS scan parameter values that are different from the default values.

An FC HT STA 2G4 that is associated with an FC HT AP 2G4 shall perform at least one OBSS scan every dot11BSSWidthTriggerScanInterval seconds, unless the FC HT STA 2G4 satisfies the conditions described in 10.15.6.

10.15.6 Exemption from OBSS scanning

An FC HT STA 2G4 shall maintain a local variable *ActivityFraction*. The value of *ActivityFraction* is defined by Equation (10-3).

$$ActivityFraction = \frac{T_{ACTIVE}}{T_{MEASURE\text{-}ACTIVE}} \tag{10-3}$$

where

T_{ACTIVE} is the total duration of transmitted MSDUs and received individually addressed MSDUs during the previous $T_{MEASURE\text{-}ACTIVE}$ seconds

$T_{MEASURE\text{-}ACTIVE}$ is dot11BSSWidthChannelTransitionDelayFactor × dot11BSSWidthTriggerScanInterval seconds.

An FC HT STA 2G4 may transmit to its associated AP a 20/40 BSS Coexistence Management frame with the Scanning Exemption Request field in the 20/40 Coexistence element set to 1.

If the last 20/40 BSS Coexistence Management frame received by an FC HT STA 2G4 in an individually addressed frame from its associated AP has the Scanning Exemption Grant field equal to 1, the STA is exempted from scanning whenever the value of its local variable *ActivityFraction* is less than dot11OBSSScanActivityThreshold/10000.

An FC HT AP 2G4 shall not transmit a 20/40 BSS Coexistence Management frame with the Scanning Exemption Grant field set to 1 addressed to an FC HT STA if the following condition is true:

— The FC HT STA has transmitted one or more channel report elements and is the only STA in the BSS that has indicated one or more channels on which a STA has found conditions that disallow the use of a 20/40 MHz BSS.

If there is more than one FC HT STA in the BSS that has indicated conditions that disallow the use of 20/40 MHz BSS on a specific channel, then the following apply:

— If all the FC HT STAs that have indicated unavailability of a channel have also requested to be exempt from scanning, the AP shall disallow at least one of the FC HT STA to be exempt from scanning.

— If, from the group of FC HT STAs that have indicated unavailability of a channel, there is at least one FC HT STA that has not requested to be exempt from scanning, the AP may allow all the STAs that have requested to be exempt from scanning to be exempted from scanning.

10.15.7 Communicating 20/40 BSS coexistence information

In addition to the 20/40 BSS Coexistence Management frame, a STA can include the 20/40 BSS Coexistence element in transmitted Beacon, Probe Request, Probe Response, (Re)Association Request, and (Re)Association Response frames.

10.15.8 Support of DSSS/CCK in 40 MHz

Transmission and reception of PPDUs using DSSS/CCK by FC HT STAs is managed using the DSSS/CCK Mode in 40 MHz subfield of the HT Capabilities Info field (see 8.4.2.58.2).

An HT STA declares its capability to use DSSS/CCK rates while it has a 40 MHz operating channel width through the DSSS/CCK Mode in 40 MHz subfield of its (Re)Association Request frames.

If the DSSS/CCK Mode in 40 MHz subfield is equal to 1 in Beacon and Probe Response frames, an associated HT STA in a 20/40 MHz BSS may generate DSSS/CCK transmissions. If the subfield is equal to 0, then the following apply:

— Associated HT STAs shall not generate DSSS/CCK transmissions.

— The AP shall not include an ERP element in its Beacon and Probe Response frames.

— The AP shall not include DSSS/CCK rates in the Supported Rates element.

— The AP shall refuse association requests from a STA that includes only DSSS/CCK rates in its Supported Rates and Extended Supported Rates elements.

10.15.9 STA CCA sensing in a 20/40 MHz BSS

A STA may transmit a 20 MHz mask PPDU in the primary channel following the rules in 9.19.2.

A STA transmitting a 40 MHz mask PPDU that begins a TXOP using EDCA as described in 9.19.2.3 or that is using a PIFS as permitted in 9.3.2.3.4 shall sense CCA on both the 20 MHz primary channel and the 20 MHz secondary channel before the 40 MHz mask PPDU transmission starts.

Unless explicitly stated otherwise, a STA may treat a PHY-CCA.indication primitive that is BUSY as though it were IDLE in the following cases:

— If the channel-list parameter is present and equal to {secondary} and the STA is transmitting a 20 MHz mask PPDU on the primary channel, or

— If the channel-list parameter is present and equal to {primary} and the STA is transmitting a 20 MHz mask PPDU on the secondary channel.

NOTE—Transmission of PPDUs on the secondary channel is also subject to constraints in 10.15.2.

At the specific slot boundaries (defined in 9.3.7) determined by the STA based on the 20 MHz primary channel CCA, when the transmission begins a TXOP using EDCA (as described in 9.19.2.3), the STA may transmit a pending 40 MHz mask PPDU only if the secondary channel has also been idle during the times the primary channel CCA is performed (defined in 9.3.7) during an interval of a PIFS for the 5 GHz band and DIFS for the 2.4 GHz band immediately preceding the expiration of the backoff counter. If a STA was unable to transmit a 40 MHz mask PPDU because the secondary channel was occupied during this interval, it may take one of the following steps:

a) Transmit a 20 MHz mask PPDU on the primary channel.

b) Restart the channel access attempt. In this case, the STA shall invoke the backoff procedure as specified in 9.19.2 as though the medium is busy as indicated by either physical or virtual CS and the backoff timer has a value of 0.

NOTE—As a result of this rule, the STA selects a new random number using the current value of CW[AC], and the retry counters are not updated.

When a TXOP is obtained for a 40 MHz PPDU, the STA may transmit 40 MHz PPDUs and/or 20 MHz PPDUs during the TXOP. When the TXOP is obtained by the exchange of 20 MHz PPDUs only in the primary channel, the STA shall not transmit 40 MHz PPDUs during the TXOP.

10.15.10 NAV assertion in 20/40 MHz BSS

An HT STA shall update its NAV using the Duration/ID field value in any frame received in a 20 MHz PPDU in the primary channel or received in a 40 MHz PPDU and that does not have an RA matching the STA's MAC address.

NOTE—A STA need not set its NAV in response to 20 MHz frames received on the secondary channel or any other channel that is not the primary channel, even if it is capable of receiving those frames.

10.15.11 Signaling 40 MHz intolerance

An HT STA 2G4 shall set the Forty MHz Intolerant field to 1 in transmitted HT Capabilities elements if dot11FortyMHzIntolerant is true; otherwise, the field shall be set to 0.

A STA 2G4 shall set the Forty MHz Intolerant field to 1 in transmitted 20/40 BSS Coexistence fields if dot11FortyMHzIntolerant is true; otherwise, the field shall be set to 0. A STA 2G4 that is not an HT STA 2G4 shall include a 20/40 BSS Coexistence element in management frames in which the element may be present if the STA has a MIB attribute dot11FortyMHzIntolerant and dot11FortyMHzIntolerant is true.

A STA 5G shall set the Forty MHz Intolerant field to 0 in transmitted HT Capabilities elements and 20/40 BSS Coexistence fields.

10.15.12 Switching between 40 MHz and 20 MHz

The following events are defined to be BSS channel width trigger events (TEs):

— **TE-A**: On any of the channels of the channel set defined in Clause 19, reception of a Beacon frame that does not contain an HT Capabilities element.

— **TE-B**: On any of the channels of the channel set defined in Clause 19, reception of a 20/40 BSS Coexistence Management, Beacon, Probe Request, or Probe Response frame that contains a value of 1 in a Forty MHz Intolerant field and that has the Address 1 field equal to the receiving STA's address or to a group address value, with no further addressing qualifications.

— **TE-C**: Reception of a 20/40 BSS Coexistence Management frame with the 20 MHz BSS Width Request field equal to 1 and with a value for the Address 1 field that matches the receiving STA using either individual or group addressing and with a value for the TA field that corresponds to the MAC address of a STA with which the receiver is associated.

— **TE-D**: Reception of a 20/40 BSS Coexistence Management frame containing at least one 20/40 BSS Intolerant Channel Report element with a nonzero length and with a value for the Address 1 field equal to the receiving STA's address or to a group address value, but with no qualification of the Address 3 value.

An FC HT AP 2G4 shall reevaluate the value of the local variable *20/40 Operation Permitted* (see 10.15.3.2) when either of the following events occurs:

— A BSS channel width trigger event TE-A is detected.

- A BSS channel width trigger event TE-D is detected.

An FC HT AP 2G4 may reevaluate the value of the local variable *20/40 Operation Permitted* (see 10.15.3.2) when either of the following situations occurs:

- No BSS channel width trigger events TE-A are detected for a period of time equal to dot11BSSWidthChannelTransitionDelayFactor × dot11BSSWidthTriggerScanInterval seconds.
- No BSS channel width trigger events TE-D are detected for a period of time equal to dot11BSSWidthChannelTransitionDelayFactor × dot11BSSWidthTriggerScanInterval seconds.

An FC HT AP 2G4 that detects either BSS channel width trigger event TE-B or TE-C or that determines that the value of its variable *20/40 Operation Permitted* has changed from true to false shall set the Secondary Channel Offset field to SCN in transmitted HT Operation elements beginning at the next DTIM or next TBTT if no DTIMs are transmitted to indicate that no secondary channel is present (i.e., that the BSS operating width is 20 MHz).

An FC HT AP 2G4 shall not set the Secondary Channel Offset field to a value of SCA or SCB in transmitted HT Operation elements unless the following two conditions have been met:

- A period of dot11BSSWidthChannelTransitionDelayFactor × dot11BSSWidthTriggerScanInterval seconds have elapsed during which no BSS channel width trigger events TE-B or TE-C are detected.
- The value of the local variable *20/40 Operation Permitted* (see 10.15.3.2) is true.

To request an update of the status of the 20 MHz BSS Width Request field, an FC HT AP 2G4 can transmit a 20/40 BSS Coexistence Management frame with a value of 1 in the Information Request field as described in 10.17.

An FC HT STA 2G4 that is associated with an FC HT AP 2G4 shall maintain a record of detected BSS channel width trigger events as follows:

- For each detected BSS channel width trigger event TE-A:
    - If a DSSS Parameter Set field is present in the received Beacon frame, the channel of the BSS channel width trigger event is the value of the Current Channel field of the DSSS Parameter Set field; otherwise, the channel of the BSS channel width trigger event is the channel on which the detecting STA received the Beacon frame.
    - If a Supported Operating Classes element is present in the received Beacon frame, the operating class of the BSS channel width trigger event is the value of the Current Operating Class field of the Supported Operating Classes element of the received Beacon frame; otherwise, the operating class of the BSS channel width trigger event is "unknown."
- For each detected BSS channel width trigger event TE-A of a unique combination of operating class and channel, the FC HT STA 2G4 shall maintain a record containing two variables:
    - The operating class of the BSS channel width trigger event
    - The channel of the BSS channel width trigger event

NOTE—If a BSS channel width trigger event TE-A is detected for an operating class and channel combination for which no record exists, the STA creates such a record.

If a BSS channel width trigger event TE-A is detected for an operating class and channel combination for which a record already exists, the information in that record shall be updated with the information determined from the new trigger event.

For all BSS channel width trigger events TE-B, the FC HT STA 2G4 shall maintain a single record containing an indication of whether one or more trigger events TE-B have been detected.

At the completion of an OBSS scan operation (i.e., at the end of the period of time equal to dot11BSSWidthTriggerScanInterval) or when it receives a 20/40 BSS Coexistence Management frame from its associated AP that contains a value of 1 in the Information Request field, an FC HT STA 2G4 that is associated with an FC HT AP 2G4 shall create a 20/40 BSS Coexistence Management frame by including a value of 0 for all fields of a 20/40 BSS Coexistence Management frame and then transferring information from the BSS channel width trigger event TE-A and TE-B records to the frame according to the following four steps:

- For each unique operating class that is stored in the set of BSS channel width trigger event TE-A records, the STA shall create a 20/40 BSS Intolerant Channel Report element for inclusion in the frame and include all of the unique channels associated with the operating class in the channel list of that element.
- The STA sets the Forty MHz Intolerant field of the 20/40 BSS Coexistence element based on the value of dot11FortyMHzIntolerant (see 10.15.11).
- The STA shall set to 1 the 20 MHz BSS Width Request field of the 20/40 BSS Coexistence element for inclusion in the frame if a record for BSS channel width trigger event TE-B exists and indicates that at least one trigger event TE-B has been detected.
- The STA may set to 1 the Information Request field.

Upon completion of these four steps, the FC HT STA 2G4 shall delete all records for trigger events TE-A and TE-B. Subsequently detected trigger events cause the creation of new records as necessary to be used in subsequently generated 20/40 BSS Coexistence Management frames. Following the record deletion, the FC HT STA 2G4 shall transmit to its associated FC HT AP 2G4 the 20/40 BSS Coexistence Management frame if any of the following conditions is true:

- At least one 20/40 BSS Intolerant Channel Report element with the Length field equal to a nonzero value is included.
- The Forty MHz Intolerant field is equal to 1.
- The 20 MHz BSS Width Request field is equal to 1.
- The Information Request field is equal to 1.
- The frame was created in response to the reception of an Information Request field that was equal to 1.

10.16 Phased coexistence operation (PCO)

10.16.1 General description of PCO

PCO is an optional coexistence mechanism in which a PCO active AP divides time into alternating 20 MHz and 40 MHz phases (see Figure 10-20). The PCO active AP reserves the 20 MHz primary channel and the 20 MHz secondary channel in turn to start the 40 MHz phase and resets the NAV in the 20 MHz channels in the opposite order to start the 20 MHz phase. Due to the protection of the 40 MHz period in both channels, it is tolerant of OBSSs on both 20 MHz halves of a 40 MHz channel.

A PCO active STA that does not know the current PCO phase shall transmit using a 20 MHz PPDU.

During the 40 MHz phase, a PCO active STA shall transmit data frames using a 40 MHz HT PPDU and control frames using a non-HT duplicate or a 40 MHz HT PPDU, with the following exceptions:

- Any CF-End frame shall be sent using only a 40 MHz HT PPDU.
- A PCO active AP may transmit 20 MHz group addressed frames as defined in 9.7.5.3.

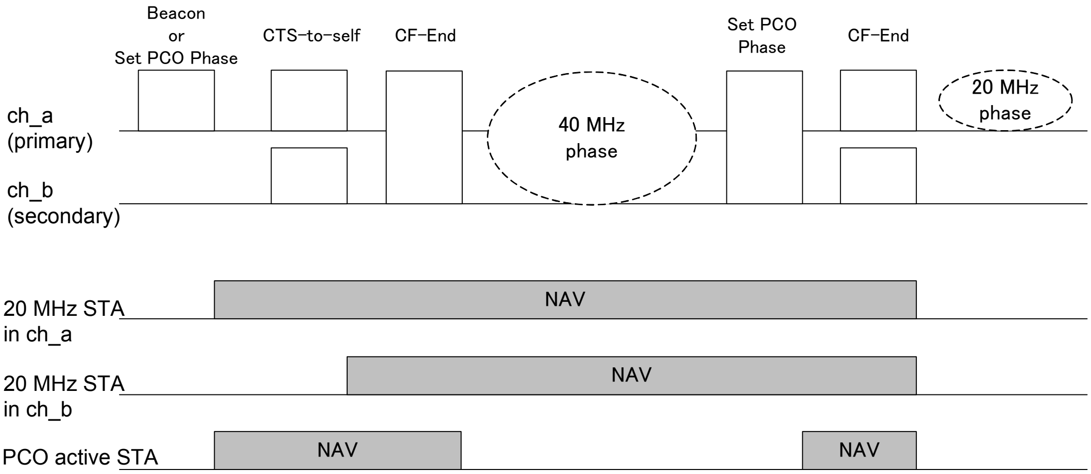


Figure 10-20—Phased coexistence operation (PCO)

A PCO active STA shall transmit management frames in 20 MHz or 40 MHz PPDUs according to 9.7.5 during the 40 MHz phase, except that Set PCO Phase frames shall be sent following the rules specified in 10.16.2.

During the 40 MHz phase, a PCO active STA may act as though the HT Protection field were equal to no protection mode, as defined in 9.23.3.1.

During the 20 MHz phase, a PCO active STA shall not transmit frames using a 40 MHz (HT or non-HT duplicate) PPDU. The protection of a PCO active STA during the 20 MHz phase is the same as protection in a 20 MHz BSS.

During the 20 MHz phase, a STA may transmit a 40 MHz mask PPDU that is not also a 40 MHz PPDU.

NOTE—This rule allows a STA to transmit 20 MHz PPDUs without requiring it to change to a 20 MHz transmit mask.

A PCO-capable AP may set the PCO Active field to 1 only if it is in a 20/40 MHz BSS.

NOTE—A non-PCO-capable 20/40 STA regards the PCO active BSS as a PCO inactive BSS. A non-PCO-capable 20/40 STA that associates with a PCO active BSS protects its transmissions as though the BSS were a PCO inactive BSS.

The value indicated by the PCO Transition Time field in the HT Extended Capabilities field is measured from the end of the PPDU carrying the Set PCO Phase frame. The PCO active STA shall be able to receive a PPDU using the new channel width no later than the value specified by the PCO Transition Time field after the end of the PPDU carrying the Set PCO Phase frame.

10.16.2 Operation at a PCO active AP

A PCO-capable AP activates PCO if it decides that PCO active BSS is more appropriate than either PCO inactive BSS or 20 MHz BSS in the current circumstances. The algorithm for making this decision is beyond the scope of this standard.[41]

A PCO active AP shall set the PCO Active field in the HT Operation element to 1.

[41] A PCO-capable AP might consider the performance impact, e.g., throughput and jitter, caused by and given to STAs based on their capabilities, traffic types, or load to determine the BSS's PCO mode. STAs under consideration might be not only associated STAs but also those that were detected in OBSSs.

When a PCO active AP detects that PCO is not providing a performance benefit, the PCO active AP may deactivate PCO and operate in either a PCO inactive BSS or 20 MHz BSS. A PCO-capable AP shall set the PCO Active field in the HT Operation element to 0 when PCO operation is disabled. Since the AP advertises the current mode in its Beacon and Probe Response frames, its associated STAs are informed of the mode change.

Values of the PCO Transition Time field in the HT Extended Capabilities field from 1 to 3 indicate the maximum time the PCO active STA takes to switch between a 20 MHz channel width and a 40 MHz channel width. A PCO active AP may set the PCO Transition Time field to 0 when it requires the associated PCO active STAs to be able to receive 40 MHz frames and respond with 40 MHz frames during the 20 MHz phase.

The PCO active AP shall increase the value of the PCO Transition Time field if the PCO active AP accepts the association of a PCO-capable STA whose value of the PCO Transition Time field exceeds the one currently used by the PCO active AP. If the PCO active AP decides not to extend its transition time to meet the value of the requesting STA, the PCO active AP shall deny the association. The AP may choose to continue PCO when a non-PCO-capable 20/40 STA requests association, and in such cases, the PCO active AP shall be able to receive 40 MHz frames and respond using 40 MHz frames during the 20 MHz phase.

A PCO active AP that indicates a switch to the 40 MHz phase by a PCO Phase field in a Beacon frame or by a PCO Phase Control field in a Set PCO Phase frame and that transmits a nonzero value of the PCO Transition Time field shall wait for at least the transition time specified by the PCO Transition Time field before sending a CF-End frame in the 40 MHz channel to start the 40 MHz operating phase.

When switching to the 40 MHz phase, a PCO active AP indicates a NAV duration either in the CF Parameter Set element of a Beacon frame or in the Duration/ID field of a Set PCO Phase frame sent on the primary channel that shall protect up to the end of the intended 40 MHz phase plus a transition time. A PCO active AP may continue the CFP after the 40 MHz phase by setting a longer duration for the CFP. The value of the Duration/ID field in a CTS-to-self frame sent to protect a 40 MHz phase shall be set to protect up to the intended end of the 40 MHz phase plus a transition time. The CTS-to-self shall be sent in a non-HT duplicate PPDU. The transmission of the CTS-to-self shall be delayed until the secondary channel CCA has indicated idle for at least a PIFS interval. It need not sense the primary channel because it is already reserved by a Beacon frame or a Set PCO Phase frame.

If the PCO Transition Time field is nonzero, a PCO active AP shall start a timer with a timeout value equal to the time specified by the PCO Transition Time field after transmitting a Beacon frame or a Set PCO Phase frame. If this timer expires while attempting to reserve the secondary channel, the AP shall transmit a Set PCO Phase frame indicating a switch back to the 20 MHz phase and shall transmit a CF-End frame on the primary channel.

NOTE—If this timer expires while attempting to reserve the secondary channel, the AP abandons switching to the 40 MHz phase to avoid an unexpectedly long delay.

A PCO active AP may transmit a Set PCO Phase frame in a non-HT duplicate PPDU followed by a CF-End frame in a 40 MHz HT PPDU to reserve both the primary and secondary channels again for the 40 MHz phase or to extend the 40 MHz phase. The value of the Duration/ID field in a Set PCO Phase frame contained in a non-HT duplicate PPDU for this intent shall protect up to the end of the intended 40 MHz phase plus the transition time.

To start the 20 MHz phase, a PCO active AP shall send a Set PCO Phase frame in a 40 MHz HT PPDU or in a non-HT duplicate PPDU with the Duration/ID field set to cover the transition time. It may also send a CF-End frame in both primary and secondary channels following the Set PCO Phase frame, where a CF-End frame in the primary channel shall be sent out at least after the transition time. The Duration/ID field of the Set PCO Phase frame for this case shall cover the transition time plus the duration of a CF-End frame.

A PCO active AP may broadcast a Set PCO Phase frame to advertise the current PCO phase to PCO active STAs.

Although PCO improves throughput in some circumstances, PCO might also introduce jitter. To minimize the jitter, the maximum duration of 40 MHz phase and 20 MHz phase is dot11PCOFortyMaxDuration and dot11PCOTwentyMaxDuration, respectively. Also in order for the PCO active AP to give opportunities for each STA to send frames, the minimum duration of 40 MHz phase and 20 MHz phase is dot11PCOFortyMinDuration and dot11PCOTwentyMinDuration, respectively.

10.16.3 Operation at a PCO active non-AP STA

If the PCO field in the Association Request frame to a PCO active AP is equal to 1 and the association succeeds, the STA shall operate in PCO mode. When requesting association, a PCO-capable STA shall set the PCO Transition Time field to 0 if the PCO active AP has set the PCO Transition Time field to 0. A PCO-capable STA may attempt to associate with a transition time that is larger than one currently advertised by the PCO active AP. If such an association fails, the PCO-capable non-AP STA may regard the BSS as a PCO inactive BSS and may attempt an association as a non-PCO-capable 20/40 STA.

NOTE—A STA that does not support the PCO transition time indicated by an AP might still attempt association with that AP. The AP will either refuse the association based on PCO transition time or respond by adjusting its PCO transition time to suit the STA.

A PCO active non-AP STA may transmit a Probe Request frame to the associated PCO active AP to determine the current PCO phase. A PCO active STA associated with a PCO active AP shall switch its operating phase from 20 MHz channel width to 40 MHz channel width when it receives from its AP a Beacon frame or a Probe Response frame that contains the PCO Phase field equal to 1 or a Set PCO Phase frame with the PCO Phase Control field equal to 1. The value of the CFP DurRemaining field in the CF Parameter Set element of a Beacon frame or the value of the Duration/ID field of a Set PCO Phase frame shall be interpreted as the duration of the PCO 40 MHz phase.

A PCO active STA associated with a PCO active AP shall switch its operating phase from 40 MHz channel width to 20 MHz channel width when it receives a Beacon frame or a Probe Response frame that contains the PCO Phase field equal to 0 or a Set PCO Phase frame with the PCO Phase Control field equal to 0. It also may switch from 40 MHz channel width to 20 MHz channel width based on the expiry of the value in the Duration/ID field of a Set PCO Phase frame that indicated a 40 MHz phase or based on the expiry of the value in the CFP DurRemaining field of the CF Parameter Set element of a Beacon frame that indicated a 40 MHz phase.

A PCO active STA shall halt PCO operation if it receives an HT Operation element from its AP with the PCO Active field equal to 0.

NOTE—An HT STA might change its PCO capabilities by disassociating followed by associating or reassociating with an AP.

10.17 20/40 BSS Coexistence Management frame usage

A STA that supports the 20/40 BSS Coexistence Management frame type shall set the 20/40 BSS Coexistence Management Support field to 1 in transmitted Extended Capabilities elements.

A STA that supports the 20/40 BSS Coexistence Management frame type shall include an Extended Capabilities element in transmitted Beacon, (Re)Association Request, (Re)Association Response, Probe Request, and Probe Response frames.

A STA shall not transmit to another STA a 20/40 BSS Coexistence Management frame with an individual address in the Address 1 field if the most recently received Extended Capabilities element from the recipient

STA contained a value of 0 in the 20/40 BSS Coexistence Management Support field. A STA that transmits a 20/40 BSS Coexistence Management frame may set the Address 1 field to a group address.

NOTE—A 20/40 BSS Coexistence Management frame is a class 1 frame and, therefore, can be sent to a STA that supports reception of such frames and that is not a member of the same BSS as the transmitting STA. In such a case, the BSSID of the frame is set to the wildcard BSSID value, regardless of whether the Address 1 field contains an individual or group address value.

A STA may transmit a 20/40 BSS Coexistence Management frame that contains a value of 1 for the Request Information field to another STA that supports the transmission of and reception of the 20/40 BSS Coexistence Management frame, except when the frame is a response to a 20/40 BSS Coexistence Management frame that contains a value of 1 for the Request Information field.

A STA that receives a 20/40 BSS Coexistence element with the Information Request field equal to 1, a value for the Address 1 field that matches the receiving STA using an individual address, and a nonwildcard BSSID field that matches the STA's BSS shall immediately queue for transmission a 20/40 BSS Coexistence Management frame with the transmitting STA as the recipient.

10.18 RSNA A-MSDU procedures

When dot11RSNAActivated is true, a STA indicates support for payload protected A-MSDUs (PP A-MSDUs) or signaling and payload protected A-MSDUs (SPP A-MSDUs) during association or reassociation. On either association or reassociation, the associating STA and its peer STA both determine and maintain a record of whether an encrypted A-MSDU sent to its peer is to be a PP A-MSDU or an SPP A-MSDU based on the value of the SPP A-MSDU Capable and SPP A-MSDU Required subfields of the RSN Capabilities field of the RSNE (see 8.4.2.27.4).

Table 10-9 defines behavior related to the transmission and reception of individually addressed A-MSDUs of a first HT STA (STA1) that has successfully negotiated an RSNA (re)association with a second HT STA (STA2). Reception and transmission of A-MSDUs using a non-RSN association is unaffected by the values of the SPP A-MSDU Capable and SPP A-MSDU Required subfields.

NOTE—This subclause does not describe the operation of group addressed A-MSDUs because the use of group addressed A-MSDUs is not permitted, as defined in 9.11.

10.19 Public Action frame addressing

A STA that is a member of a BSS that transmits a Public Action frame with an individual value in the Address 1 field corresponding to a STA that is not a member of the same BSS as the transmitting STA shall set the BSSID field of the frame to the wildcard BSSID value.

A STA that is a member of a BSS that transmits a Public Action frame to a group address shall set the BSSID field of the frame to the wildcard BSSID value or to the transmitting STA's BSSID value.

A STA that is a member of a BSS that transmits a Public Action frame with an individual value in the Address 1 field corresponding to a STA that is a member of the same BSS as the transmitting STA shall set the BSSID field of the frame to the transmitting STA's BSSID value.

A STA that is not a member of a BSS that transmits a Public Action frame shall set the BSSID field of the frame to the wildcard BSSID value.

10.20 STAs communicating data frames outside the context of a BSS

When dot11OCBActivated is true in a STA:

Table 10-9—A-MSDU STA behavior for RSN associations

STA1 state		STA2 state		STA1 action with respect to STA2
SPP A-MSDU capable	SPP A-MSDU required	SPP A-MSDU capable	SPP A-MSDU required	
0	0	X	0	May transmit PP A-MSDU. Shall not transmit SPP A-MSDU. Shall receive PP A-MSDU. Received SPP A-MSDU MIC fails.
0	0	X	1	Shall not transmit PP A-MSDU. Shall not transmit SPP A-MSDU. Shall discard received (PP and SPP) A-MSDU.
0	1	X	X	Shall not transmit PP A-MSDU. Shall not transmit SPP A-MSDU. Shall discard received (PP and SPP) A-MSDU.
1	0	0	0	May transmit PP A-MSDU. Shall not transmit SPP A-MSDU. Shall receive PP A-MSDU. Received SPP A-MSDU MIC fails.
1	0	0	1	Shall not transmit PP A-MSDU. Shall not transmit SPP A-MSDU. Shall discard received (PP and SPP) A-MSDU.
1	X	1	X	Shall not transmit PP A-MSDU. May transmit SPP A-MSDU. Received PP A-MSDU MIC fails. Shall receive SPP A-MSDU.
1	1	0	X	Shall not transmit PP A-MSDU. Shall not transmit SPP A-MSDU. Shall discard received (PP and SPP) A-MSDU.
NOTE—X = Not significant.				

a) Synchronization, authentication, association, and frame classes as defined in 10.1 and 10.3 are not used. Data confidentiality as defined in Clause 11 is not used. The STA may send management frames of subtype Action and, if the STA maintains a TSF Timer, subtype Timing Advertisement.

b) The STA may send control frames, except those of subtype PS-Poll, CF-End, and CF-End + CF-Ack.

c) The STA may send data frames of subtype Data, Null, QoS Data, and QoS Null.

d) The STA shall set the BSSID field in all management and data frames to the wildcard BSSID value.

When a STA joins a BSS, it shall set dot11OCBActivated to false. The STA shall keep dot11OCBActivated false while joined with the BSS or while the STA is the AP within a BSS. If a STA does not include the dot11OCBActivated MIB attribute, the STA shall operate as if the attribute is false.

When a mesh STA starts an MBSS or becomes a member of an MBSS, it shall set dot11OCBActivated to false. The STA shall keep dot11OCBActivated false as long as it provides the mesh facility.

Whenever MAC and PHY sublayer parameters are changed in a STA in which dot11OCBActivated is true, MAC and PHY sublayer operation shall resume with the appropriate MIB attributes in less than 2 TU.

A STA shall use information from the CF Parameter Set element of all received Beacon frames, without regard for the BSSID, to update its NAV as specified in 9.4.3.3.

10.21 Timing Advertisement

10.21.1 Introduction

A STA that sends a Timing Advertisement frame shall maintain a TSF Timer in order to set the Timestamp field in this frame. When a STA transmits the Timing Advertisement, Probe Response, or Beacon frame, the Timestamp shall be set to the value of the STA's TSF timer at the time that the data symbol containing the first bit of the Timestamp is transmitted to the PHY plus the transmitting STA's delays through its local PHY from the MAC-PHY interface to its interface with the WM [e.g., antenna, light emitting diode (LED) emission surface].

A STA may advertise a time standard by transmitting a Timing Advertisement element in one of the following frames: Timing Advertisement, Probe Response, or Beacon. As defined in 8.4.2.63 the Time Advertisement element contains two estimates. The Time Value field contains an estimate of the difference between a time standard and the timestamp included in the same frame. The Time Error field contains an estimate of the standard deviation of the error in the estimate in the Time Value field. The time standard might be derived from an external time source. A STA with an external time source might implement an estimator in a variety of ways, which are beyond the scope of this standard.

10.21.2 Timing advertisement frame procedures

The SME provides the Time Advertisement element to the MLME when it requests the MLME to send a Timing Advertisement frame. When a Timing Advertisement frame is received by a STA, its MLME reports the Timestamp, Local Time, Time Advertisement element, and estimates of propagation delay to the SME. For a STA that maintains a TSF Timer and receives a Timing Advertisement frame, Local Time is the value of the STA's TSF timer at the start of reception of the first octet of the Timestamp field of the frame. Otherwise, the Local Time is unspecified. The receiving STA's SME might use the Timestamp, Local Time, and Time Advertisement element to align its estimate of the time standard to the transmitting STA's estimate of the corresponding time standard.

10.21.3 UTC TSF Offset procedures

When dot11MgmtOptionUTCTSFOffsetActivated is true, the Time Advertisement and Time Zone elements shall be included in all Probe Response frames, and the Time Advertisement element shall be included in the Beacon frame every dot11TimeAdvertisementDTIMInterval DTIMs. When the dot11MgmtOptionUTCTSFOffsetActivated is false, the Time Advertisement and Time Zone elements shall not be included in Beacon and Probe Response frames.

The AP should periodically synchronize to a UTC reference clock[B48] so that the UTC TSF offset can account for drift. The AP shall increment the Time Update Counter field value in the Time Advertisement element each time the synchronization occurs. The method the AP uses to synchronize with a UTC reference clock is out of scope of the standard.

10.22 Tunneled direct-link setup

10.22.1 General

Tunneled direct-link setup (TDLS) is characterized by encapsulating setup frames in Data frames, which allows them to be transmitted through an AP transparently. Therefore, the AP does not need to be direct-link capable, nor does it have to support the same set of capabilities that are used on the direct link between the two TDLS peer STAs. TDLS also includes power saving, in the form of TDLS Peer PSM (scheduled) and TDLS Peer U-APSD (unscheduled). STAs that set up a TDLS direct link remain associated with their BSS, but have the option of transmitting frames directly to the other TDLS peer STA.

Transmitting a TDLS frame through the AP means that the frame's RA is set to the BSSID. Transmitting a frame over the direct path means that the frame's RA is set to the MAC address of the TDLS peer STA.

To set up and maintain a direct link, both TDLS peer STAs shall be associated with the same infrastructure BSS.

A TDLS peer STA may be involved in direct links with multiple TDLS peer STAs at the same time. Simultaneous operation of DLS and TDLS between the same pair of STAs is not allowed. A DLS Request frame shall not be transmitted to a STA with which a TDLS direct link is currently active. A DLS Request frame received from a STA with which a TDLS direct link is currently active shall be discarded.

The channel on which the AP operates is referred to as the base channel. If the AP operates in a 40 MHz channel, then the base channel refers to the primary channel. If the direct link is switched to a channel that is not the base channel, then this channel is referred to as the off-channel.

Features that are not supported by the BSS but that are supported by both TDLS peer STAs may be used on a TDLS direct link between those STAs, except PCO. An example is the use of an HT MCS on a TDLS direct link between HT STAs when these STAs are associated with a non-HT BSS. Features that are supported by the BSS shall follow the BSS rules when they are used on a TDLS direct link on the base channel. The channel width of the TDLS direct link on the base channel shall not exceed the channel width of the BSS to which the TDLS peer STAs are associated.

When admission control is required for an AC on the base channel, then the TDLS peer STA that intends to use this AC for direct-link transmissions on the base channel is responsible for setting up a TS with Direction of 'Direct Link' with the AP, as defined in 9.19.4.2.

A non-AP STA may act as TDLS initiator STA or TDLS responder STA when dot11TunneledDirectLinkSetupImplemented is true.

TDLS frames shall use the formatting as specified in 10.22.2 when they are transmitted through the AP and when they are transmitted over the TDLS direct link. A STA shall not transmit a TDLS Action field in a frame with the Type field of the frame set to Management. A received TDLS Action field in a frame with the Type field equal to Management shall be discarded. Note that the TDLS Discovery Response frame is not a TDLS frame but a Public Action frame.

TDLS shall not be used in an IBSS.

TDLS shall not be used in an MBSS.

Security is only available on the TDLS direct link when both TDLS peer STAs have an RSNA with the BSS.

TDLS shall not be used when the TDLS Prohibited subfield included in the Extended Capability element of the Association Response frame or Reassociation Response frame that led to the current association is equal to 1.

The HT Operation element shall be present in a TDLS Setup Confirm frame when both STAs are HT capable but the BSS is not.

10.22.2 TDLS payload

TDLS uses Ethertype 89-0d frames, as defined in Annex H. The TDLS payload contains a TDLS Action field as is specified in 8.5.13. The UP shall be AC_VI, unless otherwise specified.

10.22.3 TDLS Discovery

To discover TDLS capable STAs in the same BSS, a TDLS initiator STA may send a TDLS Discovery Request frame to an individual DA, through the AP. The TDLS responder STA Address field contained in the Link Identifier element of the TDLS Discovery Request frame shall be set to the Destination Address of the TDLS Discovery Request frame. A TDLS capable STA that receives a TDLS Discovery Request frame with a matching BSSID in the Link Identifier element shall send a TDLS Discovery Response frame to the requesting STA, via the direct path. The TDLS responder STA Address field contained in the Link Identifier element of the TDLS Discovery Response frame shall be set to the MAC address of the STA sending the TDLS Discovery Response frame. A TDLS Discovery Request frame shall not be sent within dot11TDLSDiscoveryRequestWindow DTIM intervals after transmitting TDLS Discovery Request frame.

A TDLS STA may send an individually addressed TDLS Discovery Response frame via the direct path without prior reception of a TDLS Discovery Request frame. A TDLS STA that receives such an unsolicited TDLS Discovery Response frame may respond with an individually addressed TDLS Discovery Response frame.

A TDLS Discovery Request frame shall not be sent to a group address. A TDLS Discovery Response frame shall not be sent to a group address.

A TDLS STA may also send a TDLS Setup Request frame to a STA in the same BSS to discover whether the TDLS peer STA is TDLS capable or not. A TDLS Setup Response frame transmitted in response to TDLS Setup Request frame indicates that the TDLS peer STA sending the TDLS Setup Response is TDLS capable.

An alternative mechanism to discover TDLS capable STAs in the same BSS, is provided by the TDLS Capability ANQP-element, as described in 10.24.3.2.10. This mechanism allows the ANQP request/response frames to use the direct path between the peer STAs.

10.22.4 TDLS direct-link establishment

To establish a TDLS direct link, the TDLS initiator STA shall send a TDLS Setup Request frame to the intended TDLS responder STA.

TDLS Setup Request frames, TDLS Setup Response frames, and TDLS Setup Confirm frames shall be transmitted through the AP and shall not be transmitted to a group address.

Upon receipt of a TDLS Setup Request frame, the following options exist at the TDLS responder STA:

a) The TDLS responder STA accepts the TDLS Setup Request frame, in which case the TDLS responder STA shall respond with a TDLS Setup Response frame with status code 0 (“Successful”).

b) The TDLS responder STA declines the TDLS Setup Request frame, in which case the TDLS responder STA shall respond with a TDLS Setup Response frame with status code 37 (“The request has been declined”). A TDLS setup request shall be declined when the BSSID in the received Link Identifier does not match the BSSID of the TDLS responder STA.

c) The TDLS Setup Request frame is received after sending a TDLS Setup Request frame and before receiving the corresponding TDLS Setup Response frame, and the source address of the received TDLS Setup Request frame is higher than its own MAC address, in which case the TDLS responder

STA shall discard the message and the TDLS responder STA shall send no TDLS Setup Response frame.

d) The TDLS Setup Request frame is received after sending a TDLS Setup Request frame and before receiving the corresponding TDLS Setup Response frame, and the source address of the received TDLS Setup Request frame is lower than its own MAC address. In this case, the TDLS responder STA shall terminate the TDLS setup it initiated. The TDLS responder STA shall send a response according to item a) or item b) above in this case.

e) If a TDLS Setup Request frame is received from a TDLS responder STA with which a currently active TDLS session exists, then the receiving STA shall tear down the existing TDLS direct link as if a TDLS Teardown frame was received, and respond with a TDLS Setup Response frame.

If no TDLS Setup Response frame is received within dot11TDLSResponseTimeout, or if a TDLS Setup Response frame is received with a nonzero status code, the TDLS initiator STA shall terminate the setup procedure and discard the TDLS Setup Response frame. Otherwise, the TDLS initiator STA shall send a TDLS Setup Confirm frame to the TDLS responder STA to confirm the receipt of the TDLS Setup Response frame.

When the BSS does not support EDCA, EDCA may be used on the direct link (on the base channel and on the off-channel), with the default EDCA Parameter Set, per 9.2.4.2.

If the STA has security enabled on the link with the AP, then the TPK Handshake messages are included in the TDLS Setup messages, as follows:

— TPK Handshake Message 1 shall be included in the TDLS Setup Request frame.
— TPK Handshake Message 2 shall be included in the TDLS Setup Response frame.
— TPK Handshake Message 3 shall be included in the TDLS Setup Confirm frame.

When the TDLS Setup Handshake has been completed, the TDLS initiator STA and the TDLS responder STA are TDLS peer STAs. A TDLS peer STA shall accept data frames received from the respective TDLS peer STA directly and Data frames destined for the respective TDLS peer STA may be transmitted over the direct link.

Subsequent to the successful completion of the TPK Handshake, all frames transmitted and received on the TDLS direct link shall be protected using the TPKSA, per the procedures defined in Clause 11.

A TDLS Setup Request frame received at a STA that does not support TDLS shall be ignored.

To avoid possible reordering of MSDUs, a TDLS initiator STA shall cease transmitting MSDUs to the TDLS responder STA through the AP after sending a TDLS Setup Request frame, and a TDLS responder STA shall cease transmitting MSDUs to the TDLS initiator STA through the AP after sending a TDLS Setup Response frame indicating status code 0 (Success).

The TDLS Setup Request frame and the TDLS Setup Response frame shall be transmitted using the lowest AC that was used for transmitting MSDUs to the respective TDLS peer STA during the previous dot11TDLSACDeterminationInterval seconds, or at AC_BK. When no MSDUs were transmitted during the previous dot11TDLSACDeterminationInterval seconds, then the TDLS Setup Request frame and the TDLS Setup Response frame may be sent at any AC, subject to applicable Admission Control rules.

If no TDLS Setup Response frame is received within dot11TDLSResponseTimeout, or if a TDLS Setup Response frame is received with status code other than 0 (“Success”), the TDLS initiator STA may resume transmitting MSDUs to the TDLS responder STA through the AP.

If a TDLS Setup Confirm frame is transmitted with a status code other than 0 (“Success”), the TDLS initiator STA may resume transmitting MSDUs to the TDLS responder STA through the AP.

If a TDLS Setup Confirm frame is received with a status code other than 0 (“Success”), the TDLS responder STA may resume transmitting MSDUs to the TDLS initiator STA through the AP.

A TDLS peer STA shall not transmit MSDUs over the direct link before transmitting or receiving a TDLS Setup Confirm frame with status code 0 (“Success”).

10.22.5 TDLS direct-link teardown

To tear down a direct link, a TDLS peer STA shall send a TDLS Teardown frame to the respective TDLS peer STA. A TDLS peer STA shall disable the direct link and destroy the related security parameters after successfully transmitting or receiving a TDLS Teardown frame. If the STA has security enabled on the link with the AP, then the FTE shall be included in the TDLS Teardown frame.

The TDLS Teardown frame shall be sent over the direct path and the reason code shall be set to “TDLS direct-link teardown for unspecified reason,” except when the TDLS peer STA is unreachable via the TDLS direct link, in which case, the TDLS Teardown frame shall be sent through the AP and the reason code shall be set to “TDLS direct-link teardown due to TDLS peer STA unreachable via the TDLS direct link.” If the direct link is on an off-channel when this condition occurs, then the TDLS peer STA may switch back to the base channel without initiating a channel switch frame exchange, before transmitting the TDLS Teardown frame.

If present, the contents of the FTE in the TDLS Teardown frame shall be the same as that included in the TPK Handshake Message 3 with the exception of the MIC field. The MIC shall be calculated on the concatenation, in the following order, of:

- Link Identifier element
- Reason Code
- Dialog token that was used in the MIC calculation for TPK Handshake Message 3
- Transaction Sequence number (1 octet) which shall be set to the value 4
- FTE, with the MIC field of the FTE set to 0

The MIC shall be calculated using the TPK-KCK and the AES-128-CMAC algorithm. The output of the AES-128-CMAC shall be 128 bits.

If the TPK Handshake was successful for this TDLS session, then a receiving STA shall validate the MIC in the TDLS Teardown frame prior to processing the TDLS Teardown frame. If MIC validation fails, the receiver shall ignore the TDLS Teardown frame.

When a TDLS direct link gets torn down, any related TSs shall be deleted by the TDLS peer STAs.

A TDLS Teardown frame with Reason Code 3 (“Deauthenticated because sending STA is leaving (or has left) IBSS or ESS”) shall be transmitted to all TDLS peer STAs (via the AP or via the direct path) prior to transmitting a Disassociation frame or a Deauthentication frame to the AP. After receiving a Deauthentication frame or a Disassociation frame from the AP, a Deauthentication frame with Reason Code 3 (“Deauthenticated because sending STA is leaving (or has left) IBSS or ESS”) shall be transmitted via the direct path to all TDLS peer STAs that are in the wake state, if robust management frame protection has not been negotiated on the TDLS direct link.

10.22.6 TDLS channel switching

When a STA enables support for TDLS channel switching, it shall set dot11TDLSChannelSwitching-Activated, dot11MultiDomainCapabilityActivated and dot11ExtendedChannelSwitchActivated to true. When TDLS channel switching is enabled, the STA may set TDLS Channel Switching capability field to 1. The STA shall include a Supported Channels element and a Supported Operating Classes element in all TDLS Setup Request and TDLS Setup Response frames that have a TDLS Channel Switching capability field equal to 1. The STA shall include only channels in the Supported Channels element for which it can adhere to the local power constraint. A channel switch shall not be initiated by a STA when the TDLS peer STA did not set the TDLS Channel Switching capability field to 1 in the transmitted TDLS Setup Request frame or the TDLS Setup Response frame that caused the TDLS direct link to be set up.

TDLS Channel Switch Request frames and TDLS Channel Switch Response frames shall be transmitted over the TDLS direct link.

TDLS channel switching is different from (I)BSS channel switching as defined in 10.9.8.

The channel on which the AP operates is referred to as the base channel. If the AP operates in a 40 MHz channel, then the base channel refers to the primary channel. If the direct link is switched to a channel that is not the base channel, then this channel is referred to as the off-channel.

The target channel is the destination channel of an intended channel switch. The target channel is specified by the STA that initiates a channel switch, from the set of operating classes supported by both TDLS peer STAs. The target channel and operating class are specified in the TDLS Channel Switch Request frame. The Country and Coverage Class settings on the target channel are the same as in the BSS to which both TDLS peer STAs are currently associated. Both STAs are entitled to request a channel switch. The events occurring for a channel switch are illustrated in Figure 10-21.

In Figure 10-21, the TDLS peer STAs (STA1 and STA2) are operating on an initial channel. After contending for the medium, STA1 transmits a TDLS Channel Switch Request frame to STA2, via the direct link, indicating a requested switch to a target channel. STA2 transmits an ACK frame (denoted ACK1 in Figure 10-21) after SIFS, and processes the TDLS Channel Switch Request frame. After contending for the medium, STA2 transmits a TDLS Channel Switch Response frame to STA1, possibly also after entering power save mode with the AP. STA1 responds with an ACK frame (denoted ACK2 in Figure 10-21) after SIFS. If the TDLS Channel Switch Response frame indicated with status code 37 (“The request has been declined”), then both STAs continue to operate on the current channel. If the TDLS Channel Switch Response frame indicated with status code 0 (“Successful”), then both STAs shall be listening on the target channel not later than SwitchTime after the end of the last symbol of ACK2, as measured at the air interface. After switching channels, each TDLS peer STA shall perform a clear channel assessment (CCA) on the target channel, until a frame sequence is detected by which it can correctly set its NAV, or until a period of time equal to at least dot11TDLSProbeDelay has transpired (this combined event is indicated as “ProbeTime” in Figure 10-21). The first transmission on the target channel shall be preceded by a random backoff, which shall start at the end of the ProbeTime. The first transmission on the new channel shall not start before the end of SwitchTime. The initiator of the channel switch shall transmit a Data frame on the target channel, unless the SwitchTimeout has expired or the responder to the channel switch transmitted a Data frame on the target channel.

If no successful frame exchange has occurred on an off-channel within SwitchTimeout after the end of the last symbol of ACK2, as measured at the air interface, the STAs shall go back to the base channel, where they shall be listening not later than SwitchTime after the end of the SwitchTimeout. After changing channels (either from the base channel to the off-channel or from the off-channel to the base channel), a TDLS peer STA shall perform CCA until a frame sequence is detected by which it can correctly set its NAV, or until a period of time equal to the ProbeTime has transpired.

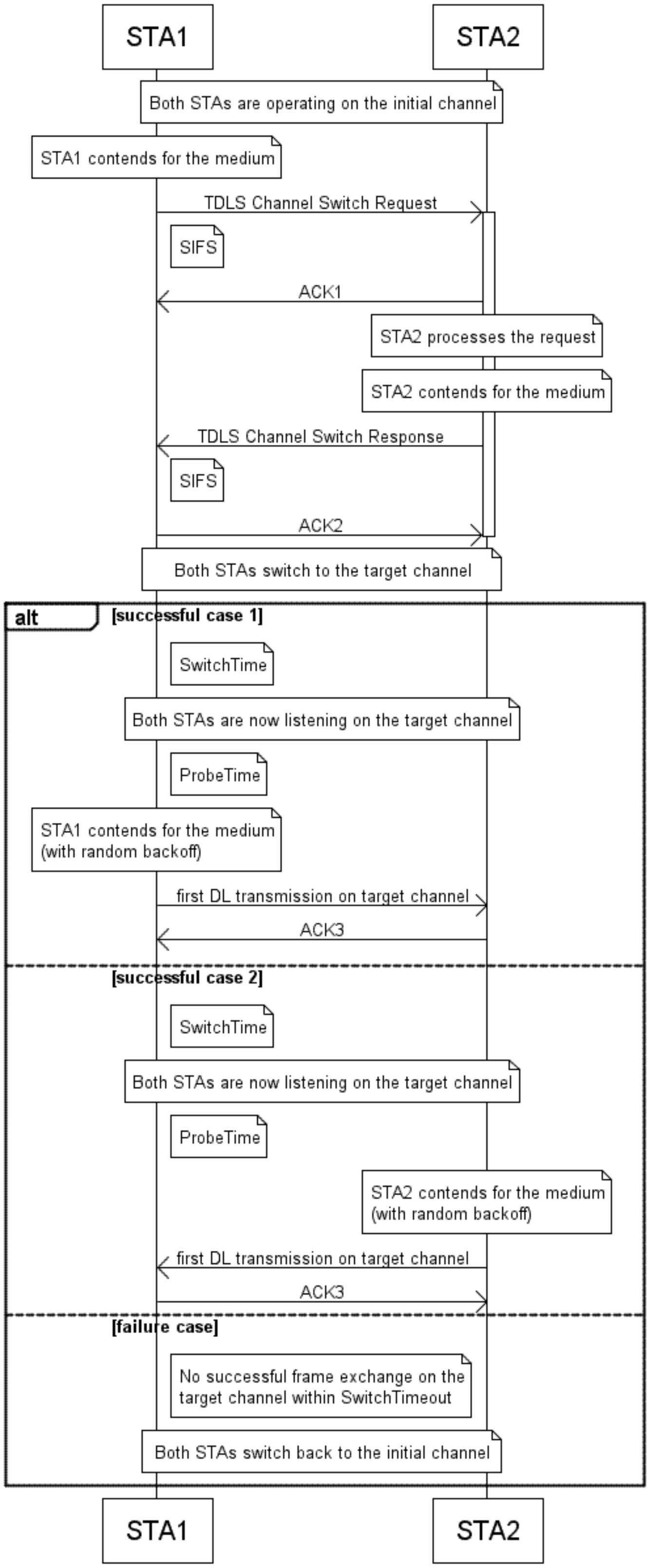

STA1
STA2
Both STAs are operating on the initial channel
STA1 contends for the medium
TDLS Channel Switch Request
SIFS
ACK1
STA2 processes the request
STA2 contends for the medium
TDLS Channel Switch Response
SIFS
ACK2
Both STAs switch to the target channel
alt
[successful case 1]
SwitchTime
Both STAs are now listening on the target channel
ProbeTime
STA1 contends for the medium (with random backoff)
first DL transmission on target channel
ACK3
[successful case 2]
SwitchTime
Both STAs are now listening on the target channel
ProbeTime
STA2 contends for the medium (with random backoff)
first DL transmission on target channel
ACK3
[failure case]
No successful frame exchange on the target channel within SwitchTimeout
Both STAs switch back to the initial channel
STA1
STA2

Figure 10-21—Events occurring for a TDLS direct-link channel switch

Both the TDLS Channel Switch Request frame and the TDLS Channel Switch Response frame shall contain a Channel Switch Timing element. The SwitchTime and SwitchTimeout values in the TDLS Channel Switch Timing element included in the TDLS Channel Switch Request frame shall specify the values required at the STA sending the TDLS Channel Switch Request frame. The SwitchTime and SwitchTimeout values specified in the TDLS Channel Switch Timing element included in the TDLS Channel Switch Response frame shall meet the requirements at the STA sending the TDLS Channel Switch Response frame and shall be equal to or larger than the values specified in the TDLS Channel Switch Request frame. The timing parameters specified in the Channel Switch Timing element included in the TDLS Channel Switch Response frame shall be used for the TDLS channel switching procedure. This procedure causes the larger of the two switch times to become the value that is transmitted in the TDLS Channel Switch Response frame.

The TDLS peer STA shall be in PS mode with the AP and shall not be involved in an active Service Period with the AP before sending a TDLS Channel Switch Request frame or a TDLS Channel Switch Response frame with Status Code set to 0 (“Successful”). The TDLS peer STA that receives a TDLS Channel Switch Request frame may enter PS mode with the AP prior to sending the TDLS Channel Switch Response frame.

Because there is at least a backoff between the TDLS Channel Switch Request frame and the TDLS Channel Switch Response frame, there is a (small) probability that two STAs issue a TDLS Channel Switch Request frame at more or less the same time. To reduce the probability for this event to occur, a TDLS peer STA should not transmit a TDLS Channel Switch Request when a TDLS Channel Switch Request frame is received and no TDLS Channel Switch Response has been transmitted in response. If a TDLS Channel Switch Request frame is received from the TDLS peer STA to which a pending TDLS Channel Switch Request frame was previously sent before receiving TDLS Channel Switch Response, the TDLS initiator STA shall not reply to the TDLS Channel Switch Request frame and the TDLS responder STA shall reply to the TDLS Channel Switch Request frame.

If a TDLS Channel Switch Response frame does not imply a channel switch because the STAs already are on the requested channel, then the SwitchTime and ProbeTime may be skipped and both TDLS peer STAs continue to operate on the requested channel. To cross means that a TDLS Channel Switch Request frame is received from a peer STA after transmitting a TDLS Channel Switch Request frame to the TDLS peer STA, instead of the expected TDLS Channel Switch Response frame.

When a TDLS peer STA does not receive an acknowledgment to a TDLS Channel Switch Response frame, it may retransmit the frame but the number of retransmissions shall be lesser of the maximum retry limit and dot11TDLSPeerSTAMissingAckRetryLimit.

A channel switch from an off-channel to the base channel may be accomplished by sending a TDLS Channel Switch Response frame indicating the base channel as the target channel, without prior TDLS Channel Switch Request frame. The Channel Switch Timing element shall be the same as contained in the Channel Switch Response frame that caused the switch to the off-channel.

TDLS Channel Switching shall not be used when the TDLS Channel Switching Prohibited subfield included in the Extended Capability element of the Association Response frame or Reassociation Response frame that led to the current association is equal to 1.

10.22.6.1 General behavior on the off-channel

If dot11SpectrumManagementRequired is true, a TDLS peer STA shall not transmit a TDLS Channel Switch request specifying an off-channel where radar detection is required, unless the STA has tested that channel for the presence of radars according to regulatory requirements. If a TDLS peer STA that is operating in such a channel detects radar, the TDLS peer STA shall discontinue transmissions according to regulatory requirements, and it shall send a TDLS Channel Switch Request indicating a switch to the base channel. The channel switch avoids an interruption on the direct link.

The TDLS peer STA initiating the switch to the channel where radar detection is required shall be the DFS owner.

The secondary channel of an existing 40 MHz network shall not be selected as an off-channel.

On an off-channel, the TDLS peer STAs remain associated with their BSS, so the BSSID remains the same.

It is recommended that in general TDLS STAs propose target channels that have no detectable medium occupancy. If no such channel is available, then it is recommended that the TDLS STA propose a target channel where beacons are detected but with little or no additional medium occupancy. It is further recommended that TDLS STAs do not propose a target channel where the presence of beacons indicate that ACM bits are set, unless little or no additional medium occupancy is detected.

10.22.6.2 Setting up a 40 MHz direct link

10.22.6.2.1 General

A 40 MHz off-channel direct link may be started if both TDLS peer STAs indicated 40 MHz support in the Supported Channel Width Set field of the HT Capabilities element (which is included in the TDLS Setup Request frame and the TDLS Setup Response frame).

Switching to a 40 MHz off-channel direct link is achieved by including the following information in the TDLS Channel Switch Request:

— Operating Class element indicating 40 MHz Channel Spacing

— Secondary Channel Offset element indicating SCA or SCB

A 40 MHz off-channel direct link shall not be established in the 2.4 GHz band.

The TDLS peer STA initiating the switch to the 40 MHz off-channel shall be the DO STA.

10.22.6.2.2 Basic 40 MHz functionality

TDLS peer STAs may transmit 40 MHz PPDUs on a 40 MHz direct link. A TDLS peer STA shall not transmit a 20 MHz PPDU in the secondary channel of its 40 MHz direct link.

10.22.6.2.3 Channel selection for a 40 MHz direct link

If a TDLS peer STA chooses to start a 40 MHz direct link that occupies the same two channels as an existing 40 MHz network (i.e., a 20/40 MHz BSSs or a 40 MHz direct link), then it shall select primary and secondary channels of the new direct link that are identical to the primary and secondary channels of the existing 40 MHz network, unless the TDLS peer STA discovers that on these two channels there are existing 40 MHz networks with different primary and secondary channels.

If a TDLS peer STA chooses to start a 40 MHz direct link, the selected secondary channel should correspond to a channel on which no beacons are detected.

10.22.6.2.4 Switching from a 40 MHz to a 20 MHz direct link

Switching from a 40 MHz off-channel direct link to a 20 MHz off-channel direct link is established through a TDLS channel switch. When on a 40 MHz off-channel direct link, a requested switch to a 20 MHz direct link shall always be accepted.

10.22.6.2.5 CCA sensing and NAV assertion in a 40 MHz direct link

When active on a 40 MHz direct link, the TDLS peer STAs shall follow the CCA rules as defined in 10.15.9 and the NAV rules as defined in 10.15.10.

10.22.6.3 TDLS channel switching and power saving

A TDLS direct link shall not be switched to an off-channel during a TDLS PU-APSD service period. While on an off-channel, a TDLS peer STA shall not enter PU-APSD power save mode.

A TDLS direct link may be switched to an off-channel when TDLS Peer PSM is active on the link. The wakeup windows occur on the off-channel in the same way they would have occurred had the STAs remained on the base channel. Suspension of the wakeup windows implies a switch back to the base channel.

10.23 Wireless network management procedures

10.23.1 Wireless network management dependencies

When dot11WirelessManagementImplemented is true, the STA is a WNM STA and dot11ExtendedChannelSwitchActivated and dot11RadioMeasurementActivated shall be true. This subclause describes WNM procedures for requesting and reporting WNM capabilities between STAs that support WNM procedures.

When dot11WirelessManagementImplemented is true, and one or more of bit 7 to bit 27 in the Extended Capabilities element have the value 1, the Extended Capabilities element shall be included in Beacon frames, Association Request and Response frames, Reassociation Request and Response frames, and Probe Request and Response frames. When dot11WirelessManagementImplemented is true, for each bit 7 to bit 27 in the received Extended Capabilities element that is 0, a STA shall not request the corresponding feature from the sending STA. A WNM STA receiving a request for a WNM feature from another STA shall reject the request if the receiving WNM STA has not advertised support for the corresponding WNM feature.

10.23.2 Event request and report procedures

10.23.2.1 Event request and event report

The Event Request and Event Report frames enable network real-time diagnostics. A STA that has a value of true for dot11MgmtOptionEventsActivated is defined as a STA that supports event requests and reports. A STA for which dot11MgmtOptionEventsActivated is true shall set the Event field of the Extended Capabilities element to 1. If dot11MgmtOptionEventsActivated is true, a STA shall log all Transition, RSNA, Peer-to-Peer, and WNM Log events, including the corresponding TSF, UTC Offset and Event Time Error.

The STA log of events shall not be cleared as a result of BSS transitions. However, if the STA moves to a different ESS or IBSS, the STA shall delete all event log entries.

A STA that supports event reporting shall only send an Event Request or Event Report frame to a STA within the same infrastructure BSS or the same IBSS whose last received Extended Capabilities element contained a value of 1 for the Event bit in the Capabilities field. If the STA receives an Event Request frame without error and it supports event reporting, it shall respond with an Event Report frame that includes the Dialog Token that matches the one in the Event Request frame.

The permitted source and destination STAs for an Event Request frame are shown in Table 10-6.

An AP may include zero or more Event Request elements in an Event Request frame. Each Event Request element contains an Event Token that associates this Event Request with any subsequent Event Report elements. When sending Event Report elements, a STA shall use the same Event Token that was included in the original request.

Only a single Event Request frame from a STA is outstanding at a given STA at any time. If a STA that supports event reporting receives a subsequent Event Request frame with a different Dialog Token before completing the Event Report for the previous request from the requesting STA, the receiving STA shall only respond to the most recent request frame.

Upon a BSS transition, the STA shall cancel any event requests in the latest Event Request frame.

The Event Request elements can contain conditions that specify events to be reported and conditions that establish event reporting when a STA experiences problems or failures. A STA sends an Event Request frame containing one or more Event Request elements, each of which contains zero or more subelements. Subelements are defined for each event type. A STA shall include in the corresponding Event Report element only those events that meet the specified event conditions within the current ESS or IBSS.

In each Event Report element, a STA shall include a Status field that indicates the result of the event request/report transaction. If the STA is able to return one or more Event Report elements, then the STA shall return a value of “Successful” in the Event Report element. If the STA has no logged events of the requested type, the STA shall return a value of Successful in the Event Report element without an event included in the Event Report field. If the STA is unable to process the request at that time, the STA shall return a value of “Request failed” in the Event Report element. If a STA refuses an event request, the STA shall return a value of “Request refused” in the Event Report element. The reasons for refusing an event request are outside the scope of this standard but may include reduced quality of service, unacceptable power consumption, measurement scheduling conflicts, or other significant factors. If the STA is incapable of generating an Event Report of the type specified in the Event Request frame, the STA shall return a value of “Request incapable” indicating that the requester should not request again.

If the Event Report elements do not fit into a single MMPDU, the reporting STA shall send the remaining elements in additional Event Report frames until all Event Report elements have been returned to the requesting STA. In any subsequent Event Report frame and for all remaining Event Report elements a reporting STA shall include the same Dialog Token and Event Token, respectively, that was sent in the corresponding Event Request frame. When multiple MMPDUs are required, the non-AP STA shall include complete Event Report elements and shall not fragment an element across multiple MMPDUs.

A STA shall transmit both the Event Request frame and the Event Report frame only with an individually addressed destination address. In an infrastructure BSS, only an AP shall transmit an Event Request frame to a non-AP STA. An AP that supports event reporting shall discard received Event Request frames.

When a STA transmits an Event Request frame to another STA it shall indicate the types of events requested by setting the Event Type field and shall indicate the maximum number of logged events to report by using the Event Response Limit field in each included Event Request element. If the number of available logged events of the requested type exceeds the Event Response Limit, the STA shall only report an Event Response Limit number of the most recent events. If there are no available logged events of the type specified in the Event Request frame, the STA shall send the Event Report frame without any Event Report element. The reporting STA shall send all available Event Report elements for the requested Event Type when the Event Request field is not present in the Event Request element.

A STA may include a Destination URI element in the Event Request frame. The AP includes the URI in the Destination URI element that the reporting non-AP STA may use to send Event Reports, upon the loss or interruption of connectivity to the BSS.

On receipt of an Event Request frame with an Destination URI element, the reporting non-AP STA SME may send the Event Report to the AP using the Destination URI with another network interface (if available). The non-AP STA SME shall only send the Event Report to the URI contained in the Destination URI element after detecting loss of BSS connection.

The non-AP STA shall determine loss of connection to the AP that issued the Event Request frame when it has not detected any Beacon frames from the AP for a period no less than the ESS Detection Interval.

If the BSS connection is reestablished to the AP that transmitted the Event Request frame, the non-AP STA shall transmit the corresponding Event Report frame to the AP without using the Destination URI.

When the non-AP STA uses the Destination URI mechanism, it shall transport the payload of the Event Report frame using the URI given in the Destination URI Element. An example use of the Destination URI is given in Annex U.

10.23.2.2 Transition event request and report

The Transition Event report provides information on the previous transition events for a given non-AP STA. The Transition Event request and report are only permitted in the infrastructure BSS.

Each STA supporting the Transition Event shall log up to and including the last five Transition events occurring since the STA associated to the ESS. A STA may log more than five of the most recent Transition events.

Upon receipt of an Event Request frame containing an Event Request element including a Transition Event request, the non-AP STA shall respond with an Event Report frame that includes available Event Report elements within the current ESS for the Transition event type.

Transition Event Request subelements are used to specify conditions for reporting of transition events. If any Transition Event Request subelements are present in the Event Request frame, the reporting non-AP STA shall include in the Event Report frame only those available Transition Event Report elements that meet the transition event reporting condition(s) specified in the Event Request frame. If no transition event subelements are present in the Event Request field, the reporting STA shall include all available Transition Event Report elements. A STA that encounters an unknown subelement ID value in a transition event request received without error shall ignore that subelement and shall parse remaining Event Request fields for additional information subelements with recognizable subelement ID values.

A Frequent Transition subelement in an Event Request frame defines conditions for frequent transition. Frequent transition occurs when the number of BSS transitions exceeds the value of the Frequent Transition Count Threshold within the indicated Time Interval value as defined in the Frequent Transition subelement in 8.4.2.69.2. A STA that receives a Frequent Transition subelement shall, at each BSS transition, apply the conditions for frequent transition to the log of transition events. If the logged transition events meet the conditions for frequent transition, the STA shall send an Event Report frame including a Transition Event Report element with Event Report Status set to Detected Frequent Transition and include in that Event Report element the last logged transition event.

For transition logging and reporting purposes, the transition time is defined as the time difference between the starting time and the ending time of a transition between APs, even if the transition results in remaining on the same AP.

The starting time is one of the following items:

- The start of a search for an AP, when the transition reason is 4 (first association to WLAN).
- The latest time that a frame could have been transmitted or received on the source BSS.

— The start of a search for an AP, after determination that a transition has failed.

The ending time is one of the following items:

— The earliest time that a data frame can be transmitted or received on the target BSS, after completion of RSN, IEEE 802.1X, or other authentication and key management transmissions, when such are required by the target BSS.

— The time that a determination is made that the transition has failed.

10.23.2.3 RSNA event request and report

The RSNA Event Report provides authentication events for a given non-AP STA. The RSNA Event Request and Report are only permitted in an infrastructure BSS.

Each STA supporting the RSNA Event shall log up to and including the last five RSNA events occurring since the STA associated to the ESS. A STA may log more than five of the most recent RSNA events.

Upon receipt of an Event Request frame containing an Event Request element including an RSNA Event request, the non-AP STA shall respond with an Event Report frame that includes available Event Report elements within the current ESS for the RSNA event type.

If an RSNA Event Request subelement is present in the Event Request field, the reporting non-AP STA shall include available Event Report elements that meet the specified condition for the RSNA event type. If no RSNA Event Request subelement is present in the Event Request field, the reporting STA shall include all available RSNA Event Report elements. A STA that encounters an unknown subelement ID value in an RSNA event request received without error shall ignore that subelement and shall parse remaining Event Request fields for additional information subelements with recognizable subelement ID values.

10.23.2.4 Peer-to-Peer Link event request and report

The Peer-to-Peer Link event report provides peer-to-peer connectivity events for a given non-AP STA. A Peer-to-Peer event occurs when a Peer-to-Peer link is initiated or terminated.

Each STA supporting the Peer-to-Peer event shall log up to and including the last five Peer-to-Peer events occurring since the STA associated to the ESS or IBSS. A STA may log more than five of the most recent Peer-to-Peer events. When a link is initiated, a STA shall log and record the TSF time of the Peer-to-Peer event without a connection time. When a Peer-to-Peer link is terminated, a STA shall log the Peer-to-Peer Link event including the connection time for the terminated link and shall delete from the log any initiation event for the same Peer-to-Peer link.

Upon receipt of an Event Request frame containing an Event Request element including a Peer-to-Peer Link event request, the non-AP STA shall respond with an Event Report frame that includes available Event Report elements within the current ESS or IBSS for the Peer-to-Peer event type. When a STA includes a Peer-to-Peer event report element for a link initiation, the STA shall include a connection time for the event report element which indicates the time difference from the event timestamp to the current time.

If a Peer-to-Peer Link Event Request subelement is present in the Event Request field, the reporting non-AP STA shall include available Event Report elements that meet the specified condition for the Peer-to-Peer event type. If no Peer-to-Peer Link Event Request subelements are present in the Event Request field, the reporting STA shall include all available Peer-to-Peer Event Report elements. A STA that encounters an unknown subelement ID value in a Peer-to-Peer event request received without error shall ignore that subelement and shall parse remaining Event Request fields for additional information subelements with recognizable subelement ID values.

10.23.2.5 WNM Log event request and report

The WNM Log event report is intended to capture PHY and MAC layer events related to the operation of those layers in vendor-specific, human-readable (ASCII text) form. The WNM Log is a general reporting mechanism that can apply to configuration or monitoring behavior for PHY and MAC. The WNM log is particularly useful for logging success or failure events across areas such as driver status, IEEE 802.11 or IEEE 802.1X authentication, authorization, status changes while associated or unassociated.

For example:

```
<0>Oct 03 17:47:00 00:01:02:03:04:05 Adapter DLL Service initialized
<1>Oct 03 17:48:40 00:01:02:03:04:05 Authentication started
<1>Oct 03 17:48:46 00:01:02:03:04:05 802.1X Authentication Failed, credential failure
<1>Oct 03 17:49:00 00:01:02:03:04:05 Authentication success
```

A non-AP STA that supports event reporting may be queried at any time for its current set of WNM Log messages. The WNM Log messages returned by the non-AP STA may provide insight into the trouble being experienced by non-AP STA.

Upon receipt of an Event Request frame containing an Event Request element including a WNM Log Event request, the non-AP STA shall respond with an Event Report frame that includes WNM Log Event Report elements.

10.23.2.6 Vendor Specific event request and report

The procedures for use of the Vendor Specific Event Request and Report are vendor specific and are not part of this standard.

10.23.3 Diagnostic request and report procedures

10.23.3.1 Diagnostic request and diagnostic report

The Diagnostic Request and Diagnostic Report protocol provides a tool to diagnose and debug complex network issues. A STA that has a value of true for dot11MgmtOptionDiagnosticsActivated is defined as a STA that supports diagnostics reporting. A STA for which dot11MgmtOptionDiagnosticsActivated is true shall set the Diagnostics field of the Extended Capabilities element to 1.

A STA that supports diagnostics reporting shall only send a Diagnostics Request or Diagnostics Report frame to a STA within the same infrastructure BSS or the same IBSS whose last received Extended Capabilities element contained a value of 1 for the Diagnostics bit in the Capabilities field.

The Diagnostic Request frame contains a unique Dialog Token. A Dialog Token value of 0 is a reserved value and shall not be used. The source and destination of a Diagnostic Request frame shall both be members of the same BSS. The permitted source and destination STAs for a Diagnostic Request frame are shown in Table 10-6. A STA may include one or more Diagnostic Request elements in a Diagnostic Request frame. Each Diagnostic Request element contains a unique Diagnostic Token that identifies the element within the Diagnostic Request frame.

If a STA that supports diagnostic reporting receives a Diagnostic Request frame without error, the STA shall respond with a Diagnostic Report frame that includes the Dialog Token that matches the one in the Diagnostic Request frame. Each Diagnostic Report element that corresponds to the Diagnostic Request element shall contain the same Diagnostic Token that was included in the original request.

Only a single Diagnostic Request frame from a STA is outstanding at a given STA at any time. If a STA receives a subsequent Diagnostic Request frame with a different Dialog Token before completing the Diagnostic Report for the previous request from the requesting STA, the STA shall only respond to the most recent Request frame. The STA need not send a Diagnostic Report frame for the previous Diagnostic Request frame.

All outstanding diagnostic requests, as indicated by received MLME-DIAGREQUEST.indication primitives, are cancelled upon a BSS transition, except when the BSS transition occurs as a result of responding to or initiating a diagnostic request. All outstanding diagnostic requests, as indicated by received MLME-DIAGREQUEST.indication primitives, are cancelled after the time indicated in the Diagnostic Timeout field, in the Diagnostic Request frame. When a STA that supports diagnostic reporting receives a Diagnostic Request frame with a Diagnostic Request Type of Cancel Diagnostic Request, the STA cancels all outstanding diagnostic requests, and discards all pending diagnostic reports, as indicated by received MLME-DIAGREQUEST.indication primitives.

All Diagnostic Report elements shall include a Status field that indicates the overall result of the transaction. If the STA is able to complete the diagnostic request made in the Diagnostic Request element, then a value of “Successful” shall be returned. If the STA is unable to process the request at that time a value of “Fail” shall be returned. If the STA is incapable of generating a report of the type specified, it shall return a value of “Incapable.” If the STA cannot support the request for any other reason, the value of Refused shall be returned.

A STA shall only transmit both the Diagnostic Request frame and the Diagnostic Report frame with an individually addressed destination address.

A STA may include a Destination URI element in the Event Request frame. The AP includes the URI in the Destination URI element that the reporting non-AP STA may use to send Diagnostic Reports, upon the loss or interruption of connectivity to the BSS.

On receipt of an Diagnostic Request frame with an Destination URI element, the reporting non-AP STA SME may send the Diagnostic Report to the AP using the Destination URI with another network interface (if available). The non-AP STA SME shall only send the Diagnostic Report to the URI contained in the Destination URI element after detecting loss of BSS connection.

The non-AP STA shall determine loss of connection to the AP that issued the Diagnostic Request frame when it has not detected any Beacon frames from the AP for a period no less than the ESS Detection Interval.

If the BSS connection is reestablished to the AP that transmitted the Diagnostic Request frame, the non-AP STA shall transmit the corresponding Diagnostic Report frame to the AP without using the Destination URI.

When the non-AP STA uses the Destination URI mechanism, it shall transport the payload of the Diagnostic Report frame using the URI given in the Destination URI Element. An example use of the Destination URI is given in Annex U.

If a non-AP STA that receives an Destination URI subelement in an Diagnostic Request fails to detect any Beacon frames, belonging to the AP that issued the Diagnostic Report request, for the period specified by the ESS detection interval, it may use the URI specified in the Destination URI subelement to transport the Diagnostic Report to the AP.

If the Diagnostic Report elements do not fit into a single MMPDU, the reporting STA shall send the remaining elements in additional frames until all Diagnostic Report elements have been returned to the requesting STA. A STA shall include the same Dialog Token and Diagnostic Token that was transmitted in the corresponding Diagnostic Request frame in each subsequent Diagnostic Report frame and Diagnostic

Report elements. When multiple MMPDUs are required, the STA shall include complete Diagnostic Report elements and shall not fragment an element across multiple MMPDUs.

A STA that supports diagnostic reporting may cancel a previously sent Diagnostic Request frame for which it has not yet received a Diagnostic Report frame by sending a Diagnostic Request frame with the Diagnostic Request Type field value of 0, indicating "Cancelled," to the STA to which it previously sent the Diagnostic Request frame. A STA that supports diagnostic reporting shall inform a STA from which it has previously received a Diagnostic Request frame that the request has been locally cancelled by sending a Diagnostic Report frame with the Diagnostic Status field set to a value of 4, indicating "Cancelled," to the requesting STA. In a Diagnostic Request frame with the Diagnostic Request Type field value of 0, and a Diagnostic Report frame with the Diagnostic Status field set to a value of 4, no Diagnostic Information subelements are included in the Diagnostic Request or Response element.

10.23.3.2 Configuration Profile report

The Configuration Profile report enables an AP to discover the current profile in use for an associated device, and additional profiles for the current ESS. A non-AP STA that supports diagnostic reporting and receives a Configuration Profile report request type shall respond with a Diagnostic Report frame that includes all available Diagnostic elements allowed for the type.

Devices that support multiple configuration profiles for an ESS may include multiple Diagnostic Report elements in a single Diagnostic Report frame (or multiple frames if required). Each Diagnostic Report element shall contain a Profile ID element that uniquely identifies the configuration profile(s) for the current ESS that are available on the device.

10.23.3.3 Manufacturer information STA report

The Manufacturer Information STA Report enables an AP to discover static manufacturer specific data about an associated STA device. A non-AP STA that supports diagnostic reporting and receives a Manufacturer Information STA Report request type shall respond with a Diagnostic Report frame that includes all available Diagnostic elements allowed for the type.

When more than one Antenna Type/Antenna Gain pair is enabled, multiple Antenna Type subelements, shall be included in the Manufacturer Information STA Report Diagnostic Report element.

When more than one Collocated Radio Type, or Device Type is supported, multiple Collocated Radio Type subelements, or Device Type subelements shall be included in the Manufacturer Information STA Report Diagnostic Report element. If the existence or the type of collocated radio is unknown, no Collocated Radio Type subelements shall be included.

10.23.3.4 Association diagnostic

The purpose of the association diagnostic is to determine that a STA is able to associate with a designated BSS. This test directs an association to be completed with a specific AP.

To initiate the test, an AP that supports diagnostic reporting shall send a Diagnostic Request frame containing a Diagnostic Request Type field set to 3 (i.e., Association Diagnostic) to a STA that supports diagnostic reporting. The AP shall not send an association diagnostic request with a designated BSS that is not part of the ESS and the STA receiving an association diagnostic request shall reject requests to perform diagnostics on a BSS that is not part of the ESS. All parameters required to complete the association are included in the Diagnostic Request element.

Upon receipt of the Diagnostic Request frame containing a Diagnostic Request element specifying an Association request, the STA determines whether to accept the request. If the STA declines the request, it shall send a Diagnostic Report frame with the Status field of a Diagnostic Report element set to Refused. If the STA accepts the request, it shall cause an authentication to occur to the AP indicated in the Diagnostic Request element and the STA's SME shall issue an MLME-DIAGREPORT.request primitive, indicating the results of the diagnostic.

One means to cause an authentication to occur is for the STA's SME to issue an MLME-DEAUTHENTICATE.request primitive to deauthenticate from the current AP, and an MLME-AUTHENTICATE.request primitive to authenticate to the AP indicated in the Diagnostic Request element. If successful, the STA shall issue an MLME-(RE)ASSOCIATE.request to associate with the AP indicated in the Diagnostic Request element. If successful, the STA's SME shall then issue an MLME-DEAUTHENTICATE.request to deauthenticate from the AP indicated in the Diagnostic Request element, reassociate with the AP from which it received the Diagnostic Request, and issue an MLME-DIAGREPORT.request primitive, indicating the results of the diagnostic

10.23.3.5 IEEE 802.1X authentication diagnostic

The purpose of the IEEE 802.1X authentication diagnostic is to determine that the STA is able to complete an IEEE 802.1X authentication with a designated BSS. This test directs an association and IEEE 802.1X authentication to be completed with a specific AP. If a STA that supports diagnostic reporting also supports RSN, the STA shall support the IEEE 802.1X authentication diagnostic.

To initiate the test, an AP that supports diagnostic reporting shall send a Diagnostic Request frame containing a Diagnostic Request Type field set to 4 (i.e., IEEE 802.1X Authentication Diagnostic) to a STA that supports diagnostic reporting. A STA that supports diagnostic reporting in an IBSS or an AP that supports diagnostic reporting shall not send an IEEE 802.1X authentication diagnostic request with a designated BSS that is not part of the ESS, or IBSS and a STA that supports diagnostic reporting which receives a diagnostic request shall reject requests to perform diagnostics on other networks. The AP, EAP method and credential type values included in the AP Descriptor, EAP Method and Credential Type subelements in the Diagnostic Request element shall be used to complete the association and IEEE 802.1X authentication.

Upon receipt of the Diagnostic Request frame containing a Diagnostic Request element specifying an IEEE 802.1X Authentication Diagnostic request, the STA determines whether to accept the request. If the STA declines the request, it shall send a Diagnostic Report frame with the Status field of a Diagnostic Report element set to Refused. If the STA accepts the request, it shall cause an IEEE 802.1X authentication to occur to the AP indicated in the Diagnostic Request element and the STA's SME shall issue an MLME-DIAGREPORT.request primitive indicating the results of the diagnostic.

One means to cause an authentication to occur is for the STA's SME to issue an MLME-DEAUTHENTICATE.request to deauthenticate from the current AP, and an MLME-AUTHENTICATE.request to authenticate to the AP indicated in the Diagnostic Request element. If successful, the STA shall issue an MLME-(RE)ASSOCIATE.request to the AP indicated in the Diagnostic Request element. If (re)association succeeds, the STA shall try to complete IEEE 802.1X authentication using parameters specified in the Diagnostic Request element. The STA shall then issue an MLME-DEAUTHENTICATE.request to deauthenticate from the AP indicated in the Diagnostic Request element, would reassociate with the AP from which it received a diagnostic request, and issue an MLME-DIAGREPORT.request primitive, indicating the results of the diagnostic.

10.23.4 Location track procedures

10.23.4.1 Location track configuration procedures

A STA that has a value of true for dot11MgmtOptionLocationTrackingActivated is defined as a STA that supports location. A STA for which dot11MgmtOptionLocationTrackingActivated is true shall set the Location field of the Extended Capabilities element to 1.

In an infrastructure BSS, a non-AP STA shall not transmit Location Configuration Request frames.

A STA that supports location may configure another STA to transmit Location Track Notification frames for the purpose of tracking the receiving STA's location by sending Location Indication Channels, Location Indication Interval and Location Indication Broadcast Data Rate subelements in a Location Parameters element in a Location Configuration Request frame. The minimum Normal and In-Motion Report Interval in a Location Configuration Request frame is 500 ms.

A STA may transmit the Location Configuration Request frame as a broadcast or individually addressed frame. A STA that supports location and receives a broadcast Location Configuration Request frame shall only send a Location Configuration Response frame if the STA does not accept the parameters included in the Location Configuration Request.

A STA that supports location and receives an individually addressed Location Configuration Request shall respond with a Location Configuration Response frame. Upon successful reception of a new Location Configuration Request frame, the STA shall override any previously received Location Configuration Request frame with the new frame. If all Location Parameter subelements included in the Location Configuration Request are successfully configured on the receiving STA, then the STA shall include in the Location Configuration Response frame a single Location Status subelement indicating success. Upon successful configuration, the receiving STA shall start transmitting the Location Track Notification frames based on the Location Configuration Request frame parameters, as described in 10.23.4.2. If one or more Location Parameter subelements are unsuccessfully configured, then the STA shall include in the Location Configuration Response frame a Location Status subelement for each failed subelement indicating the subelement ID, the status value and the corresponding Location Parameter subelement as described in the paragraphs that follow.

The Location Status subelement has four possible status values: Success, Fail, Refused and Incapable. When the requesting STA receives a Location Configuration Response frame with Location Status indicating anything other than Success, the requesting STA shall assume the original request was not processed and no configuration took effect on the receiving STA and the requesting STA should take appropriate action based on the status value returned.

For Location Status Fail:

- If the receiving STA has been configured successfully prior to the current Location Configuration Request and continues to transmit Location Track Notification frames based on those parameters, the STA shall respond with its current Location Parameters subelements values.
- If the STA has no previously configured value, the STA shall respond with its minimum Location Parameters subelements that it is capable of supporting.
- The configuring STA may either retry the original request or send an alternate request.

For Location Status Incapable:

- The STA responding to the configuration request may include the minimum Location Parameters subelements that it is capable of supporting.

- The configuring STA shall not send another configuration request matching the previous configuration request while the reporting STA is associated to the same BSS.
- The configuring STA may send an alternate request.

For Location Status Refuse:

- The STA responding to the configuration request may include the minimum Location Parameters subelements that it is capable of supporting.
- The configuring STA may send an alternate request.

The location configuration methods, from highest to lowest precedence, are as follows: 1) an individually addressed Location Configuration Request frame, 2) broadcast Location Configuration Request frame. When a STA receives a new Location Configuration frame at the same or higher precedence than the previous it shall cancel the previous configuration and begin using the newest configuration.

The Location Indication Broadcast Data Rate subelement included in Location Configuration Request frames indicates the target data rate at which the STA shall transmit Location Track Notification frames. The Location Indication Broadcast Data Rate included in the Location Configuration Request frame should be a data rate defined in the basic data rate set.

The Indication Multicast Address field configured in the Location Indication Parameters subelement shall be a multicast locally administered IEEE MAC address as defined in IEEE Std 802 that is shared across all APs in the same ESS. The STA shall transmit Location Track Notification frames to the Indication Multicast Address with the BSSID field set to the wildcard BSSID. APs shall discard Location Track Notification frames that are not addressed to the Indication Multicast Address field configured for the ESS.

A non-AP STA shall terminate the transmission of Location Track Notification frames for any of the following reasons:

- The non-AP STA receives a Location Configuration Request frame from the STA to which it is currently associated that includes a Location Parameters element with a Location Indication Parameters subelement specifying an interval of 0.
- The non-AP STA fails to detect any Beacon frames, belonging to the same ESS that originally configured the non-AP STA, for the period specified by the essDetectionInterval value included in the Location Parameters element transmitted in the Location Configuration Request frame.
- The dot11MgmtOptionLocationTrackingActivated MIB attribute for the STA is false.
- The non-AP STA is disassociated for any reason from the ESS that configured it, including power off, or is configured by a different ESS.
- In an IBSS when the STA detects that it is no longer connected to the other STA that formed the IBSS.

NOTE 1—All Public Action frames, including the Location Track Notification frames, are Class 1 frames and the treatment of Public Action frames upon reception by STAs is defined in 10.3.

NOTE 2—User Applications should not send location information to other stations without the express permission of the user. User agents acquire permission through a user interface, unless they have prearranged trust relationships with users. Those permissions that are acquired through the user interface and that are preserved beyond the current browsing session (i.e., beyond the time when the BSS connection is terminated) are revocable and receiving stations should respect revoked permissions. Some user applications may have prearranged trust relationships that do not require such user interfaces. For example, while a social networking application might present a user interface when a friend performs a location request, a VOIP telephone may not present any user interface when using location information to perform an E911 function.

10.23.4.2 Location track notification procedures

Implementation of Location Track Notification is optional for a WNM STA. A STA that implements Location Track Notification has dot11MgmtOptionLocationTrackingImplemented set to true. When dot11MgmtOptionLocationTrackingImplemented is true, dot11WirelessManagementImplemented shall be true. A STA in which dot11MgmtOptionLocationTrackingActivated is true is defined as a STA that supports Location Track Notification. When Location Track Notification is supported, a STA configured by another STA as described in the previous subclause shall transmit Location Track Notification frames as shown in the informative diagram in Figure 10-22.

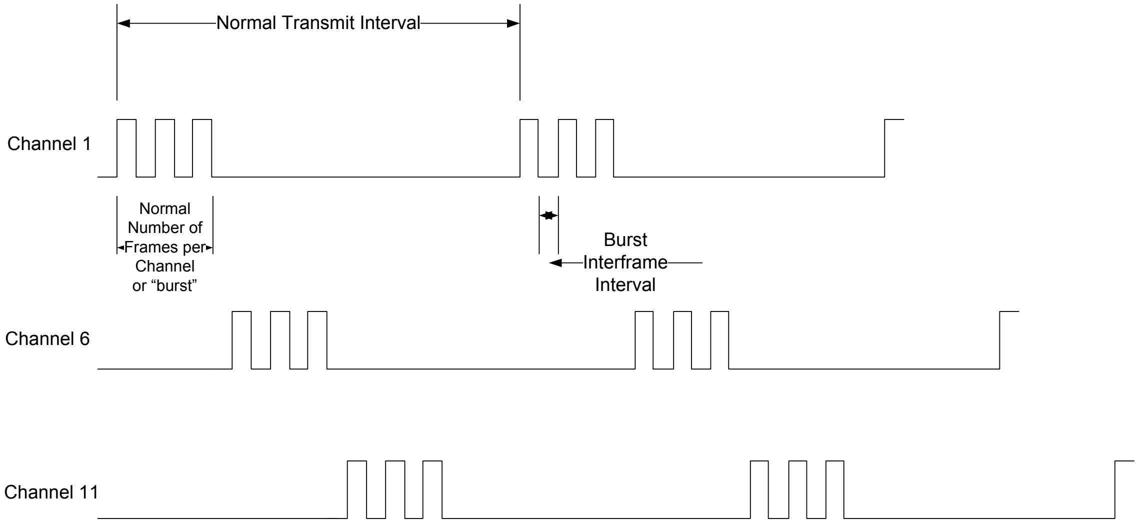

Normal Transmit Interval
Channel 1
Normal Number of Frames per Channel or "burst"
Burst Interframe Interval
Channel 6
Channel 11

Figure 10-22—STA transmission on three channels, three frames per channel with Normal Report Interval

Implementation of Motion Detection or the Time of Departure reporting is optional for a WNM STA. A STA that implements Motion Detection has dot11MgmtOptionMotionDetectionImplemented set to true. When dot11MgmtOptionMotionDetectionImplemented is true, dot11WirelessManagementImplemented shall be true. A STA with a value of true for dot11MgmtOptionMotionDetectionActivated is defined as a STA that supports Motion Detection. A STA that implements Time of Departure has dot11MgmtOptionMotionTODImplemented set to true. When dot11MgmtOptionTODImplemented is true, dot11WirelessManagementImplemented shall be true. A STA with a value of true for dot11MgmtOptionTODActivated is defined as a STA that supports Time of Departure for location tracking.

The STA transmits Location Track Notification frames based on the following parameters and procedures described in 10.23.4.1:

a) Location Indication Channels. This subelement indicates the channels on which the STA shall transmit Location Track Notification frames.
b) Indication Multicast Address
    1) For non-IBSS networks, the STA shall transmit the Location Track Notification frames to the Indication Multicast Address field in the Location Indication Parameters subelement configured by the Location Configuration Request frame.
    2) An AP shall discard any Location Track Notification frame received from a STA that does not match the Indication Multicast Address field value for the AP's ESS.
    3) For IBSS networks, the STA shall transmit the Location Track Notification frames to the destination address of the STA that configured the STA using Location Configuration Request frames.

c) Location Indication Interval

   1) When the STA is stationary or dot11MgmtOptionMotionDetectionActivated is false, the STA shall transmit a sequence of groups of Location Track Notification frames on each channel. Each group of frames shall contain Normal Number of Frames Per Channel field frames. The first frame in each group of Location Track Notification frames shall be separated from the first frame in the previous group of Location Track Notification frames by a minimum time duration indicated by the value of the Normal Report Interval times the value of the Normal Report Interval Units field.

   2) When the STA is in motion and dot11MgmtOptionMotionDetectionActivated is true, the STA shall transmit a sequence of groups of Location Track Notification frames on each channel. Each group of frames shall contain In-Motion Number of Frames Per Channel field frames. The first frame in each group of Location Track Notification frames shall be separated from the first frame in the previous group of Location Track Notification frames by a minimum time duration indicated by the value of the In-Motion Report Interval times the value of the In-Motion Report Interval Units field.

   3) If a STA is configured to transmit on multiple channels, the STA shall transmit the frames on a single channel before continuing onto the next channel in the configured list of channels.

   4) All Location Track Notification frames transmitted on a single channel shall be transmitted with a minimum gap specified by the Burst Interframe Interval field.

   5) A STA can never be stationary and in-motion at the same time, and therefore only the Normal Interval field or the In-Motion Interval field apply at any given moment.

d) Tracking Duration

   1) The STA shall transmit Location Track Notification frames until the Tracking Duration duration is reached.

   2) The duration starts as soon as the STA sends a Configuration Location Response frame with a Location Status value of Success.

   3) If the Tracking Duration is a nonzero value the STA shall transmit Location Track Notification frames, based on the Normal and In-Motion Report Interval field values, until the duration ends or is configured to terminate transmission as described in 10.23.4.1.

   4) If the Tracking Duration is 0 the STA shall continuously transmit Location Track Notification frames as defined by Normal and In-Motion Report Interval field values until configured to terminate transmission as described 10.23.4.1.

e) ESS Detection

   1) The ESS Detection Interval field is specified in the Location Indication Parameters subelement configured by the Location Configuration Request frame. The ESS Detection Interval defines how often the STA should check for beacons transmitted by one or more APs belonging to the same ESS that configured the STA.

   2) If no beacons from the ESS are received during this interval, the STA shall terminate transmission of Location Track Notification frames.

f) Location Indication Options

   1) The RM Enabled Capabilities element contained in (Re)Association Request frames indicates the STA’s ability to perform radio measurements as described in 8.4.2.47. The Location Indication Options subelement Options Used field Beacon Measurement Mode Used bit shall be set to 0 by the AP when the RM Enabled Capabilities element bits (defined in Table 8-119), Beacon Passive Measurement capability enabled, Beacon Active Measurement capability enabled and Beacon Table Measurement capability enabled are all set to 0. If any of RM Enabled Capabilities element bits Beacon Passive Measurement capability enabled, Beacon Active Measurement capability enabled or Beacon Table Measurement capability enabled are equal to 1 then the Location Indication Options subelement Options Used field Beacon Measurement Mode Used bit may be set to 1.

2) If the Location Indication Options subelement is included and the Options Used field with the Beacon Measurement Mode Used bit equal to 1 in the most recently received Location Configuration Request frame, the STA shall include in the Location Track Notification frames, the result of the most recent successful beacon measurement that was performed using the requested Beacon Measurement Mode contained in the Location Indications Options subelement.

3) If the STA has never performed a successful beacon measurement using the requested Beacon Measurement Mode prior to transmission of the Location Track Notification frame, the STA shall perform the beacon measurement using the requested Beacon Measurement Mode and include the results of that measurement in Location Track Notification frames.

4) After a successful Location Configuration Request that included the Location Indication Options subelement and Options Used field with Beacon Measurement Mode Used bit equal to 1, a STA should continue to perform beacon measurement as defined by the Beacon Measurement Mode periodically. How often and under what circumstances the STA performs this measurement is out of scope of this standard.

5) Whenever a STA includes the beacon measurement in the Location Track Notification frames, the STA shall set the Measurement Token field in the Measurement Report element to the same value as the Dialog Token field in the Location Configuration Request frame that initiated the transmission of the location track notification frames by the STA.

6) If the Location Indication Options subelement is not included in the most recently received Location Configuration Request frame or the Location Indication Options subelement is included with the Options Used field with Beacon Measurement Mode Used bit equal to 0, the STA shall not include any beacon measurements in the Location Track Notification frame

g) Location Indication Broadcast Data Rate. The STA shall transmit Location Track Notification frames at the data rate specified in this subelement.

h) Time of Departure

1) If dot11MgmtOptionTODActivated is true, the STA shall transmit this subelement in the Location Track Notification frame.

2) For all location tracking frames transmitted by a STA following a successful configuration, the Time of Departure subelement TOD Clock Rate field shall be set to the same value.

3) If the STA has multiple antennas, it shall transmit using an approximation to an omnidirectional pattern.

NOTE—The values of the fields in the Time of Departure subelement are measured by the PHY in real time, then passed without real-time requirements to the MAC via the TXSTATUS parameter of the PHY-TXSTATUS.confirm primitive.

The diagram in Figure 10-22 shows an example of Location Track Notification frame transmission, for a STA configured to transmit on three channels, with three frames per channel. In this example, a Normal Transmit Interval and Normal number of frames per channel are shown. When a STA is capable of motion detection and is in motion, the In-Motion Report Interval and In-Motion number of frames per channel would apply.

10.23.5 Timing measurement procedure

Implementation of Timing Measurement is optional for a WNM STA. A STA that has a value of true for dot11MgmtOptionTimingMsmtImplemented is defined as a STA that supports timing measurement. A STA for which dot11MgmtOptionTimingMsmtImplemented is true shall set the Timing Measurement field of the Extended Capabilities element to 1.

If dot11MgmtOptionTimingMsmtActivated is true, the Timing Measurement field in the Extended Capabilities element shall be set to 1 and the STA supports the timing measurement procedure. If dot11MgmtOptionTimingMsmtActivated is false the STA shall set the Timing Measurement field in the

Extended Capabilities element to 0 and STA does not support the timing measurement procedure. A STA that does not support the timing measurement procedure shall ignore a received Timing Measurement frame.

A STA that supports the timing measurement procedure may transmit a Timing Measurement Request frame to a peer STA to request it to initiate or to stop an ongoing Timing Measurement procedure, depending on the value of the Trigger field in the request frame. See Figure 10-23.

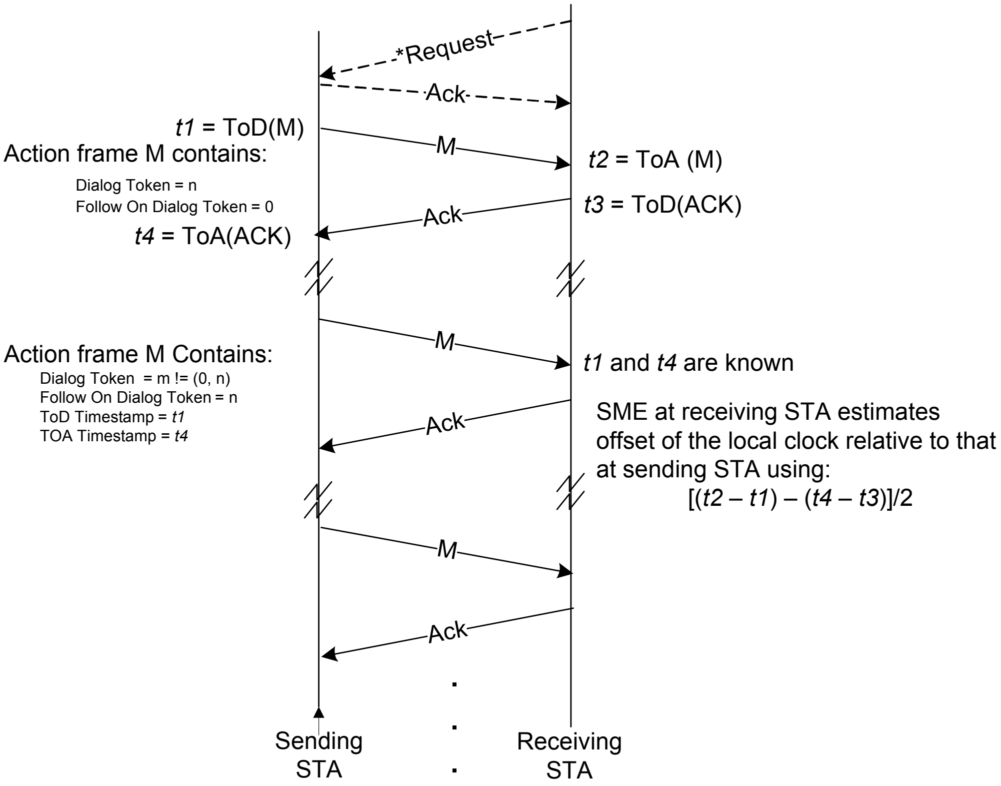


Figure 10-23—Timing measurement procedure

A STA that supports the timing measurement procedure may transmit Timing Measurement frames addressed to a peer STA that also supports the timing measurement procedure. One higher-layer protocol for synchronizing a local clock time between STAs using this feature is specified in IEEE Std 802.1AS.

A sending STA transmits Timing Measurement frames in overlapping pairs. The first Timing Measurement frame of a pair contains a nonzero Dialog Token. The follow up Timing Measurement frame contains a Follow Up Dialog Token set the value of the Dialog Token in the first frame of the pair. With the first Timing Measurement frame, both STAs capture timestamps. The sending STA captures the time at which the Timing Measurement frame is transmitted (t1). The receiving STA captures the time at which the Timing Measurement frame arrives (t2) and the time at which the ACK response is transmitted (t3). The sending STA captures the time at which the ACK arrives (t4). See Figure 6-16 in 6.3.57. In the follow up Timing Measurement frame, the sending STA transfers the timestamp values it captured (t1 and t4) to the receiving STA.

NOTE—A Timing Measurement frame can contain nonzero values in both the Dialog Token and Follow Up Dialog Token fields, meaning that the Action frame contains follow up information from a previous measurement, and new Timestamp values are captured to be sent in a future follow up Timing Measurement frame.

The offset of the clock at the receiving STA with respect to the clock at the sending STA is calculated using the equation that follows(assuming a symmetric wireless channel). See Figure 6-16 in 6.3.57.

Clock offset at receiving STA relative to sending STA = [(*t2* – *t1*) – (*t4* – *t3*)]/2

NOTE—An example of state machines and other computations for synchronizing a local clock time between IEEE 802.11 stations using the values of t1, t2, t3, and t4 provided by the Timing Measurement feature is found in Clause 12 of IEEE P802.1AS.

The acknowledgement procedure for Timing Measurement frames is the same as that for regular management frames (see 9.3.2.8). If the ACK for a transmitted Timing Measurement frame is not received, the sending STA may retransmit the frame. The sending STA shall capture a new set of timestamps for the retransmitted frame and its ACK.

On receiving a Timing Measurement frame with a Dialog Token for which timestamps have previously been captured, the receiving STA shall discard previously captured timestamps and capture a new set of timestamps.

10.23.6 BSS transition management for network load balancing

10.23.6.1 BSS Transition capability

The BSS Transition capability enables improved throughput, effective data rate and/or QoS for the aggregate of STAs in a network by shifting (via transition) individual STA traffic loads to more appropriate points of association within the ESS. In addition, the BSS Transition capability provides accounting session control information to a non-AP STA, which might be used to provide an alert to the non-AP STA's user that their session is almost over and the STA will be disassociated from the ESS.

The BSS Transition Management Query, BSS Transition Management Request, BSS Transition Management Response frames provide a means and a protocol to exchange the information needed to enable an AP to inform associated STAs that the BSS will be terminated and to enable a network to manage BSS loads by influencing STA transition decisions. A STA may provide neighbor report information to its associated AP for BSSs that it considers to be transition targets. This information enables the AP to request that the STA transition to a BSS that the STA also prefers.

Implementation of BSS Transition Management is optional for a WNM STA. A STA that implements BSS Transition Management has dot11MgmtOptionBSSTransitionImplemented set to true. When dot11MgmtOptionBSSTransitionImplemented is true, dot11WirelessManagementImplemented shall be true. A STA that has a value of true for dot11MgmtOptionBSSTransitionManagementActivated is defined as a STA that supports BSS transition management. A STA for which dot11MgmtOptionBSSTransitionActivated is true shall set the BSS Transition field of the Extended Capabilities element to 1.

The provisions in this clause for BSS transition management and network load balancing do not apply in an IBSS.

10.23.6.2 BSS transition management query

A non-AP STA supporting BSS transition management may request a BSS Transition Candidate list by sending a BSS Transition Management Query frame to its associated AP if the associated AP indicates that it supports the BSS Transition Capability in the Extended Capabilities element. A non-AP STA should include the BSS Transition Candidate List Entries field in the BSS Transition Management Query frame to indicate the non-AP STA's transition preferences. If the non-AP STA transmits a BSS Transition Query frame only to provide transition preferences to the AP, then the BSS Transition Query Reason field of the BSS Transition Management Query frame shall be set to a value of 19, indicating "Preferred BSS Transition Candidate List Included."

The BSS Transition Candidate List Entries field of a BSS Transition Management Query frame contains zero or more Neighbor Report elements describing the non-AP STA's preferences for target BSS candidates. The Preference field value of a Neighbor Report element used in a BSS Transition Management Query frame shall be between 1 and 255. The value of 0 is reserved. The values between 1 and 255 provide the indication of order, with 255 indicating the most preferred BSS within the given candidate list, decreasing numbers representing decreasing preference relative only to entries with lower values of the Preference field, and equal numbers representing equal preference.

10.23.6.3 BSS transition management request

An AP that supports BSS transition management shall respond to a BSS Transition Management Query frame with a BSS Transition Management Request frame. The AP may send an unsolicited BSS Transition Management Request frame to a non-AP STA at any time if the non-AP STA indicates that it supports the BSS Transition Management capability in the Extended Capabilities element. The AP may transmit a group addressed BSS Transition Management Request frame to associated non-AP STAs if all associated non-AP STAs support the BSS Transition Management capability. When the BSS Transition Management Request frame is transmitted as a group addressed frame, a receiving non-AP STA shall not respond with a BSS Transition Management Response frame. A non-AP STA that supports BSS transition management shall respond to an individually addressed BSS Transition Management Request frame with a BSS Transition Management Response frame.

The AP shall include the BSS Transition Candidate List Entries field in the BSS Transition Management Request frame if the AP has information in response to the BSS Transition Management Query frame. The BSS Transition Candidate List Entries field contains one or more Neighbor Report elements describing the preferences for target BSS candidates. A Preference field value of 0 indicates that the BSS listed is an excluded BSS. The STA should refrain from associating to an AP corresponding to an excluded BSS. The Preference field values are used to establish the relative order of entries within the given list at the given time, and for the given AP.

The values between 1 and 255 provide the indication of order, with 255 indicating the most preferred BSS within the given candidate list, decreasing numbers representing decreasing preference relative only to entries with lower values of the Preference field, and equal numbers representing equal preference. The Preference field value is only valid before the Validity Interval has expired. The AP may include zero or more subelements in the BSS Transition Candidate List Entries field.

Upon successful reception of a BSS Transition Management Query frame or BSS Transition Response frame from a non-AP STA that contains a nonempty BSS Transition Candidate List Entries field, the AP should include at least one BSS candidate from that list with a nonzero Preference field value in the BSS Transition Candidate List Entries field of any subsequent BSS Transition Management Request frame with the Preferred Candidate List Included field set to 1 transmitted to the non-AP STA. The AP shall evaluate the BSSs indicated in the BSS Transition Candidate List Entries field in the latest BSS Transition Management Query frame or BSS Transition Management Response frame received from the non-AP STA as BSS transition candidate(s) for the non-AP STA. The means by which the AP evaluates and determines BSS transition candidates is outside the scope of this specification.

A non-AP STA that receives the Abridged bit with a value of 1 shall treat any BSSID in the current ESS that does not appear in the BSS Transition Candidate List as if it were present in the BSS Transition Candidate List with a Preference value of 0.

The AP may include one BSS Termination Duration subelement for each BSS in the BSS Transition Candidate List Entry field, including the BSS Termination Duration value and a BSS Termination TSF value. The BSS Termination Duration value may be different for each BSS.

When the AP transmits a BSS Transition Management Request frame with the Disassociation Imminent field set to 1 to a non-AP STA, the Disassociation Timer field in the BSS Transition Management Request frame shall be set to 0 or set to the number of TBTTs that will occur prior to the AP disassociating the non-AP STA. The AP shall start a timer for the non-AP STA with the initial timer value set to the Disassociation Timer field value. The AP shall decrement the timer by one after transmitting each Beacon frame until the timer has value of 0. In subsequent BSS Transition Management Request frames that the AP transmits to the non-AP STA, the Disassociation Timer field shall be set to the value of the timer.

If the most recent BSS Transition Management Request frame that an AP has transmitted to a non-AP STA has the Disassociation Imminent field set to 1, then the AP shall not transmit a Disassociation frame to the non-AP STA unless the timer for the non-AP STA has value of 0.

An AP's SME may have accounting session control information, such as a notice of session expiry. The means by which the AP's SME obtains accounting session control information is out of scope of this specification. Accounting session control information might include a time duration after which the non-AP STA will be disassociated from the ESS and an optional session information URL at which information may be obtained to extend the accounting session. When an AP's SME has accounting session control information, it shall issue an MLME-BTM.request to the AP's MLME and shall encode the time to session expiry in the Disassociation Timer parameter and shall encode the URL, if available, in the SessionInformationURL parameter. A non-AP STA's SME receiving an MLME-BTM.indication forwards the MLME-BTM.indication parameters to the appropriate entity within the device (e.g., web-browser) to inform the end-user; the means and protocol by which the SME accomplishes this is outside the scope of this specification.

A STA's SME receiving an MLME-BTM.indication primitive containing the BSS Transition Candidate List Entries field should use this list of candidates, with individual transition preference values, to make BSS transition decisions. Upon receiving an MLME-BTM.indication primitive, the STA's SME shall disregard any previous MLME-BTM.indication primitive received from the same AP. The STA shall not use the corresponding BSS Transition Candidate List Entries field information after the indicated Validity Interval. The STA may send a BSS Transition Management Query frame at any time to obtain an updated BSS Transition Candidate List Entries field or to indicate the STA's BSS transition candidates.

A STA's SME receiving an MLME-BTM.indication primitive containing a nonzero value of the Disassociation Timer field should attempt to find a suitable AP with which to reassociate before the indicated disassociation time.

10.23.6.4 BSS transition management response

When the STA's SME receives an MLME-BTM.indication primitive, it may issue an MLME-BTM.response primitive.

The STA's SME may include the result of its BSS transition decision in the Target BSSID field and Status Code field in the MLME-BTM.response primitive. A Status Code set to a value of 0 (i.e., Accept) indicates the STA will transition from the current BSS. The STA's SME receiving an MLME-BTM.indication primitive may issue an MLME-BTM.response primitive with a valid nonzero status code indicating rejection if it is unable to comply with this BSS transition management request.

When a non-AP STA's SME receives an MLME-BTM.indication primitive with the BSS Termination Included field in the Request Mode field equal to 1 it may issue an MLME-BTM.response primitive with the Status code set to one of the following values:

- 0 - Accept. Accept the BSS Termination request.
- 4 - Reject, BSS Termination Undesired. Request for deferral of BSS Termination, interval not specified.

- 5 - Reject, BSS Termination Delay. Request for deferral of BSS Termination interval specified in the BSS Termination Delay field in the BSS Transition Management Response frame.

The AP's SME may terminate or delay BSS Termination based on policies that are out of the scope of this standard. The MLME-RESET.request primitive is invoked to terminate the BSS. The AP shall disassociate all STAs immediately prior to termination of the BSS.

When a non-AP STA's SME receives an MLME-BTM.indication primitive with both the Disassociation Imminent and Preferred Candidate List Included fields equal to 0, the non-AP STA's SME shall issue a MLME-BTM.response primitive with the Status code set to one of the following values:

- 0 - Accept.
- 2 - Reject, Insufficient Beacon or Probe Response frames received from all candidates.
- 3 - Reject, Insufficient available capacity from all candidates.
- 6 - Reject, STA BSS Transition Candidate List provided.
- 7 - Reject, No suitable BSS transition candidates.
- 8 - Reject, Leaving ESS.

When an AP's SME receives an MLME-BTM.confirm primitive with the Status code field equal to a value of 2, indicating "Reject, Insufficient Beacon or Probe Response frames received from all candidates," the AP's SME should generate an MLME-BTM.request primitive with both the Disassociation Imminent and Preferred Candidate List Included fields set to 0 after providing sufficient time for the non-AP STA to perform its scanning procedures.

If the Status code field is a value of 6, indicating "Reject, STA BSS Transition Candidate List provided," the non-AP STA shall include a nonempty BSS Transition Candidate List Entries field in the BSS Transition Management Response frame to indicate the non-AP STA's transition preferences. The Status code field is a value of 8, indicating "Reject, Leaving ESS" if the non-AP STA intends to disassociate from the ESS.

The BSS Transition Candidate List Entries field of a BSS Transition Management Response frame contains zero or more Neighbor Report elements describing the non-AP STA's preferences for target BSS candidates. The Preference field value of a Neighbor Report element used in a BSS Transition Management Response frame shall be between 1 and 255. The value of 0 is reserved. The values between 1 and 255 provide the indication of order, with 255 indicating the most preferred BSS within the given candidate list, decreasing numbers representing decreasing preference relative only to entries with lower values of the Preference field, and equal numbers representing equal preference. The non-AP STA should not list any BSS that is not considered as a target BSS candidate.

When a non-AP STA receives a BSS Transition Management Request frame that has both the Disassociation Imminent and Preferred Candidate List Included fields equal to 1 and a nonempty BSS Transition Candidate List Entries field, if the non-AP STA transmits a BSS Transition Management Response frame to the AP with the Status Code field set to 0 (Accept), then the non-AP STA shall either disassociate from the AP or attempt to reassociate with an AP corresponding to one of the nonexcluded BSSs in the BSS Transition Candidate List Entries field of the received BSS Transition Management Request frame.

Prior to transitioning to an excluded BSS listed in the BSS Transition Candidate List Entries field of a received BSS Transition Management Request frame, the non-AP STA shall transmit a BSS Transition Management Response frame to the AP indicating the reject reason.

10.23.7 FMS multicast rate processing

An AP that supports FMS indicates its ability to deliver group addressed frames at alternate delivery intervals by its advertisement of the FMS capability. A STA that supports FMS includes the FMS Request element in FMS request frames to indicate a request to use the FMS service, including use of a higher multicast rate. The AP selects the multicast rate to use with the STA and indicates the rate and multicast address in the FMS Response element in the FMS Response frame. The AP shall not select a rate that is higher than the lowest rate value provided by the currently associated STAs requesting FMS service from the AP for the same FMS stream identified by FMSID.

The STA's SME may request membership in a multicast group or changes in multicast data rate by issuing an MLME-FMS.request primitive. Upon receipt of an FMS Request frame at the AP's SME as indicated by reception of the MLME-FMS.indication primitive the AP's SME shall issue an MLME-FMS.response primitive, indicating the FMS Request element, including the multicast address. The AP may send an FMS Response frame to the STA to change the STA's multicast rate. When the AP sends an FMS Response frame to the STA with an Element Status field value of 8, indicating "Alternate Preferred, due to AP multicast rate policy," the STA shall not send further FMS Request frames to request a change in the multicast rate while the STA is associated to the AP.

10.23.8 Collocated interference reporting

Collocated interference may cause degradation of IEEE 802.11 STA performance either periodically or continuously. Collocated interference reporting allows a requesting STA to receive information concerning the collocated interference being experienced by another STA that is operating on the same channel as the requesting STA. Such interference may be due to an interaction between radios where a reporting STA is collocated with another radio device. Collocated interference information might be used by the requesting STA to manage communication to the reporting STA such that the effect of the interference might be limited.

Implementation of Collocated interference reporting is optional for a WNM STA. A STA that implements Collocated Interference reporting has dot11MgmtOptionCoLocIntfReportingImplemented set to true. When dot11MgmtOptionCoLocIntfReportingImplemented is true, dot11WirelessManagementImplemented shall be true. A STA that has a value of true for dot11MgmtOptionCoLocIntfReportingActivated is defined as a STA that supports collocated interference reporting. A STA for which dot11MgmtOptionCoLocIntfReportingActivated is true shall set the Collocated Interference Reporting field of the Extended Capabilities element to 1.

A requesting STA may request that collocated interference reporting is enabled at another STA that has indicated support for the interference reporting capability. To enable collocated interference reporting, the STA shall send a Collocated Interference Request frame with Automatic Response Enabled bit set to the value representing the requested reporting type; see 8.5.14.13. A STA accepting a request for collocated interference reporting shall send the first report when it has knowledge of collocated interference.

Subsequently, a STA accepting a request with the Automatic Response Enabled subfield equal to 1 shall send a collocated interference report when the temporal characteristics of the interference or the level of the interference caused by collocated interferer significantly change to provide updated information, subject to meeting the Report Timeout requirement. A STA accepting a request with the Automatic Response Enabled subfield equal to 2 or 3 shall send the collocated interference reports periodically using the period included in the Report Period field in the Collocated Interference Reporting element in the report frames. In addition to sending reports periodically, a STA accepting a request with the Automatic Response Enabled filed equal to 3 shall send a collocated interference report when the temporal characteristics of the interference or the level of the interference caused by collocated interferer significantly change, subject to meeting the Report Timeout requirement.

The criteria a reporting STA uses for determining significant changes are internal to the reporting STA and outside the scope of this standard. When the reporting STA sends a collocated interference report, it shall restart the Report Period timer for periodic reporting. For either periodic reporting or reporting due to the changes in collocated interference, the reporting STA shall not generate collocated interference reports more frequently than indicated by the Report Timeout field in Interference Request field in the Collocated Interference Request frame.

The requesting STA may disable reporting by sending a Collocated Interference Request frame with the Automatic Response Enabled subfield set to 0. The collocated interference reporting shall be terminated on receipt of a Collocated Interference Request frame with the Automatic Response Enabled subfield equal to 0. All outstanding collocated interference requests are cancelled upon a BSS transition and/or channel switch. A new collocated interference request included in the new collocated Interference Request frame supersedes any previously received requests sent by the same STA.

A STA that supports collocated interference reporting may send a Collocated Interference Request frame to another STA that supports collocated interference reporting immediately after they are associated, so that the reporting STA can send a collocated interference report as soon as it has knowledge of collocated interference. The Dialog Token field is the nonzero value received in the corresponding Collocated Interference Request frame which was used to enable reporting.

The reporting STA shall use the Interference Index field in the Collocated Interference Report frame to indicate different types of interference. The reporting STA shall select unique Interference Index value for each collocated interference source. For example if the reporting STA has knowledge of collocated interference originating from two interference sources the reporting STA shall report both type of interference using separate Collocated Interference Report elements having separate Interference Index field. Both Collocated Interference Report elements can be sent in the same Collocated Interference Report frame and both can have the same report period. A report with the Interference Index field in the Collocated Interference Report element equal to 0 indicates that no collocated interference is present.

The characteristics of the interference are known a priori without requiring interference detection, measurement, and characterization by the IEEE 802.11 STA. The methods used by a reporting STA to know the periodicity, level of interference, accuracy of the reported interference level, interference center frequency and interference bandwidth are outside the scope of this standard.

10.23.9 QoS Traffic capability procedure

Implementation of the QoS Traffic capability is optional for a WNM STA. A STA that implements QoS Traffic capability has dot11MgmtOptionQoSTrafficCapabilityImplemented set to true. When dot11MgmtOptionQoSTrafficCapabilityImplemented is true, dot11WirelessManagementImplemented shall be true. A STA that has a value of true for dot11MgmtOptionQoSTrafficCapabilityActivated is defined as a STA that supports QoS Traffic Capability. A STA for which dot11MgmtOptionQoSTrafficCapabilityActivated is true shall set the QoS Traffic Capability field of the Extended Capabilities element to 1.

If dot11MgmtOptionQoSTrafficCapabilityActivated is true, a QoS AP may use the QoS Traffic Capability field values received from a non-AP QoS STA as one of the factors when determining association, reassociation, and the BSS transition of the STA. A non-AP STA with a value of true for dot11MgmtOptionQoSTrafficCapabilityActivated may send an Association Request, a Reassociation Request or QoS Traffic Capability Update frame to an AP whose last received Extended Capabilities element contained a value of 1 for the QoS Traffic Capability bit in the Capabilities field.

If dot11MgmtOptionQoSTrafficCapabilityActivated is true, a non-AP QoS shall construct the QoS Traffic Capability Flags as specified in 8.4.2.80 and 8.5.14.22. QoS Traffic Capability Flags are constructed at the SME of the non-AP QoS STA, from application requirements supplied to the SME. The QoS Traffic

Capability Flags are constructed from two application requirements: whether generation of traffic is required for applications and whether a specific UP is required for the generated traffic. If such requirements are supplied to the SME, the SME shall set the flag corresponding to the specific UP to 1.

NOTE—The requirements might be known before the traffic is actually generated. For example, a phone application might configured to generate UP 6 traffic upon the initiation of a voice session.

Unless application requirements for a specific UP are supplied to the SME, the SME shall set the flag corresponding to the UP to 0.

If dot11MgmtOptionQoSTrafficCapabilityActivated is true, a non-AP QoS STA shall include the QoS Traffic Capability element in an Association Request frame or in a Reassociation Request frame when it is sending such a frame to associate or reassociate with an AP. If there is any change in QoS Traffic Capability Flags while associated with an AP, the non-AP STA shall send a QoS Traffic Capability Update frame (see 8.5.14.22) including the updated QoS Traffic Capability Flags to the AP.

10.23.10 AC Station Count

Implementation of AC Station Count is optional for a WNM STA. A STA that implements AC Station Count has dot11MgmtOptionACStationCountImplemented set to true. When dot11MgmtOptionACStationCountImplemented is true, dot11WirelessManagementImplemented and dot11MgmtOptionQoSTrafficCapabilityImplemented shall be true. When dot11MgmtOptionACStationCountActivated is true, the STA shall set the AC Station Count field to 1 in the Extended Capabilities element to indicate that the STA supports the AC Station Count capability. When dot11MgmtOptionACStationCountActivated is false, the STA shall set the AC Station Count field in the Extended Capabilities element to 0 to indicate that the STA does not support this capability.

If dot11MgmtOptionACStationCountActivated is true, a QoS AP shall construct the QoS Traffic Capability Bitmask and AC STA Count list as specified in 8.4.2.80. The AP shall construct the STA Count List value based on the UP-to-AC mappings as defined in Table 9-1, the QoS Traffic Capability Bitmask/Flags of the non-AP STAs that are currently associated with it, and additional information. If dot11MgmtOptionACStationCountActivated is true, a QoS AP shall include the QoS Traffic Capability element in a Probe Response frame and in a Beacon frame.

If dot11MgmtOptionACStationCountActivated is true, a non-AP QoS STA may use the STA Count field values as one of the factors when determining association, reassociation, and the BSS transition. If dot11MgmtOptionACStationCountActivated is false, a non-AP QoS STA shall not use the STA Count field values as one of the factors when determining association, reassociation, and the BSS transition.

10.23.11 TFS procedures

10.23.11.1 TFS capability

Implementation of the TFS capability is optional for a WNM STA. A STA that implements TFS has dot11MgmtOptionTFSImplemented set to true. When dot11MgmtOptionTFSImplemented is true, dot11WirelessManagementImplemented shall be true. A STA that has a value of true for dot11MgmtOptionTFSActivated is defined as a STA that supports TFS. A STA for which dot11MgmtOptionTFSActivated is true shall set the TFS field of the Extended Capabilities element to 1.

A STA with a value of true for dot11MgmtOptionTFSActivated may send a TFS Request, TFS Response or TFS Notify frame to a STA within the same infrastructure BSS whose last received Extended Capabilities element contained a value of 1 for the TFS bit in the Capabilities field. The Traffic Filtering service is not supported in an IBSS.

A traffic filter is established using the TFS Request element and the patterns to be matched specified in one or more enclosed TFS subelements. A frame matches the traffic filter when at least one TCLAS based classifier matches the frame. Using multiple TFS subelements in a TFS Request element is the equivalent to a logical OR operation on the match conditions of each TFS subelement. Using multiple TCLAS elements in a TFS subelement is the equivalent to a logical AND operation on the match condition of each TCLAS element. An AP may propose an alternative TCLAS-based classifier by returning a TFS subelement in the TFS response (see 10.23.11.3).

When a traffic filter for group addressed frames is enabled at the AP, the group addressed frames are still delivered, without regard to the frames matching the traffic filter, since other associated STAs may also receive these frames. Because a STA using TFS can be in power save mode for an extended period of time, group addressed frames that match the traffic filter might be delivered before the STA is aware that the traffic filter has been matched. It is likely (but not guaranteed) that the STA does not receive those group addressed frames matching the traffic filter at the scheduled group addressed delivery time. To prevent this from happening, the STA can request a notification frame be sent when requesting the establishment of the traffic filter. If negotiated with the AP, the frames that do match at least one of the set of specified traffic filters are indicated to the non-AP STA via a notification frame.

10.23.11.2 TFS non-AP STA operation

To use the TFS, the non-AP STA's SME that supports TFS shall issue an MLME-TFS.request primitive to send a TFS Request frame. The MLME-TFS.request primitive shall include a valid TFSRequest parameter as defined in the TFS Request elements that the AP uses as triggers for the non-AP STA.

If the non-AP STA requests TFS Notify frames to be sent by the AP, the Notify bit field of the TFS Action code field shall be set to 1 in the TFS Request element.

The receipt of an MLME-TSF.confirm primitive with a valid TFSResponse parameter indicates to the STA's SME that the AP has processed the corresponding TFS request. The content of the TFSResponse parameter provides the status of each of the TFS elements processed by the AP. A TFSResponse parameter containing a TFS subelement may contain a modified TCLAS element having a Classifier Mask field equal to 0 or having one or more Classifier Parameter subfields equal to 0. If so, the non-AP STA shall interpret these (sub)fields to mean that no suggested value has been provided by the AP.

The non-AP STA may indicate that it is no longer using a particular TFS element by transmitting a TFS Request frame without that TFS element. The AP shall send a TFS Response frame with the Response element Status field value set to Accept, upon receipt of the TFS Request frame.

The non-AP STA may choose to terminate use of the TFS service by sending a TFS Request frame with no TFS elements in the request thereby canceling all traffic filters at the AP.

10.23.11.3 TFS AP operation

When an AP's SME receives an MLME-TFS.indication primitive with a valid TFSRequest parameter, it shall establish one or more traffic filters for the requesting STA and issue an MLME-TFS.response primitive with a TFSResponse parameter indicating the status of the associated request. When the AP establishes any filters for a requesting STA, the AP shall establish a traffic filter that matches individually addressed EAPOL-Key messages addressed to the requesting STA, with bits 0 and 1 of the TFS Action Code field set to 0.

When an AP's SME receives an MLME-TFS.indication primitive with a valid TFSRequest parameter having a requested TCLAS-based classifier which it is unable to provide, the SME shall issue an MLME-TFS.response primitive indicating the status of the corresponding request and may include a TFSResponse parameter having a suggested TCLAS-based classifier.

When TFS is enabled for an associated STA, the AP shall discard all individually addressed frames destined for the non-AP STA until a frame is found that matches one or more traffic filters established by the STA. When a frame is found that matches one or more of the traffic filters enabled at the STA (a matching frame), the AP shall perform the following actions, in order.

If bit 1 of the TFS Action Code field is set for any of the traffic filters that matched the matching frame, a TFS Notify frame shall be queued for transmission to the STA.

For an incoming individually addressed frame, the AP shall send the matching frame to the destination STA.

If bit 0 of the TFS Action Code field is set for a traffic filter that matched the matching frame, the AP shall delete the traffic filter.

NOTE—Due to the operation of group addressed frame delivery, a group addressed frame that matches a traffic filter might result in the STA receiving indication of the group addressed frame either before or after the group addressed frame is transmitted by the AP, if the TFS Notify frame is queued in the STA's power save queue. This might result in the STA receiving the group addressed frame in some cases and not receiving it in other cases.

Upon receiving an MLME-TFS.indication primitive, the AP's SME shall disregard any previous MLME-TFS.indication primitive received from the same STA.

The AP shall terminate any TFS operation for a STA when no traffic filters remain for a STA or if the AP's SME receives an MLME-TFS.indication primitive with a null TFSRequest.

10.23.12 BSS Max idle period management

If dot11MaxIdlePeriod is a nonzero, the STA shall include the BSS Max Idle Period element in the Association Response frame or the Reassociation Response frame. Otherwise, the STA shall not include the BSS Max Idle Period element in the Association Response frame or the Reassociation Response frame. STAs may send security protocol protected or unprotected keep-alive frames, as indicated in the Idle Options field.

If the Idle Options field requires security protocol protected keep-alive frames, then the AP shall disassociate the STA if no protected frames are received from the STA for a period of duration BSS Max idle period. If the Idle Options field allows unprotected or protected keep-alive frames, then the AP shall disassociate the STA if no protected or unprotected frames are received from the STA for a period of duration BSS Max idle period.

NOTE—The AP can disassociate or deauthenticate the STA at any time for other reasons even if the STA satisfies the keep-alive frame transmission requirements.

10.23.13 Proxy ARP (including Proxy Neighbor Discovery) service

Implementation of the Proxy ARP Service is optional for a WNM STA. A STA that implements the Proxy ARP Service has dot11MgmtOptionProxyARPImplemented set to true. When dot11MgmtOptionProxyARPImplemented is true, dot11WirelessManagementImplemented shall be true. When dot11MgmtOptionProxyARPActivated is true, the Proxy ARP Service bit in the Extended Capabilities field shall be set to 1 to indicate that the AP supports the Proxy ARP Service. When dot11MgmtOptionProxyARPActivated is false, the Proxy ARP Service bit shall be set to 0 to indicate that the AP does not support the Proxy ARP Service.

When the AP sets the Proxy ARP field to 1 in the Extended Capabilities element, the AP shall maintain a Hardware Address to Internet Address mapping for each associated station, and shall update the mapping when the Internet Address of the associated station changes. When the IPv4 address being resolved in the ARP request packet is used by a non-AP STA currently associated to the BSS, the Proxy ARP service shall respond to the request on behalf of the non-AP STA (IETF RFC 925).

When an AP receives an ARP Request from one associated STA or from the DS with a Target IP Address that corresponds to a second associated STA, the AP shall insert the second STA MAC address as the Sender's MAC Address in the ARP Response packet.

When an IPv6 address is being resolved, the Proxy Neighbor Discovery service shall respond to an Internet Control Message Protocol version 6 (ICMPv6) Neighbor Solicitation Message (Section 4.3, IETF RFC 4861) with a Neighbor Advertisement Message (Section 4.4, IETF RFC 4861) on behalf of an associated STA. When MAC address mappings change, the AP may send unsolicited Neighborhood Advertisement Messages on behalf of a STA.

10.23.14 Channel usage procedures

Channel Usage information is a set of channels provided by an AP to non-AP STAs for operation of a noninfrastructure network or an off-channel TDLS direct link. The Channel Usage information provided by the AP to the non-AP STA is to advise the STA on how to co-exist with the infrastructure network.

Implementation of Channel Usage is optional for a WNM STA. A STA that implements Channel Usage has dot11MgmtOptionChannelUsageImplemented set to true. When dot11MgmtOptionChannelUsageImplemented is true, dot11WirelessManagementImplemented shall be true. A STA that has a value of true for dot11MgmtOptionChannelUsageActivated is defined as a STA that supports Channel Usage. A STA for which dot11MgmtOptionChannelUsageActivated is true shall set the Channel Usage field of the Extended Capabilities element to 1.

A non-AP STA that supports Channel Usage and is not associated to an AP prior to using a noninfrastructure network or an off channel TDLS direct link may transmit a Probe Request frame including both Supported Operating Classes and Channel Usage elements. A non-AP STA supporting Channel Usage may send a Channel Usage Request frame at any time after association to the AP that supports the use of Channel Usage to request the Channel Usage information for supported operating classes.

Upon receipt of a Channel Usage element in the Probe Request frame, the AP supporting Channel Usage shall send a Probe Response frame including one or more Channel Usage elements. Upon receiving a Channel Usage Request frame, the AP supporting Channel Usage shall send a Channel Usage Response frame including one or more Channel Usage elements. Channel Usage elements shall only include channels that are valid for the regulatory domain in which the AP transmitting the element is operating and consistent with the Country element in the Beacon or Probe Response frame. When the Channel Usage element in a received Probe Request or Channel Usage Request frame includes one or more Operating Class/Channel Pair fields, the Operating Class/Channel Pair field(s) indicate(s) the requested non-AP STA operating class/ channels for the usage mode indicated in the frame.

The AP may send an unsolicited group addressed or individually addressed Channel Usage Response frame to the STAs that have requested Channel Usage information if the corresponding Channel Usage information needs to be updated. The Country element shall be included in the unsolicited and/or group addressed Channel Usage Response frame. The AP may include the Power Constraint information and EDCA Parameter in the Channel Usage Response frame. The values of the fields in the Power Constraint and EDCA Parameter Set elements included in the Channel Usage Response frame shall be the same values of the fields in the Power Constraint and EDCA Parameter Set elements that are transmitted by the AP.

Upon receipt of a Channel Usage element in the Probe Response or Channel Usage Response frame, the receiving STA may use the following:

- The channel usage information as part of channel selection processing to start a noninfrastructure network or an off-channel TDLS direct link
- The Power Constraint element, if present, as part of determining its maximum transmit power for transmissions for the noninfrastructure network or an off-channel TDLS direct link

— The EDCA Parameter Set element, if present, as part of determining its EDCA parameters for transmissions for the noninfrastructure network or an off-channel TDLS direct link

If either a recommended operating class, or a recommended channel, or both are not supported or understood by the recipient, or if the operating country of the sender is unknown, the recipient shall discard the corresponding channel usage recommendation. A STA that has not requested Channel Usage information shall discard an unsolicited group addressed Channel Usage Response frame.

10.23.15 DMS procedures

The Directed Multicast Service (DMS) is a service that may be provided by an AP to its associated non-AP STAs that support the DMS service, where the AP transmits group addressed MSDUs as individually addressed A-MSDUs.

Implementation of DMS is optional for a WNM STA. A STA that implements DMS has dot11MgmtOptionDMSImplemented set to true. When dot11MgmtOptionDMSImplemented is true, dot11WirelessManagementImplemented and dot11HighThroughputOptionImplemented shall be true. A STA that has a value of true for dot11MgmtOptionDMSActivated is defined as a STA that supports Directed Multicast. A STA for which dot11MgmtOptionDMSActivated is true shall set the DMS field of the Extended Capabilities element to 1.

A non-AP STA that supports DMS may request use of DMS of one or more flows by sending a DMS Request frame or Reassociation Request frame that includes a DMS Request element containing one or more DMS Descriptors with the Request Type field set to “Add” per flow. Each DMS Descriptor field in the DMS Request element identifies group addressed frames that shall be transmitted to the requesting non-AP STA as individually addressed frames in addition to the group address frame transmission. In the TCLAS element, the Classifier Type subfield shall be set to the value 0, 1, or 4, and the Destination Address or Destination IP Address subfield shall be set to the multicast address of the flow that the STA requests to receive as individually addressed frames. In the TSPEC element, the STA may define the characteristics and QoS expectations of the corresponding traffic flow.

Upon receipt of a DMS Request frame or Reassociation Request frame from a non-AP STA, the AP shall respond with a corresponding DMS Response frame or Reassociation Response frame. If the AP accepts a DMS request identified by a DMS Descriptor, the Response Type field of the corresponding DMS Status field in the DMS Response element shall be set to “Accept” and a nonzero DMSID is assigned. A Response Type value of “Deny” shall be set in the corresponding Response Type field of the DMS Status field in the DMS Response element when the AP denies a DMS request identified by a DMS Descriptor, and the DMSID shall be set to 0. If the Response Type field is set to “Accept” or “Denied,” then the TCLAS Elements, TCLAS Processing Element, TSPEC Element and Optional Subelements fields of a DMS Status field in a DMS Response element shall be copied from the respective TCLAS Elements, TCLAS Processing Element, TSPEC Element and Optional Subelements fields of the corresponding DMS request. When one or more STAs send a DMS request to an AP, containing a DMS descriptor with a set of TCLAS element and TCLAS processing elements that are identical irrespective of ordering to another successfully received DMS request that is not yet terminated, the AP shall assign the same DMSID as was assigned to the previous DMS request.

When the AP denies the DMS Request, it may suggest an alternative TCLAS-based classifier by including one or more TCLAS elements and an optional TCLAS Processing element. The AP may include fewer TCLAS elements in the DMS Response element than were present in the request; when the AP's response includes a single TCLAS element, it shall not include a TCLAS processing element. If the AP changes a TCLAS element’s Classifier Type field in the DMS Response element but is unable to suggest a value for the Classifier Mask field, it shall set that field to 0. If the AP changes a TCLAS element’s Classifier Type field or Classifier Mask field in the DMS Response element but is unable to suggest values for one or more Classifier Parameter subfields, it shall set those subfields to 0.

A non-AP STA receiving a DMS Response frame containing a modified TCLAS element having a Classifier Mask field equal to 0 or having one or more Classifier Parameter subfields equal to 0 shall interpret the values of 0 to mean that no suggested value has been provided by the AP.

If the requested DMS is accepted by the AP, the AP shall send subsequent group addressed MSDUs that match the frame classifier specified in the DMS Descriptors to the requesting STA as A-MSDU subframes within an individually addressed A-MSDU frame (see 8.3.2.2 and 9.11). The A-MSDU shall be formatted as specified in 8.3.2.2 which includes the A-MSDU subframe headers' DA address set to the multicast address for the corresponding MSDUs. The AP shall continue to transmit the matching frames as group addressed frames (see 9.3.6, and 10.2.1.16) if at least one associated STA has not requested DMS for these frames.

A non-AP STA may request modification of the traffic characteristics or attributes of one or more accepted DMS traffic flows by sending a DMS Request frame or Reassociation Request frame containing one or more DMS Descriptors with the Request Type set to "Change" and with the DMSIDs that identify the DMS traffic flows to be modified. If the Request Type of a DMS Descriptor is set to "Change," then the values of at least one of the TSPEC Element and Optional Subelement fields shall be different from those of the accepted DMS traffic flow corresponding to the DMSID.

If the AP accepts a DMS change request identified by a DMS Descriptor, the Response Type field of the corresponding DMS Status field in the DMS Response element shall be set to "Accept" and the DMSID shall be set to that of the DMS Descriptor. If the AP denies a DMS change request identified by a DMS Descriptor, the Response Type field of the corresponding DMS Status field in the DMS Response element shall be set to "Deny" and the DMSID shall be set to that of the DMS Descriptor. When the AP denies a DMS change request identified by a DMS Descriptor, the existing DMS traffic flow of the corresponding DMSID shall remain unchanged.

The non-AP STA may request removal of one or more accepted DMS traffic flows by sending a DMS Request frame or Reassociation Request frame that includes a DMS Request element containing one or more DMS Descriptors with the Request Type set to "Remove" and the DMSID field set to that the DMSID of the accepted DMS traffic flow to be removed. The DMS Length field in this DMS Descriptor is set to 1. The TLCAS Elements, TCLAS Processing Element TSPEC Element and Optional Subelements fields shall not be included in the DMS Descriptor if the Request Type is set to "Remove." The AP shall terminate individually addressed frame delivery for the requested group addressed frames identified by the DMSID for the requesting non-AP STA upon receipt of a DMS Request frame or Reassociation Request frame with the Request Type field equal to "Remove." The AP shall respond to the termination request by sending a DMS Response frame including the corresponding DMSID and a Response Type value of "Terminate" in the Response Type field of the corresponding DMS Status field. The DMS Length field in this DMS Status field is set to 3. The TLCAS Elements, TCLAS Processing Element, TSPEC Element and Optional Subelement fields shall not be included in the DMS Status field if the Response Type field is set to "Terminate."

The AP may send an unsolicited DMS Response frame at any time to cancel a granted DMS identified by the DMSID by including the DMSID and a Response Type value of "Terminate" in the DMS Status field. The AP may decide to reject a new DMS or cancel a granted DMS at any time based on network condition, for example the number of associated STAs and channel load.

The non-AP STA shall keep a list of group addresses for which the non-AP STA has requested DMS and that have been accepted by the AP. The requesting STA shall discard group addressed frames that match a group address in this list until the DMS has been terminated. When the DMS is terminated, and if the sequence number value is provided in the Last Sequence Control field in the DMS response frame, using the value of the Last Sequence Control field, the non-AP STA shall discard the group addressed frames that are the duplicates of the individually addressed frames.

NOTE—When the Last Sequence Control field in the DMS response frame is not supported at the AP (i.e., the sequence number value is not provided in the field), and a multicast MSDU that has sent using both individually addressed and group addressed frame transmission, termination of the DMS stream by the AP might result in a non-AP STA receiving undetectable duplicate MSDUs that are not filtered out by MAC

If the length of the DMS Descriptors exceeds 255 octets, then multiple DMS Request elements shall be included, each containing only those DMS Descriptors that are completely contained within 255 octets. If the length of the DMS status fields exceeds 255 octets, then multiple DMS Response elements shall be included, each containing only those DMS Status fields that are completely contained within the first 255 octets.

If the non-AP STA supports both DMS and FMS, the non-AP STA shall not request both services for the same group addressed frames simultaneously. The non-AP STA may request the different service (DMS vs. FMS) for different group addressed frames.

If the AP supports both DMS and TFS, the AP shall first apply TFS to the frame and then apply DMS.

10.23.16 WNM-Notification

Implementation of the WNM-Notification capability is optional for a WNM STA. A STA that implements the WNM-Notification capability has dot11MgmtOptionWNMNotificationImplemented set to true. When dot11MgmtOptionWNMNotificationImplemented is true, dot11WirelessManagementImplemented shall be true. A STA that has a value of true for dot11MgmtOptionWNMNotificationActivated is defined as a STA that supports WNM-Notification. A STA for which dot11MgmtOptionWNMNotificationActivated is true shall set the WNM-Notification Enabled field of the Extended Capabilities element to 1.

A STA that supports WNM-Notification reporting shall only send a WNM-Notification Request or Response frame to a STA within the same infrastructure BSS or the same IBSS whose last received Extended Capabilities element contained a value of 1 for the WNM-Notification bit in the Capabilities field.

A STA shall only transmit both the WNM-Notification Request frame and the WNM-Notification Response frame with an individually addressed destination address.

The WNM-Notification capability enables a STA to indicate management information, including information about its firmware to a peer STA. Use of the information provided is outside the scope of this standard. A non-AP STA that supports WNM-Notification and receives a WNM-Notification request frame with the Type field equal to 0 shall respond with a WNM-Notification Response frame with the Response Status field set to 0.

10.24 WLAN interworking with external networks procedures

10.24.1 General

This subclause describes the actions and the procedures that provide interworking capabilities between IEEE 802.11 infrastructure and external networks.

10.24.2 Interworking capabilities and information

STAs indicate their support for interworking service by setting the dot11InterworkingServiceActivated MIB variable to true. When dot11InterworkingServiceActivated is true, STAs include the Interworking element in Beacon and Probe Response frames, and non-AP STAs include the Interworking element in Probe Request frames.

When dot11InterworkingServiceActivated and dot11ExtendedChannelSwitchActivated are both true in an infrastructure BSS, the AP may provide its operating channel and operating class to an Interworked SSPN using the values from dot11OperatingClassesTable MIB entry.

In an infrastructure BSS, the Interworking element contains signaling for Homogeneous ESSs. The HESSID is a 6-octet MAC address that identifies the homogeneous ESS. The HESSID value shall be identical to one of the BSSIDs in the homogeneous ESS. Thus, it is a globally unique identifier that, in conjunction with the SSID, may be used to provide network identification for an SSPN.

NOTE—This standard assumes that the HESSID field in the Interworking element is administered consistently across all BSSs in a homogeneous ESS.

The Interworking element also provides an access network type in Beacon and Probe Response frames to assist the non-AP STA with network discovery and selection.

10.24.3 Interworking procedures: generic advertisement service (GAS)

This subclause describes the actions and procedures that are used to invoke GAS. GAS may be used to enable network selection for STAs when dot11InterworkingServiceActivated is true. GAS provides transport mechanisms for advertisement services while STAs are in the unassociated state as well as the associated state. This is accomplished via the use of Public Action management frames, which are Class-1 frames. GAS messages shall be transmitted using individually addressed Public Action frames. When management frame protection is negotiated, stations shall use individually addressed Protected Dual of Public Action frames instead of individually addressed Public Action frames.

A GAS message exchange may take place between two STAs; one STA transmits a GAS Query Request and the other STA transmits the GAS Query Response as described in 10.24.3.1. The Advertisement Protocol transported by the GAS is one of the query protocols in Table 8-175.

GAS shall be supported by a STA when dot11InterworkingServiceActivated is true. ANQP shall be supported by a STA when dot11InterworkingServiceActivated is true. Other advertisement protocols shall be supported when the corresponding dot11GASAdvertisementID is present.

STAs shall not transmit a GAS Query for any Advertisement Protocol unless that Advertisement Protocol ID is included in the Advertisement Protocol element in a Beacon or Probe response frame. The Advertisement Protocol element specifies the Advertisement Protocols that a STA may use to communicate with Advertisement Servers, which may be collocated with a STA or in an external network. The Advertisement Protocol identifies the query language used by the Advertisement Server. The GAS protocol, which is used to transport Queries and Query Responses, is transparent to the Advertisement Protocol.

10.24.3.1 GAS Protocol

10.24.3.1.1 General

The presence of the Interworking element in Beacon or Probe Response frames indicates support for the GAS protocol. The presence of the Advertisement Protocol element in Beacon or Probe Response frames indicates the Advertisement Protocol IDs supported in the BSS or IBSS. A STA transmits a GAS Query Request in a GAS Initial Request frame and the responding STA provides the GAS Query Response or information on how to receive the GAS Query Response in a GAS Initial Response frame. The GAS Query Response shall be delivered in a single GAS Initial Response frame or in one or more GAS Comeback Response frames; the GAS Query Response shall not be split between a GAS Initial Response frame and one or more GAS Comeback Response frames. The GAS message sequence diagrams are shown in Figure 10-24, Figure 10-25, and Figure 10-26.

Figure 10-24 describes the GAS message exchange sequence when dot11GASPauseForServerResponse is true and the GAS Query Response fits within one MMPDU.

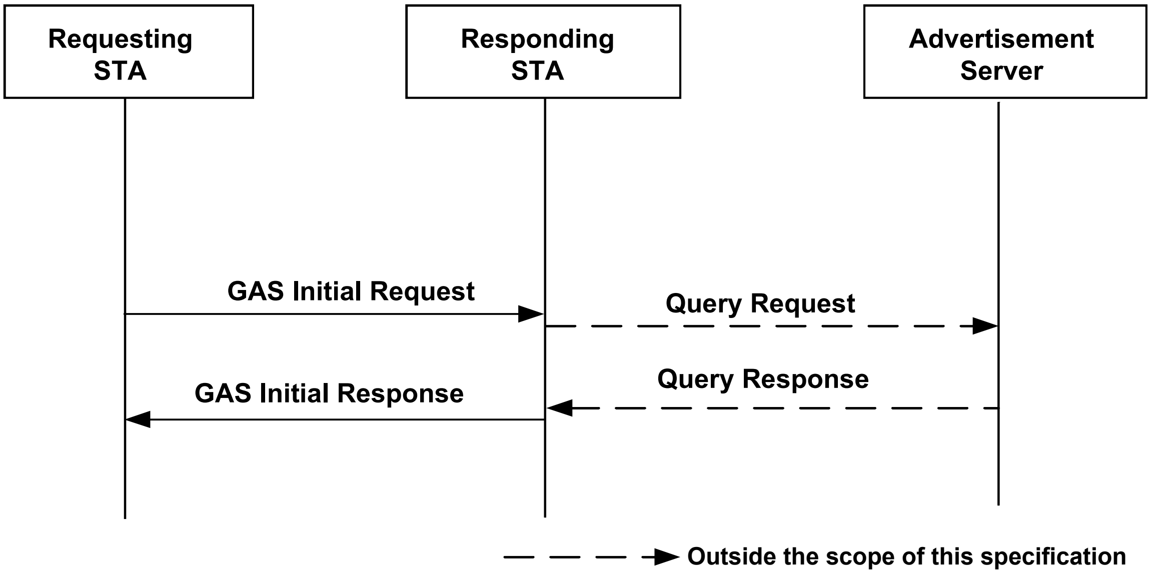


Figure 10-24—GAS message sequence with dot11GASPauseForServerResponse equal to true

Figure 10-25 describes the GAS message exchange sequence when dot11GASPauseForServerResponse is true and the GAS Query Response is too large to fit in one MMPDU and GAS fragmentation is used for delivery. The number of GAS Comeback Request and GAS Comeback Response messages depends on the number of GAS fragments required for delivery of the GAS Query Response.

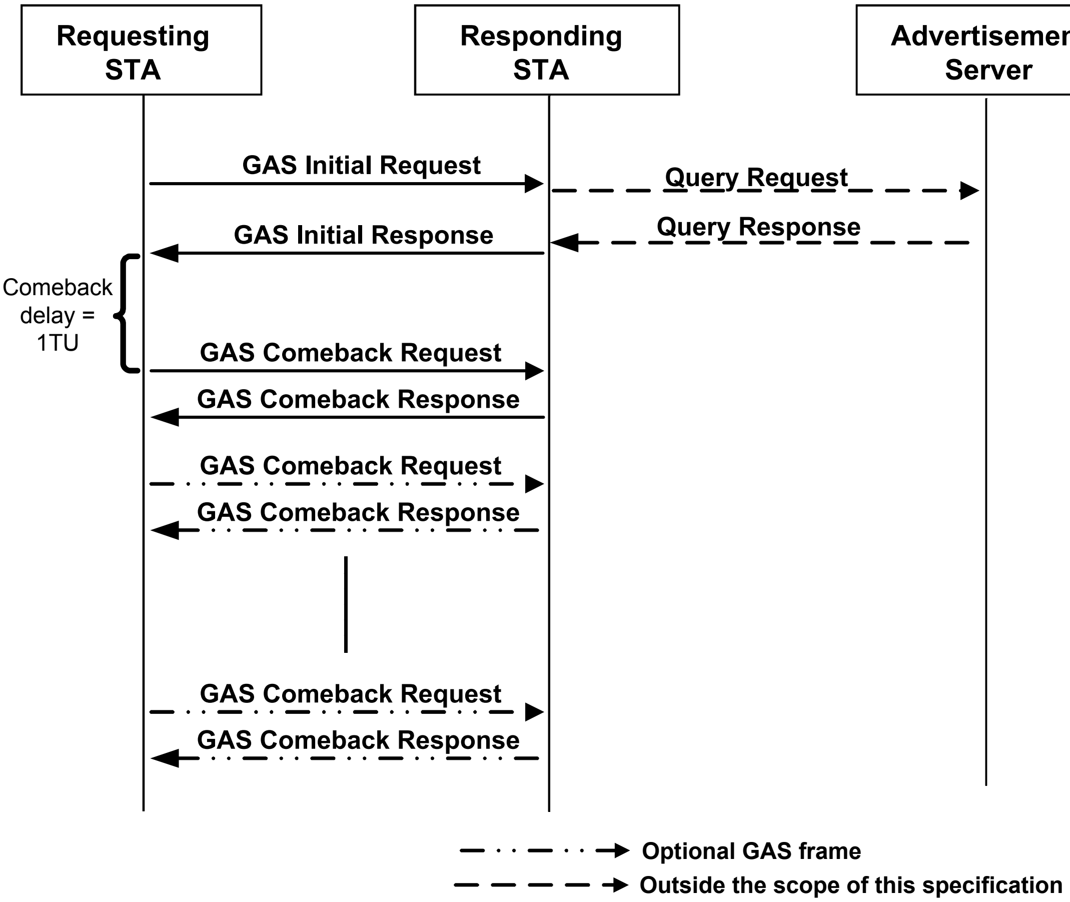


Figure 10-25—GAS message sequence with GAS fragmentation and dot11GASPauseForServerResponse equal to true

Figure 10-26 describes the GAS message exchange sequence when dot11GASPauseForServerResponse is false. The number of GAS Comeback Request and GAS Comeback Response messages depends on the number of GAS fragments required for delivery of the GAS Query Response.

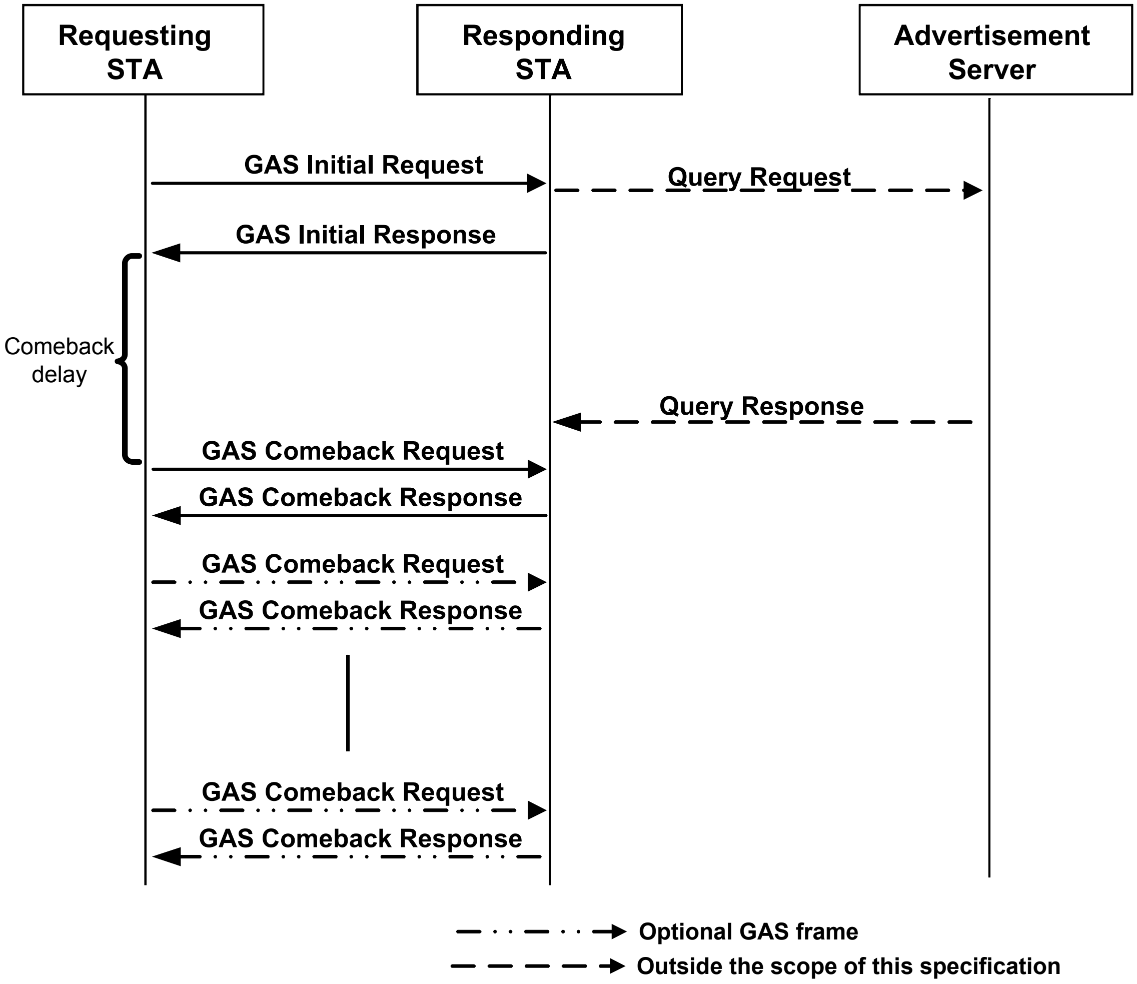


Figure 10-26—GAS message sequence with GAS fragmentation and dot11GASPauseForServerResponse equal to false

10.24.3.1.2 STA procedures to transmit a GAS Query

Upon receipt of an MLME-GAS.request primitive, the requesting STA shall engage in the following procedure to transmit a query:

a) The requesting STA sends a GAS Query by transmitting a GAS Initial Request frame containing a Dialog Token, an Advertisement Protocol element containing an Advertisement Protocol ID and the GAS Query in the Query Request field.

b) Upon transmission of the GAS Initial Request frame, the STA shall set a timer, referred to as the dot11GASResponseTimer, equal to dot11GASResponseTimeout or the QueryFailureTimeout parameter provided in the MLME-GAS.request primitive. If both values are present, the timer shall be set to the lesser of the two values.

c) If the requesting STA is not in the associated state, it shall remain in active mode until the receipt of a GAS Initial Response frame with the same Dialog Token as in the GAS Initial Request frame or until the expiry of the timer, whichever occurs first. If the requesting STA is in the associated state, it may go into power save state until the GAS Initial Response frame is available for receipt or the timer expiry, whichever occurs first.

d) If the dot11GASResponseTimer expires before a GAS Initial Response frame is received, the GAS Query was not successful and the MLME shall issue an MLME-GAS.confirm primitive indicating "timeout" and shall set the Query Response Length field to 0.

10.24.3.1.3 STA procedures to post a GAS Query to an Advertisement Server

Upon receipt of a GAS Initial Request frame, an MLME-GAS.indication primitive shall be issued to the STA's SME. Upon receipt of an MLME-GAS.response primitive, the STA shall transmit a GAS Initial Response frame to the requesting STA according to the following procedures. If the requesting STA is in the associated state and in the power save mode, the responding STA shall buffer the MMPDU for transmission according to the procedures in 10.2.1; otherwise the STA shall queue the MMPDU for transmission as follows:

a) If the Advertisement Protocol ID in the Advertisement Protocol element does not equal the value contained in any dot11GASAdvertisementID MIB object, then the STA shall not post the query to an Advertisement Server. The STA shall transmit an individually addressed GAS Initial Response frame to the requesting STA containing a dialog token whose value is identical to the dialog token in the GAS Initial Request frame, a Status Code equal to "GAS Advertisement Protocol not supported" (see Table 8-37), an Advertisement Protocol element containing the Advertisement Protocol ID used in the GAS Initial Request frame and a Comeback Delay and Query Response Length both set to 0.

b) If the query request corresponds to an Advertisement Protocol whose server is currently unreachable, the responding STA shall transmit an individually addressed GAS Initial Response frame to the requesting STA containing a dialog token whose value is identical to the dialog token in the GAS Initial Request frame, a Status Code equal to "Advertisement Server in the network is not currently reachable," an Advertisement Protocol element containing an Advertisement Protocol ID equal to the Advertisement Protocol ID contained in the GAS Initial Request frame and a Comeback Delay and Query Response Length both set to 0. The method used by the AP to determine the server is unreachable is out of scope of this specification. A STA receiving a status code indicating the Advertisement Server is unreachable should wait at least 1 minute before transmitting any further queries using the same Advertisement Protocol ID to the responding STA.

c) If the Advertisement Protocol ID in the Advertisement Protocol element equals the value contained in any dot11GASAdvertisementID MIB object, then the STA shall initialize a timer, referred to as the PostReplyTimer, to the value in dot11GASResponseTimeout and post the query to the Advertisement Server identified by the Advertisement Protocol ID. The methods and protocols the STA uses to post the query are outside the scope of this specification.

d) If dot11GASPauseForServerResponse is false, the responding STA shall transmit an individually addressed GAS Initial Response frame to the requesting STA containing a dialog token whose value is identical to the dialog token in the GAS Initial Request frame, a Status Code set to "success," an Advertisement Protocol element containing the Advertisement Protocol ID used in the GAS Initial Request frame, a GAS Comeback Delay set to the value in dot11GASComebackDelay for this Advertisement Protocol and a Query Response Length set to 0.

e) If dot11GASPauseForServerResponse is true, the GAS Query Response is delivered as defined in 10.24.3.1.4.

10.24.3.1.4 STA procedures for transmitting the GAS Query Response

After receiving a query response from the Advertisement Server, the responding STA shall buffer the query response for a minimum of dot11GASResponseBufferingTime after the expiry of the GAS Comeback Delay or until the query response is delivered. If the responding STA does not receive a GAS Comeback Request frame whose source MAC address and Dialog Token match the source MAC address and Dialog Token respectively of the corresponding GAS Initial Response frame within this time, it may drop the query response. If the query response is larger than the configured Query Response Length Limit, the responding STA shall discard the response and instead return a status code of "GAS Query Response larger than query response length limit" in the GAS Comeback Response frame. This behavior helps to prevent abuses of the medium that may be caused by overly general queries, which evoke a very large query response.

The GAS protocol supports Query Responses whose length is greater than the IEEE 802.11 maximum MMPDU size by the STA's use of the GAS Query Response Fragment ID field in the GAS Comeback Response frame; the Query Response Fragment ID shall be set to 0 for the initial fragment and incremented by 1 for each subsequent fragment in a multi-fragment query response. If the Query Response is a multi-fragment response (i.e., contains more than 1 fragment), the STA shall transmit all fragments that belong to the same Query Response until all fragments are exhausted. The STA shall set the More GAS Fragments field of the GAS Query Response Fragment ID to 0 when the transmitted fragment is the final fragment.

The following procedures shall be used by the responding STA to deliver the query response to the requesting STA:

a) If dot11GASPauseForServerResponse is true:
   1) If the PostReplyTimer expires before the GAS Query Response is received from the Advertisement Server, then the responding STA shall transmit an individually addressed GAS Initial Response frame to the requesting STA containing a dialog token whose value is identical to the dialog token in the GAS Initial Request frame, a Status Code set to GAS_QUERY_TIMEOUT (see Table 8-37), an Advertisement Protocol element containing the Advertisement Protocol ID used in the GAS Initial Request frame, a GAS Comeback Delay set to 0 and a Query Response Length set to 0. If the query response is subsequently received from the Advertisement Server, it shall be dropped by the responding STA.
   2) If the Query Response received from the Advertisement Server is larger than dot11GASQueryResponseLengthLimit or requires more than 128 fragments for transmission to the requesting STA, it shall be dropped by the responding STA. Then the responding STA shall transmit an individually addressed GAS Initial Response frame to the requesting STA containing a dialog token whose value is identical to the dialog token in the GAS Initial Request frame, a Status Code set to "Query Response too large" (see Table 8-37), an Advertisement Protocol element containing the Advertisement Protocol ID used in the GAS Initial Request frame, a GAS Comeback Delay set to 0 and a Query Response Length set to 0.
   3) If the query response's length is equal to or less than the maximum MMPDU size, the STA shall transmit an individually addressed GAS Initial Response frame to the requesting STA containing a dialog token whose value is identical to the dialog token in the GAS Initial Request frame, a Status Code set to "success," an Advertisement Protocol element containing the Advertisement Protocol ID used in the GAS Initial Request frame, a GAS Comeback Delay set to 0, the Query Response and a Query Response Length set to the query response length. This completes the GAS Query and GAS Query Response exchange.
   4) If the query response's length is larger than the maximum MMPDU size, the responding STA shall transmit an individually addressed GAS Initial Response frame to the requesting STA containing a dialog token whose value is identical to the dialog token in the GAS Initial Request frame, a Status Code set to "success," an Advertisement Protocol element containing the Advertisement Protocol ID used in the GAS Initial Request frame, a GAS Comeback Delay set to 1 TU, and a Query Response Length set to 0; this indicates the query response will be

transmitted using GAS Comeback Request and Response frames that support GAS fragmentation as follows.

b) If dot11GASPauseForServerResponse is false:

  1) If the PostReplyTimer expires before the GAS Query Response is received from the Advertisement Server then the responding STA shall buffer for transmission a GAS Comeback Response frame with a status code equal to GAS_QUERY_TIMEOUT (see Table 8-37). If the query response is subsequently received from the Advertisement Server, it shall be dropped by the STA.

  2) If the Query Response received from the Advertisement Server is larger than dot11GASQueryResponseLengthLimit, it shall be dropped by the responding STA. Then the STA shall buffer for transmission a GAS Comeback Response frame with status code set to "Query Response too large."

c) If the Query Response is received before the expiry of the PostReplyTimer and its length is less than dot11GASQueryResponseLengthLimit, then the Query Response shall be buffered in one or more GAS Comeback Response frames with status code set to "success." The responding STA transmits one GAS Comeback Response frame in response to each GAS Comeback Request frame. If the Query Response received from the Advertisement Server is less than or equal to the maximum MMPDU payload size, then the GAS Query Response Fragment ID shall be set to 0 and the More GAS Fragments field in the GAS Query Response Fragment ID shall be set to 0. If the Query Response received from the Advertisement Server is greater than the maximum MMPDU payload size, then the GAS Query Response Fragment ID shall be set to 0 if this is the first fragment of the Query Response transmitted; otherwise it shall be incremented by 1; the More GAS Fragments field in the GAS Query Response Fragment ID shall be set to 1 if there are more fragments of the Query Response to be transmitted; otherwise it shall be set to 0 (i.e., this fragment is the last fragment of the Query Response).

d) If a responding STA receives a GAS Comeback Request frame whose source MAC address and Dialog Token match the destination MAC address and Dialog Token respectively of an outstanding GAS Initial Response frame and the query response has not been received from the Advertisement Server and the PostReplyTimer has not expired, the responding STA shall transmit a GAS Comeback Response frame with status equal to "Response not received from server" (see Table 8-37) and GAS Comeback Delay set to the value in dot11GasComebackDelay for this Advertisement Protocol to indicate when the requesting STA should comeback to obtain its Query Response.

e) If a responding STA receives a GAS Comeback Request frame whose source MAC address and Dialog Token do not match the destination MAC address and Dialog Token respectively of an outstanding GAS Initial Response frame, the STA should transmit a GAS Comeback Response frame with a status code equal to "No request outstanding."

A requesting STA shall transmit a GAS Comeback Request frame including the Dialog Token (drawn from the corresponding GAS Initial Response frame) immediately after the expiry of the GAS Comeback Delay. In response, the responding STA provides the Query Response in one or more GAS Comeback Response frames with the corresponding Dialog Token.

If a requesting STA receives a GAS Comeback Response frame with status equal to "Query response outstanding," the requesting STA shall wait for the GAS Comeback Delay from that frame and upon expiry of the GAS Comeback Delay, transmit another GAS Comeback Request frame. If the requesting STA's dot11GASResponseTimer (set in 10.24.3.1.2 step b) expires prior to receiving a GAS Comeback Response frame whose source MAC address and Dialog Token match those in the corresponding GAS Initial Response frame, the STA shall issue an MLME-GAS.confirm primitive with result code set to "timeout" and shall set the Query Response Length to 0.

If a requesting STA receives a GAS Comeback Response frame with status equal to "success" and the More GAS Fragments field in the GAS Query Response Fragment ID equal to 1, it shall transmit another GAS Comeback Request frame in order to retrieve the next GAS fragment of a multi-fragment query response.

If a requesting STA receives a GAS Comeback Response frame with status equal to "success" and the More GAS Fragments field in the GAS Query Response Fragment ID equal to 0, the requesting STA's MLME shall determine that all fragments have been received by confirming that all fragment IDs from 0 to the value in the GAS Query Response Fragment ID when the More GAS Fragments field was equal to 0 have been received. Upon receipt of the first GAS Comeback Response frame and every GAS Comeback Response frame thereafter, the dot11GASResponseTimer shall be reset. If all of the query response fragments were received before the expiry of the dot11GASResponseTimer, then the MLME shall issue an MLME-GAS.confirm with result code set to "success" along with the query response. If all of the query response fragments were not received before the expiry of the dot11GASResponseTimer, then the MLME shall issue an MLME-GAS.confirm with result code set to "transmission failure" and shall set the Query Response Length to 0.

After a requesting STA receives the first GAS fragment of a multi-fragment query response, it shall continue retrieving the query response until all GAS fragments are received or until a transmission failure is detected; the requesting STA shall not commence the retrieval of a another GAS Query Response from the same STA until all GAS fragments are received or until a transmission failure is detected on the first GAS Query Response.

If a requesting STA receives a GAS Comeback Response with status equal to "Timeout" or "Query Response too large," then the MLME shall issue an MLME-GAS.confirm with result code so indicating and shall set the Query Response Length to 0.

If a requesting STA receives a GAS Comeback Response with status equal to "No outstanding GAS request," then the MLME shall issue an MLME-GAS.confirm with result code set to "unspecified failure" and shall set the Query Response Length to 0.

10.24.3.1.5 GAS procedures interaction with Multiple BSSID Set

Non-AP STAs in the unassociated state may use GAS procedures to query Advertisement Servers for information. As described in 10.24.3.1, APs indicate their support for a particular GAS Advertisement Protocol by including an Advertisement protocol element with that Advertisement protocol ID in Beacon and Probe Response frames as described in 8.3.3.2 and 8.3.3.10 respectively. Non-AP STAs receiving Beacon or Probe Response frames from different APs may choose to engage in GAS frame exchange sequences with one or more of these APs. In some deployment scenarios, these APs may be operating as a Multiple BSSID set (as defined in 10.11.14) and may relay the GAS queries to the same Advertisement Server. Depending on the configuration of the IEEE 802.11 access network, the external network and the Advertisement Server, a query response from the Advertisement Server may or may not be dependent on the BSSID used in the GAS frame exchange sequence and thus the STA from which the query was relayed. If the GAS Query Response is dependent on the BSSID, a requesting STA may choose to post queries using GAS procedures to more than one STA and expect possibly different Query Responses. If the Query Response is not dependent on the BSSID, then a requesting STA may choose to post queries using GAS procedures to only one STA in the Multiple BSSID set (i.e., posting the same query to another member of the Multiple BSSID set would yield the same response).

When a Multiple BSSID (as defined in 10.11.14) set contains two or more members and dot11InterworkingServiceActivated is true and dot11GASAdvertisementID is present and a query to the Advertisement Server corresponding to the value of dot11GASAdvertisementID is not dependent on the BSSID value used in the GAS frame exchange sequence to post the query, then the PAME-BI bit in the Advertisement Protocol tuple of the Advertisement Protocol element corresponding to the value of dot11GASAdvertisementID shall be set to 1; otherwise this bit shall be set to 0.

10.24.3.2 ANQP procedures

10.24.3.2.1 General

A STA may use ANQP to retrieve information as defined in Table 8-184 from a peer STA.

The ANQP query request uses the Query List ANQP-element comprised of ANQP-elements Info IDs from Table 8-184 that have an ANQP-element type of S in Table 10-10. The ANQP query request is transported in the Query Requst field of GAS Request frames as per 10.24.3.1.4. The ANQP query response is transported in the Query Response field of GAS Response frames, as per 10.24.3.1.4.

Table 10-10—ANQP usage

			BSS		IBSS
ANQP-element name	**ANQP-element (subclause)**	**ANQP-element type**	**AP**	**Non-AP STA**	**STA**
Query List	8.4.4.2	Q	T, R	T, R	T, R
Capability List	8.4.4.3	S	T, R	T, R	T, R
Venue Name	8.4.4.4	S	T	R	—
Emergency Call Number	8.4.4.5	S	T	R	—
Network Authentication Type	8.4.4.6	S	T	R	—
Roaming Consortium	8.4.4.7	S	T	R	—
Vendor Specific	8.4.4.8	Q, S	T, R	T, R	T, R
IP Address Type Availability	8.4.4.9	S	T, R	T, R	T, R
NAI Realm	8.4.4.10	S	T	R	T, R
3GPP Cellular Network	8.4.4.11	S	T	R	—
AP Geospatial Location	8.4.4.12	S	T	R	T, R
AP Civic Location	8.4.4.13	S	T	R	T, R
AP Location Public Identifier URI	8.4.4.14	S	T	R	T, R
Domain Name	8.4.4.15	S	T	R	—
Emergency Alert Identifier URI	8.4.4.16	S	T	R	T, R
TDLS Capability	8.4.4.18	Q, S	T,R	T,R	T, R
Emergency NAI	8.4.4.17	S	T	R	—
Neighbor Report	8.4.4.19	S	T	R	—

Symbols

Q	element is an ANQP query
S	element is an ANQP response
T	ANQP-element may be transmitted by MAC entity
R	ANQP-element may be received by MAC entity
—	ANQP-element is neither transmitted nor received by MAC entity

ANQP usage for infrastructure BSSs and IBSSs shall be in accordance with Table 10-10. ANQP usage defines the entities permitted to transmit and receive particular ANQP-elements.

A STA having dot11InterworkingServiceActivated equal to true shall be capable of using the Query List ANQP-element to request the Capability List ANQP-element; support for all other ANQP-elements is optional.

A STA that encounters an unknown or reserved ANQP Info ID value in a GAS frame (see 8-210) received without error shall ignore that ANQP Info ID and shall parse any remaining ANQP Info IDs.

A STA that encounters an unknown vendor-specific OI field or subfield in a GAS frame (see 8-210) received without error shall ignore that field or subfield respectively, and shall parse any remaining fields or subfields for additional information with recognizable field or subfield values.

10.24.3.2.2 Query List procedure

The Query List ANQP-element is used by a requesting STA to perform an ANQP Query using the procedures defined in 10.24.3.2.1. The requesting STA shall only include Info IDs in the Query List ANQP-element that have the sole ANQP-element type of S as shown in Table 10-10. Info IDs that have an ANQP-element type of Q shall not be included in the Query List ANQP-element (e.g., the Info ID for Vendor Specific ANQP-element shall not be included).

A responding STA that encounters an unknown or reserved ANQP Info ID value in an Query List ANQP-element received without error shall ignore that ANQP Info ID and shall parse any remaining ANQP Info IDs.

10.24.3.2.3 Roaming Consortium procedure

The Roaming Consortium ANQP-element, which contains a set of OIs, can be retrieved from an AP by a non-AP STA using the ANQP Query List procedures defined in 10.24.3.2.2. The list of OIs included in the Roaming Consortium ANQP-element shall be those OIs in the dot11RoamingConsortiumTable. An AP shall only include an OI in the dot11RoamingConsortiumTable, if in conjunction with an AS, it is capable of successfully authenticating a non-AP STA having valid security credentials for the SSPN identified by that OI. Methods used by the AP to authenticate the non-AP STA include, but are not limited to, RSNA algorithms and Open System authentication.

Each OI identifies an SSP or group of SSPs (i.e., a roaming consortium). An SSP or group of SSPs can register for and obtain an OI using the procedures defined in [B19].

A non-AP STA might have a locally stored binding between an OI and a set of security credentials with which it can authenticate to the network identified by the OI, that is, the SSPN. The method by which this binding is obtained is outside the scope of this standard. A non-AP STA can select from that list of credentials when authenticating to the BSS.

10.24.3.2.4 AP procedure for advertising EAP Method associated with an NAI Realm

When dot11RSNAActivated is true, NAI realms along with their supported authentication methods may be advertised using the NAI Realm ANQP-element (see 8.4.4.10). Each realm may be optionally associated with a set of EAP methods. Each EAP method may be optionally associated with a set of Authentication Parameters. The NAI realm ANQP-element provides a hint on the methods a STA might use to establish an association in an RSN IEEE 802.1X environment. If the non-AP STA recognizes the NAI realm, it may attempt authentication even if it believes the EAP methods are incorrect.

When dot11RSNAActivated is false and the Network Authentication Type (see 8.4.4.6) contains a Network Authentication Type Unit having a Network Authentication Type Indicator field equal to http/https redirection or DNS redirection, NAI realms without supported authentication methods may be advertised using the NAI Realm ANQP-element (see 8.4.4.10).

A non-AP STA having dot11InterworkingServiceActivated equal to true may process the NAI Realm ANQP-element. The selection of the NAI realm the non-AP STA uses for authentication is outside the scope of this standard. A non-AP STA requests the NAI Realm ANQP-element using Query List procedures defined in 10.24.3.2.2.

A non-AP STA having dot11InterworkingServiceActivated equal to true may optionally process the EAP Method list as follows:

- The EAP Method list provided by the AP shall be in priority order (the most preferred EAP Method is listed first).
- The credential types help the STA to determine what credentials to use for authentication.
- The STA should confirm the GAS advertisement after an RSNA is established by performing a GAS Query for the NAI Realm ANQP-element using Protected Dual of Public Action frames.

NOTE—The advertisements should be confirmed after the RSNA is established to avoid downgrade attacks.

The policy that determines whether a non-AP STA should attempt authentication and/or association with any particular IEEE 802.11 Access Network is outside the scope of this standard.

10.24.3.2.5 3GPP Cellular Network procedure

A STA may retrieve 3GPP Cellular Network information using the Query List procedure in 10.24.3.2.2. Realms referenced in the 3GPP Cellular Network ANQP-element, should not be included in the NAI Realm ANQP-element (see 8.4.4.10).

10.24.3.2.6 AP Geospatial Location procedure

A STA may retrieve an AP's Geospatial location using the Query List procedure in 10.24.3.2.2. A STA in the associated state should retrieve Geospatial location information from the AP using the procedures in 10.11.9.6.

10.24.3.2.7 AP Civic Location procedure

A STA may retrieve an AP's Civic location using the Query List procedure in 10.24.3.2.2. A STA in the associated state should retrieve Civic location information from the AP using the procedures in 10.11.9.9.

10.24.3.2.8 AP Location Public identifier URI procedures

A STA when dot11InterworkingServiceActivated is true may retrieve an AP's Location Public identifier URI using ANQP procedures in 10.24.3.2. A STA in the associated state should retrieve Location Public identifier URI information from the AP using the procedures in 10.11.9.10.

Due to security concerns, there are some URI schemes that should be cautiously processed when received by a STA. For example, URIs using the scheme names "data:" and "http:" may direct applications (e.g. a browser) on the STA to internet pages that contain active scripts. Therefore, URIs received via this ANQP procedure should not be processed in a general manner, as these scripts may be inadvertently activated. Instead of listing all the types of URIs that might be misused or potentially have harmful affects, Section 3.3 IANA registers acceptable URI schemes.

10.24.3.2.9 Emergency NAI procedure

A STA that does not have valid credentials to authenticate to a network may use the information within the Emergency NAI ANQP-element as its EAP identity.

The Emergency NAI ANQP-element can be retrieved from the AP using the Query List procedure in 10.24.3.2.2.

The STA uses the Emergency NAI to indicate its intention to access the network without peer authentication by using the Emergency NAI Information as its identity in the authentication process, as described in IETF RFC 5216.

10.24.3.2.10 TDLS Capability procedure

A STA may use the TDLS Capability ANQP-element (see 8.4.4.18) to discover TDLS capabilities of another TDLS capable STA using the Query List procedure in 10.24.3.2.2.

An example of TDLS capability discovery using ANQP is given in Figure 10-27.

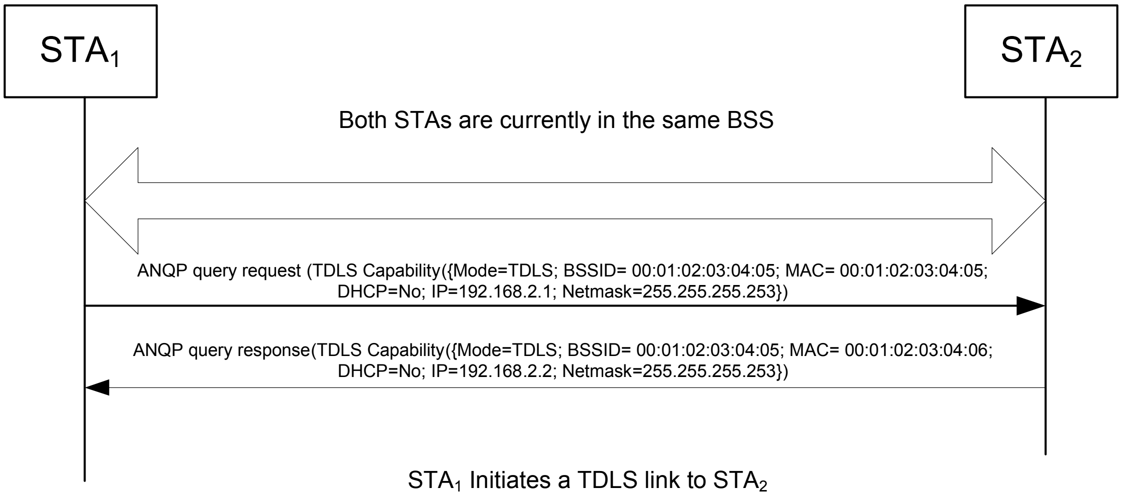

STA$_1$
STA$_2$
Both STAs are currently in the same BSS
ANQP query request (TDLS Capability({Mode=TDLS; BSSID= 00:01:02:03:04:05; MAC= 00:01:02:03:04:05; DHCP=No; IP=192.168.2.1; Netmask=255.255.255.253})
ANQP query response(TDLS Capability({Mode=TDLS; BSSID= 00:01:02:03:04:05; MAC= 00:01:02:03:04:06; DHCP=No; IP=192.168.2.2; Netmask=255.255.255.253})
STA$_1$ Initiates a TDLS link to STA$_2$

Figure 10-27—Example TDLS Capability discovery using ANQP

The mechanism shall work as follows:

a) STA1 determines the MAC address of STA2.

b) STA1 sends an ANQP query request to STA2. The ANQP query request includes a TDLS Capability ANQP-element.

c) STA2 receives the ANQP query request and updates the TDLS Capability ANQP-element based on its own capabilities that is returned in the ANQP query response to STA1.

10.24.4 Interworking procedures: IEEE 802.21 MIH support

IEEE Std 802.21-2008, the “MIH standard,” supports handovers across heterogeneous networks. STAs with dot11InterworkingServiceActivated equal to true and dot11GasAdvertisementId equal to MIH Information Service (see Table 8-175) shall support the transmission and reception of IEEE 802.21 MIIS queries for STAs in all states. STAs with dot11InterworkingServiceActivated equal to true and dot11GasAdvertisementId equal to MIH Command and Event Services Capability Discovery (see Table 8-175) shall provide support for IEEE 802.21 MICS/MIES capability discovery for non-AP STAs in all states.

Additionally, support for IEEE 802.21 MIIS query and IEEE 802.21 MICS/MIES capability discovery to non-AP STA's in the associated state is provided by the STA forwarding IP datagrams destined for the MIH point of service to the IEEE 802.21 MIIS server.

A non-AP STA discovers support for these services by receiving Beacon or Probe Response frames with an Advertisement Protocol element having Advertisement Protocol ID(s) for MIH Information Service and/or IEEE 802.21 MICS/MIES capability discovery.

A non-AP STA forms an IEEE 802.21 IS query by creating its query request according to the procedures defined in IEEE Std 802.21-2008 and formatting that request into an IEEE 802.21 MIH protocol frame as defined in 8.4 of IEEE Std 802.21-2008. The non-AP STA, using the procedures in 10.24.3.1, posts the query to an IEEE 802.21 IS server by transmitting the MIH formatted frame in the Query request field of a GAS Initial Request frame. The Advertisement Protocol ID field in the GAS Initial Request frame is set to the value of IEEE 802.21 MIH Information Service (Table 8-175).

Non-AP STAs in the unauthenticated or unassociated or associated states can use GAS procedures to discover MIH Command and Event Services Capability as specified in Table 8-175.

A non-AP STA forms an IEEE 802.21 MIH Command and Event Service discovery request by encapsulating an MIH_Capability_Discover request (see IEEE 802.21-2008) into an MIH protocol frame as defined in 8.4 of IEEE Std 802.21-2008. The non-AP STA, using the procedures in 10.24.3.1, posts the discovery request to the network by transmitting the MIH formatted frame in the Query request field of a GAS Initial Request frame. The Advertisement Protocol ID field in the GAS Initial Request frame is set to the value of MIH Command and Event Services Capability Discovery (Table 8-175). The method by which the AP relays the discovery request to the network is defined in IEEE Std 802.21-2008 and is outside the scope of this specification.

A non-AP STA retrieves the IEEE 802.21 MIH Command and Event Service discovery response according to the procedures in 10.24.3.1. The discovery response is an MIH protocol frame as defined in 8.4 of IEEE Std 802.21-2008.

10.24.5 Interworking procedures: interactions with SSPN

10.24.5.1 General operation

To provide SSPN Interface services, the IEEE 802.11 network interacts with the SSPN corresponding to the user of the non-AP STA either directly or via a roaming relationship. As part of setting up the RSN security association, user policies are communicated to the AP. If dot11SSPNInterfaceActivated is true, these permissions shall be stored in the AP's dot11InterworkingTableEntry for that STA. Thereafter, the AP shall use the dot11InterworkingTableEntry for controlling the service provision to that non-AP STA. User policies from the SSPN affect authentication, authorization, and admission control decisions at the AP. In addition, the AP collects statistics about the non-AP STA and reports the statistics to the SSPN when requested. The SSPN may also send service provision instructions to the AP, e.g., to terminate the connection to a non-AP STA. Non-AP STAs do not support the SSPN Interface.

Network deployments typically provide that the AP and the server in the SSPN have a trustworthy channel that can be used to exchange information, without exposure to or influence by any intermediate parties. The establishment of this secure connection between the IEEE 802.11 infrastructure and the SSPN is outside the scope of this standard.

10.24.5.2 Authentication and cipher suites selection with SSPN

When the non-AP STA initiates IEEE 802.1X authentication, the EAP messages are forwarded to the SSPN based on the home realm information provided by the non-AP STA. If the IEEE 802.11 infrastructure is unable to forward the EAP message, the AP when dot11SSPNInterfaceActivated is true shall disassociate the non-AP STA with Reason Code "Disassociated because lack of SSP roaming agreement to SSPN."

In addition to the EAP messages, the IEEE 802.11 infrastructure also provides extra information regarding the non-AP STA to the SSPN as defined in V.4.2, e.g., the Cipher Suite supported by non-AP STA, the location of the AP to which the non-AP STA is associated, etc. Such information may be used by the SSPN to make authentication and service provisioning decisions.

In the SSPN Interface Service, the SSPN uses more information than is carried over EAP to decide on the authentication result. The SSPN might reject a connection request if the cipher suites supported by non-AP STA does not meet its security requirements. In this situation, the SME of the AP when dot11SSPNInterfaceActivated is true shall invoke a disassociation procedure as defined in 10.3.5.8 by issuing the MLME-DISASSOCIATE.request primitive. The AP disassociates the corresponding non-AP STA with Reason Code "Requested service rejected because of SSPN cipher suite requirement."

The SSPN might reject the association request based on the location of the non-AP STA, e.g., if the non-AP STA is requesting association to an AP or associated to an AP located in a forbidden zone. In this situation, the SME of the AP when dot11SSPNInterfaceActivated is true shall invoke a disassociation procedure as defined in 10.3.5.8 by issuing the MLME-DISASSOCIATE.request primitive. The AP disassociates the corresponding non-AP STA with Reason Code "Requested service not authorized in this location."

10.24.5.3 Reporting and session control with SSPN

An AP with dot11SSPNInterfaceActivated equal to true shall create a dot11InterworkingEntry in its dot11InterworkingTable for each STA that successfully associates. Permissions received from the SSPN for each associated STA shall be populated into the table; if no permissions are received from the SSPN for a particular non-AP STA, then the default permissions or an AP's locally defined policy may be used for that STA's dot11InterworkingEntry. If the AP's local policy is more restrictive than an object's permission value received from the SSPN Interface, then the AP's local policy may be enforced instead.

In an AP when dot11SSPNInterfaceActivated is equal to true, the following procedure occurs:

— The non-AP STA's state contained within the dot11InterworkingEntry shall be transmitted to the new AP after a successful transition. The state definition and the protocol used to transfer the state are beyond the scope of this standard. The new AP shall not forward any frames for that non-AP STA until it receives the dot11InterworkingEntry from the prior AP.

— After the state is successfully transmitted to the new AP, the dot11InterworkingEntry for that non-AP STA shall be deleted from the prior AP's dot11InterworkingTable.

An AP with dot11SSPNInterfaceActivated equal to true shall delete the dot11InterworkingEntry for a non-AP STA when it disassociates from the BSS.

An AP with dot11SSPNInterfaceActivated equal to true shall enforce the dot11InterworkingEntry limits for a particular non-AP STA by comparing the values of octet counters to authorized access limits:

— dot11NonAPStationVoiceOctetCount is compared to dot11NonAPStationAuthMaxVoiceOctets. When the value of the authorized maximum octet count is exceeded, if the ACM field for AC_VO is equal to 1 then the HC shall delete all admitted TSs on this access category and deny all subsequent ADDTS request frames with TID set 6 or 7, or if the ACM field for AC_VO is equal to 0 then the non-AP STA shall be disassociated using the MLME-DISASSOCIATE.request primitive with a reason code of "Disassociated because authorized access limit reached."

— dot11NonAPStationVideoOctetCount is compared to dot11NonAPStationAuthMaxVideoOctets. When the value of the authorized maximum octet count is exceeded, if the ACM field for AC_VI is equal to 1 then the HC shall delete all admitted TSs on this access category and deny all subsequent ADDTS request frames with TID set 4 or 5, or if the ACM field for AC_VI is equal to 0 then the non-AP STA shall be disassociated using the MLME-DISASSOCIATE.request primitive with a reason code of “Disassociated because authorized access limit reached.”

— dot11NonAPStationBestEffortOctetCount is compared to dot11NonAPStationAuthMaxBestEffortOctets. When the value of the authorized maximum octet count is exceeded, if the ACM field for AC_BE is equal 1 then the HC shall delete all admitted TSs on this access category and deny all subsequent ADDTS request frames with TID set 0 or 3, or if the ACM field for AC_BE is equal 0 then the non-AP STA shall be disassociated using the MLME-DISASSOCIATE.request primitive with a reason code of “Disassociated because authorized access limit reached.”

— dot11NonAPStationBackgroundOctetCount is compared to dot11NonAPStationAuthMaxBackgroundOctets. When the value of the authorized maximum octet count is exceeded, if the ACM field for AC_BK is equal 1 then the HC shall delete all admitted TSs on this access category and deny all subsequent ADDTS request frames with TID set 1 or 2, or if the ACM field for AC_BK is equal 0 then the non-AP STA shall be disassociated using the MLME-DISASSOCIATE.request primitive with a reason code of “Disassociated because authorized access limit reached.”

— dot11NonAPStationHCCAHEMMOctetCount is compared to dot11NonAPStationAuthMaxHCCAHEMMOctets. When the value of the authorized maximum octet count is exceeded, then the HC shall delete all admitted TSs with access policy of HCCA or HEMM and deny all subsequent ADDTS request frames with access policy equal HCCA or HEMM.

— The sum of dot11NonAPStationVoiceOctetCount, dot11NonAPStationVideoOctetCount, dot11NonAPStationBestEffortOctetCount, dot11NonAPStationAuthMaxBackgroundOctets, and dot11NonAPStationHCCAHEMMOctetCount is compared to dot11NonAPStationAuthMaxTotalOctets. When the value of the authorized maximum octet count is exceeded, the non-AP STA shall be disassociated using the MLME-DISASSOCIATE.request primitive with a reason code of “Disassociated because authorized access limit reached.”

10.24.6 Interworking procedures: emergency services support

Emergency services support provides STAs with the ability to contact authorities in an emergency situation. The following procedures allow the STA to determine whether emergency services are supported by the AP or mesh STA, and whether unauthenticated emergency service access is allowed.

In an AP, when dot11ESNetwork is true, the network is dedicated and limited to accessing emergency services. When dot11ESNetwork is true, the Access Network Type field in the Interworking element shall be set to the value for “Emergency services only network” (see Table 8-174). When dot11ESNetwork is false, the network is not limited to accessing emergency services, and the access network type field in the Interworking element shall be set to a value other than “Emergency services only network.” See Table 10-11.

Table 10-11—ESR and UESA field settings

Description	ESR	UESA
It is unspecified whether emergency services are reachable.	0	0
Emergency services are only reachable for authenticated STAs.	1	0
Reserved	0	1
Emergency services are reachable for STAs.	1	1

When an AP is located in a regulatory domain that requires location capabilities, the ESR field shall only be set to 1 and the Network Type shall only be set to “Emergency services only network” (see Table 8-174), if location capability is enabled on the AP. In Beacon and Probe Response frames, location capability is advertised when the Civic Location or Geo Location field in the Extended Capabilities Element is set to 1.

In an MBSS, if location capability is supported, the mesh STA shall report its location for emergency services.

When dot11ESNetwork is true in a mesh STA, the ESR shall be set to 1. When that mesh STA receives a Mesh Peering Open frame that includes the Interworking element with the ASRA field equal to 1, it allows access to emergency services and forwards MSDUs to an emergency server.

NOTE—The ASRA bit set to 1, informs the mesh STA to prioritize resources for the emergency call, to proactively find a better path before the link conditions deteriorate below a certain threshold, and/or to change some of the mesh STA's behavior (for example, to disable any power save features).

When dot11ESNetwork is false in a mesh STA, the ESR shall be set to 0. When that mesh STA receives a Mesh Peering Open frame that includes the Interworking element with the ASRA field equal to 1, it is unable to support the mesh peering of emergency services and does not forward MDSUs to an emergency server.

10.24.7 Interworking procedures: emergency alert system (EAS) support

The EAS provides alerts, typically issued by authorities. The Interworking Procedures EAS support enables the alerts to be transmitted upon request from APs to non-AP STAs. Subsequent to advertisement in Beacon and Probe Response frames, a non-AP STA uses GAS queries to retrieve an EAS message from the network according to the following procedures.

When dot11EASActivated is true, EAS operation shall be supported. When EAS operation is not supported, dot11EASActivated shall be set to false.

When the IEEE 802.11 infrastructure is informed of the availability of an EAS message (the mechanism by which is outside the scope of this standard), an AP with dot11EASActivated equal true shall advertise the availability of the EAS message by including an Emergency Alert Identifier element (see 8.4.2.99) for that message in its Beacon and Probe Response frames. The AP shall include one instance of an Emergency Alert Identifier element in its Beacon and Probe Response frames for each active EAS Message. The Emergency Alert Identifier element provides an Alert Identifier Hash value, a unique indicator of the EAS Message of the alert to the non-AP STA. The Alert Identifier Hash value allows the non-AP STA to determine whether this is a new alert.

NOTE—The same value of hash will be computed by each AP in an ESS and by each AP in different ESSs. Thus a non-AP STA, which can download emergency alert messages when in a preassociated state, can unambiguously determine that it has already downloaded the message, avoiding unnecessary duplicates.

When an EAS Message has expired (the mechanism by which is outside the scope of this standard), an AP with dot11EASActivated equal true shall remove the corresponding instance of an Emergency Alert Identifier element from its Beacon and Probe Response frames.

The Alert Identifier Hash in the Emergency Alert Identifier element shall be computed using HMAC-SHA1-64 hash algorithm as shown in 8.4.2.99.

After receiving an Alert Identifier Hash value for an EAS Message that has not already been retrieved from the network, a non-AP STA having dot11EASActivated equal true can retrieve the EAS message from the AP using

- The procedures defined in 10.24.3.1, transmit the Alert Identifier Hash of the desired message in the Query request field of a GAS Initial Request frame. The Advertisement Protocol ID field in the GAS Initial Request frame is set to the value for EAS (see Table 8-175).
- The Query response is a message formatted in accordance with OASIS EDXL.

- A URI formed by concatenating the Emergency Alert Server URI with the hexadecimal numerals of the Alert Identifier Hash converted to UTF-8 encoded characters and the ".xml" file extension. For example, if the Emergency Alert Server URI is http://eas.server.org and the Alert Identifier Hash is "0x1234567890abcdef," then the URI would be http://eas.server.org/1234567890abcdef.xml (the mechanism by which the URI is retrieved is outside the scope of this standard). The XML file is formatted in accordance with OASIS EDXL. The non-AP STA retrieves the Emergency Alert Server URI (see 8.4.4.16) using an ANQP query according to the procedures in 10.24.3.2. This method is recommended for non-AP STAs in the associated state.

10.24.8 Interworking procedures: support for the advertisement of roaming consortiums

APs can assist non-AP STAs performing network discovery and selection through the advertisement of a Roaming Consortium element. The Roaming Consortium element contains information identifying an SSP or group of SSPs (i.e., a roaming consortium) whose security credentials might be used to authenticate with the AP transmitting this element. An SSP or group of SSPs can register for and obtain an OI using the procedures defined in [B19]. Note that a non-AP STA may also use GAS procedures defined in 10.24.3.2.3 to retrieve a Roaming Consortium ANQP-element, which might contain more OIs than the Roaming Consortium element.

APs having dot11InterworkingServiceActivated equal true and having one or more entries in the dot11RoamingConsortiumTable shall include the Roaming Consortium element in Beacon and Probe response frames. APs shall only include an OI in the dot11RoamingConsortiumTable, if in conjunction with an AS, it is capable of successfully authenticating a non-AP STA having valid security credentials for the SSPN identified by that OI. Methods used by the AP to authenticate the non-AP STA include, but are not limited to, RSNA algorithms and Open System authentication.

A non-AP STA might have a locally stored binding between an OI and a set of security credentials with which it can authenticate to the network identified by the OI, that is, the SSPN. The method by which this binding is obtained is outside the scope of this standard. A non-AP STA might select from that list of credentials when authenticating to the BSS.

10.24.9 Interworking procedures: support for QoS mapping from external networks

Maintaining proper end-to-end QoS is an important factor when providing interworking service. This is because the external networks may employ different network-layer (Layer 3) QoS practices. For example, the use of a particular differentiated services code point (DSCP) for a given service may be different between different networks. To ensure the proper QoS over-the-air in the IEEE 802.11 infrastructure, the mapping from DSCP to UP for the corresponding network needs to be identified and made known to the STAs. If an inconsistent mapping is used then:

- Admission control at the AP may incorrectly reject a service request, because the non-AP STA used the incorrect UP.
- Non-AP STAs may use the incorrect value for User Priority in TSPEC and TCLAS elements.
- The user may be given a different QoS over the IEEE 802.11 network than expected, e.g., a lower QoS may be provided than the STA expected.

Therefore, APs with dot11QosMapActivated equal true shall set the QoS Map field in the Extended Capabilities element to 1; APs with dot11QosMapActivated equal false shall set the QoS Map field in the Extended Capabilities element to 0. The AP's SME causes the QoS Map Set to be available to higher layer protocols or applications so they will be able to set the correct priority in an MA-UNITDATA.request primitive.

For frames transmitted by an AP belonging to an admitted TS, the UP obtained from the TS's TCLAS element shall be used instead of the UP derived from the QoS Map Set. For frames transmitted by an AP belonging to an admitted TS not having a TCLAS element, the UP shall be derived from the QoS Map Set.

Non-AP STAs when dot11QosMapActivated is equal true shall set the QoS Map field in the Extended Capabilities element to 1. An AP receiving an Association request frame or Reassociation Request frame when the QoS Map field in the Extended Capabilities element is equal 1 shall include the QoS Map Set element in the corresponding Association response frame or Reassociation response frame as defined in 8.3.3.6 or 8.3.3.8 respectively. Upon receiving the QoS Map Set element, the non-AP STA's SME causes the QoS Map Set to be available to higher layer protocols or applications so they will be able to set the correct priority in an MA-UNITDATA.request primitive.

When the AP's SME detects a change in the QoS mapping information, it shall update the non-AP STA with the new QoS Map Set element. It accomplishes this update by invoking the MLME-QoSMap.request primitive.

When the MAC entity at the non-AP STA receives a QoS Map Configure frame from the AP, the MLME shall issue an MLME-QoSMap.indication primitive to its SME.

When the non-AP STA's SME receives the QoS Map response, it shall make the QoS Map available to higher layers so that in turn, they can invoke the MA-UNITDATA.request with the correct priority.

11. Security

11.1 Framework

11.1.1 Classes of security algorithm

This standard defines two classes of security algorithms for IEEE 802.11 networks:

- Algorithms for creating and using an RSNA, called *RSNA algorithms*
- Pre-RSNA algorithms

NOTE—This standard does not prohibit STAs from simultaneously operating pre-RSNA and RSNA algorithms.

The use of WEP for confidentiality, authentication, or access control is deprecated. The WEP algorithm is unsuitable for the purposes of this standard.

The use of TKIP is deprecated. The TKIP algorithm is unsuitable for the purposes of this standard.

11.1.2 Security methods

Pre-RSNA security comprises the following algorithms:

- WEP, described in 11.2.2
- IEEE 802.11 entity authentication, described in 11.2.3

RSNA security comprises the following algorithms:

- TKIP, described in 11.4.2
- CCMP, described in 11.4.3
- BIP, described in 11.4.4
- RSNA establishment and termination procedures, including use of IEEE 802.1X authentication, described in 11.5 and SAE authentication described in 11.3
- Key management procedures, described in 11.6

11.1.3 RSNA equipment and RSNA capabilities

RSNA-capable equipment can create RSNAs. When dot11RSNAActivated is true, RSNA-capable equipment shall include the RSNE in Beacon, Probe Response, and (Re)Association Request frames and in Message 2 and Message 3 of the 4-Way Handshake. Pre-RSNA equipment is not capable of creating RSNAs.

All STAs implementing RSNA shall support the RSNE described in 8.4.2.27.

11.1.4 RSNA establishment

An SME establishes an RSNA in one of four ways:

a) If an RSNA is based IEEE 802.1X AKM in an ESS an RSNA-capable STA’s SME establishes an RSNA as follows:
   1) It identifies the AP as RSNA-capable from the AP’s Beacon or Probe Response frames.
   2) It shall invoke Open System authentication.
   3) It negotiates cipher suites during the association process, as described in 11.5.2 and 11.5.3.
   4) It uses IEEE Std 802.1X-2004 to authenticate, as described in 11.5.9 and 11.5.10.

   5) It establishes temporal keys by executing a key management algorithm, using the protocol defined by 11.6 or 12.
   6) It protects the data link by programming the negotiated cipher suites and the established temporal key into the MAC and then invoking protection. See, for example, 11.4.3 for a description of the RSNA data protection mechanisms.
   7) If the STAs negotiate management frame protection, the SME programs the TK and pairwise cipher suite into the MAC for protection of individually addressed robust management frames. It also installs the IGTK and IPN for protection of group addressed robust management frames.

b) If an RSNA is based on a PSK or password in an ESS, the SME establishes an RSNA as follows:
   1) It identifies the AP as RSNA-capable from the AP's Beacon or Probe Response frames.
   2) If the RSNA-capable AP advertises support for SAE authentication in its Beacon or Probe Response frames, and the STA has a group defined in the dot11RSNAConfigDLCGroupTable and a password for the AP in the dot11RSNAConfigPasswordValueTable, the STA shall invoke SAE authentication to establish a PMK. If the RSNA-capable AP does not advertise support for SAE authentication in its Beacon and Probe Response frames but advertises support for the alternate form of PSK authentication (see 4.10.3.4), and the STA also supports the alternate form of PSK authentication, the STA may invoke Open System authentication and use the PSK as the PMK with the key management algorithm in step 4) below.
   3) It negotiates cipher suites during the association process, as described in 11.5.2 and 11.5.3.
   4) It establishes temporal keys by executing a key management algorithm, using the protocol defined by 11.6.
   5) It protects the data link by programming the negotiated cipher suites and the established temporal key into the MAC and then invoking protection.
   6) If the STAs negotiate management frame protection, the STA programs the TK and pairwise cipher suite into the MAC for protection of individually addressed robust management frames. It also installs the IGTK and IPN for protection of group addressed robust management frames.

c) If an RSNA is based on a PSK or password in an IBSS the SME executes the following sequence of procedures:
   1) It identifies the peer as RSNA-capable from the peer's Beacon or Probe Response frames.

      NOTE—STAs might respond to a data MPDU from an unrecognized STA by sending a Probe Request frame to find out whether the unrecognized STA is RSNA-capable.

   2) If the RSNA-capable peer advertises support for SAE authentication in its Beacon and Probe Response frames and the STA has a group defined in the dot11RSNAConfigDLCGroupTable and a password for the peer in the dot11RSNAConfigPasswordValueTable, the STA shall invoke SAE authentication and establish a PMK. If the RSNA-capable peer does not advertise support for SAE authentication but advertises support for the alternate form of PSK authentication (see 4.10.3.4), and the STA also supports the alternate form of PSK authentication the STA may optionally invoke Open System authentication and use a PSK as the PMK with the alternate form of PSK authentication.
   3) Each STA uses the procedures in 11.6, to establish temporal keys and to negotiate cipher suites. Note that two peer STAs may follow this procedure simultaneously. See 11.5.13.
   4) It protects the data link by programming the negotiated cipher suites and the established temporal key and then invoking protection.

d) If an RSNA is based IEEE 802.1X AKM in an IBSS an RSNA-capable STA's SME establishes an RSNA as follows:
   1) It identifies the peer as RSNA-capable from the peer's Beacon or Probe Response frames.

      NOTE—STAs might respond to a data MPDU from an unrecognized STA by sending a Probe Request frame to find out whether the unrecognized STA is RSNA-capable.

2) It may optionally invoke Open System authentication.

3) Each STA uses IEEE Std 802.1X-2004 to authenticate with the AS associated with the other STA's Authenticator, as described in 11.5.9 and 11.5.10.

   NOTE—Two peer STAs may follow this procedure simultaneously.

4) Each SME establishes temporal keys by executing a key management algorithm, using the protocol defined in 11.6. Hence two such key management algorithms are happening in parallel between any two STA's Supplicants and Authenticators.

5) Both STAs use the agreed-upon temporal key portion of the PTK and pairwise cipher suite from one of the exchanges to protect the link. Each STA uses the GTK established by the exchange it initiated to protect the group addressed frames it transmits.

The time a security association takes to set up shall be less than dot11RSNAConfigSATimeout. The security association setup starts when initiated by the SME and completes when the MLME-SETPROTECTION.request primitive is invoked. The action the STA takes on the timeout is a policy decision. Some options include retrying the security association setup or trying another STA. This timeout allows recovery when one of the STAs setting up a security association fails to respond correctly to setting up the security association. It also allows recovery in IBSS when one of the two security associations fails because of a security association timeout.

Only Authentication frames with the authentication algorithm equal to Open System authentication or FT authentication may be used within an RSNA. RSNA STAs shall not associate if Shared Key authentication was invoked prior to RSN association.

11.1.5 RSNA PeerKey Support

The PeerKey protocol provides mutual authentication, session identification, and data confidentiality for a station-to-station connection. A PeerKey association, composed of an SMKSA and an STKSA, shall be allowed only within the context of an existing RSNA by both peers with a common AP. Both the initiator STA and the peer STA shall ensure that dot11RSNAActivated is true before initiating the STSL master key (SMK) Handshake and STSL transient key (STK) Handshake and establishing their respective security associations.

A STSL session may chose to allow unprotected communication between STAs. In this case, the PeerKey protocol is not used.

11.1.6 RSNA assumptions and constraints

An RSNA assumes the following:

a) Each STA can generate cryptographic-quality random numbers. This assumption is fundamental, as cryptographic methods require a source of randomness. See M.6 for suggested hardware and software methods to achieve randomness suitable for this purpose.

b) When IEEE 802.1X authentication is used, the specific EAP method used performs mutual authentication. This assumption is intrinsic to the design of RSN in IEEE 802.11 LANs and cannot be removed without exposing both the STAs to man-in-the-middle attacks. EAP-MD5 is an example of an EAP method that does not meet this constraint (see IETF RFC 3748 [B38]). Furthermore, the use of EAP authentication methods where server and client credentials are not differentiated reduces the security of the method to that of a PSK due to the fact that malicious insiders might masquerade as servers and establish a man-in-the-middle attack.

   In particular, the mutual authentication requirement implies an unspecified prior enrollment process (e.g., a long-lived authentication key or establishment of trust through a third party such as a certification authority), as the STA needs to be able to identify the ESS or IBSS as a trustworthy entity and vice versa. The STA shares authentication credentials with the AS utilized by the selected AP or, in the case of PSK, the selected AP. The SSID provides an unprotected indication that the selected AP's authentication entity shares credentials with the STA. Only the successful completion of the IEEE 802.1X EAP or PSK authentication, after association, validates any such indication that the AP is connected to an authorized network or service provider.

c) The mutual authentication method needs to be strong, meaning impersonation attacks are computationally infeasible when based on the information exposed by the authentication. This assumption is intrinsic to the design of RSN.

d) The AP and AS have a trustworthy channel between them that can be used to exchange cryptographic keys without exposure to any intermediate parties.

e) An IEEE 802.1X AS never exposes the common symmetric key to any party except the AP with which the STA is currently communicating. This is a very strong constraint. It implies that the AS itself is never compromised. It also implies that the IEEE 802.1X AS is embedded in the AP or that the AP is physically secure and the AS and the AP lie entirely within the same administrative domain. This assumption follows from the fact that if the AP and the AS are not collocated or do not share pairwise key encryption keys directly, then it is impossible to assure the mobile STA that its key, which is distributed by the AS to the AP, has not been compromised prior to use.

f) Similarly, a STA never shares with a third party a common symmetric key that it shares with a peer. Doing so destroys the utility of the key for detecting MPDU replay and forgery.

g) The Supplicant and the Authenticator generate a different, fresh PTK for each session between the pair. This assumption is fundamental, as reuse of any PTK would enable compromise of all the data protected by that key.

h) The destination STA chosen by the transmitter is the correct destination. For example, Address Resolution Protocol (ARP) and Internet Control Message Protocol (ICMP) are methods of determining the destination STA's MAC address that are not secure from attacks by other members of the ESS. One of the possible solutions to this problem might be for the STA to send or receive only frames whose final DA or SA are the AP and for the AP to provide a network layer routing function. However, such solutions are outside the scope of this standard.

An HT STA shall not use either of the pairwise cipher suite selectors: "Use group cipher suite" or TKIP to communicate with another HT STA.

11.1.7 Requirements for robust management frame protection

Action frames specified with "No" in the "Robust" column of Table 8-38 are not robust management frames and shall not be protected.

When management frame protection is negotiated, all group addressed robust management frames shall be encapsulated using the procedures defined in 10.13.

11.1.8 Emergency service establishment in an RSN

An AP or mesh STA that supports RSNAs and has the ESR bit set to 1 and the UESA bit set to 1 in the Interworking element in Beacon and Probe Response frames supports both RSNAs and emergency services associations (see 10.3.5.2) simultaneously.

In an infrastructure BSS, the STAs with emergency services association should discard all group addressed frames they receive, as they do not possess the Group Key and will not be able to decrypt group addressed frames. In an RSNA enabled BSS that has one or more STAs associated with an emergency services association, an AP should avoid transmitting unprotected group addressed frames in order not to disturb the operation of STAs that are in possession of Group Key. One possible way of achieving this is to support Proxy-ARP in the AP, as defined in 10.23.13. In addition, it is recommended that an AP supporting emergency services association should also support DMS to convert group addressed frames to individually addressed frames and transmit them to STAs associated using the emergency services association. STAs using emergency services association might request DMS if needed.

11.2 Pre-RSNA security methods

11.2.1 Status of Pre-RSNA security methods

Except for Open System authentication, all pre-RSNA security mechanisms have been deprecated, as they fail to meet their security goals. New implementations should support pre-RSNA methods only to aid migration to RSNA methods.

Open System Authentication and Open System Deauthentication shall not be used between mesh STAs.

11.2.2 Wired equivalent privacy (WEP)

11.2.2.1 WEP overview

WEP-40 was defined as a means of protecting (using a 40-bit key) the confidentiality of data exchanged among authorized users of a WLAN from casual eavesdropping. Implementation of WEP is optional. The same algorithms have been widely used with a 104-bit key instead of a 40-bit key in fielded implementations; this is called WEP-104. The WEP cryptographic encapsulation and decapsulation mechanics are the same whether a 40-bit or a 104-bit key is used. The term WEP by itself refers to either WEP-40 or WEP-104.

11.2.2.2 WEP MPDU format

Figure 11-1 depicts the encrypted frame body as constructed by the WEP algorithm.

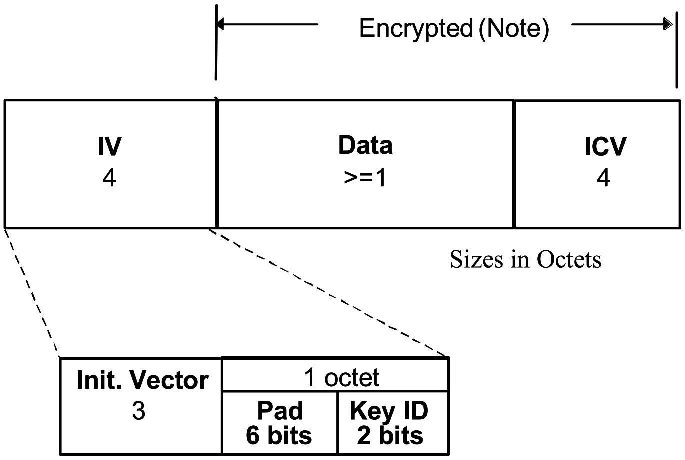


Figure 11-1—Construction of expanded WEP MPDU

The WEP ICV field shall be 32 bits in length. The expanded frame body shall start with a 32-bit IV field. This field shall contain three subfields: a 3-octet subfield that contains the IV, a 2-bit Key ID subfield, and a 6-bit Pad subfield. The ordering conventions defined in 8.2.2 apply to the IV field and its subfields and to the ICV field. The Key ID subfield contents select one of four possible secret key values for use in decrypting this frame body. When key-mapping keys are used, the Key ID field value is ignored.

Interpretation of these bits is discussed further in 11.2.2.3. The contents of the Pad subfield shall be 0. The Key ID subfield occupies the 2 MSBs of the last octet of the IV field, while the Pad subfield occupies the 6 LSBs of this octet.

11.2.2.3 WEP state

WEP uses encryption keys only; it performs no data authentication. Therefore, it does not have data integrity keys. WEP uses two types of encryption keys: key-mapping keys and default keys.

A key-mapping key is an unnamed key corresponding to a distinct transmitter address-receiver address <TA,RA> pair. Implementations shall use the key-mapping key if it is configured for a <TA,RA> pair. In other words, the key-mapping key shall be used to WEP-encapsulate or -decapsulate MPDUs transmitted by TA to RA, regardless of the presence of other key types. When a key-mapping key for an address pair is present, the WEP Key ID subfield in the MPDU shall be set to 0 on transmit and ignored on receive.

A default key is an item in a four-element MIB array called dot11WEPDefaultKeys, named by the value of a related array index called dot11WEPDefaultKeyID. If a key-mapping key is not configured for a WEP MPDU's <TA,RA> pair, WEP shall use a default key to encapsulate or decapsulate the MPDU. On transmit, the key selected is the element of the dot11DefaultKeys array given by the index dot11WEPDefault-KeyID—a value of 0, 1, 2, or 3—corresponding to the first, second, third, or fourth element, respectively, of dot11WEPDefaultKeys. The value the transmitter encodes in the WEP Key ID subfield of the transmitted MPDU shall be the dot11WEPDefaultKeyID value. The receiver shall use the Key ID subfield of the MPDU to index into dot11WEPDefaultKeys to obtain the correct default key. All WEP implementations shall support default keys.

NOTE—Many implementations also support 104-bit WEP keys. These are used exactly like 40-bit WEP keys: a 24-bit WEP IV is prepended to the 104-bit key to construct a 128-bit WEP seed, as explained in 11.2.2.4.3. The resulting 128-bit WEP seed is then consumed by the ARC4 stream cipher. This construction based on 104-bit keys affords no more assurance than the 40-bit construction, and its implementation and use are in no way condoned by this standard. Rather, the 104-bit construction is noted only to document de facto practice.

The default value for all WEP keys shall be null. WEP implementations shall discard the MSDU and generate an MA-UNITDATA-STATUS.indication primitive with transmission status indicating that a frame may not be encapsulated with a null key in response to any request to encapsulate an MPDU with a null key.

11.2.2.4 WEP procedures

11.2.2.4.1 WEP ICV algorithm

The WEP ICV shall be computed using the CRC-32, as defined in 8.2.4.7, calculated over the plaintext MPDU Data (PDU) field.

11.2.2.4.2 WEP encryption algorithm

A WEP implementation shall use the ARC4 stream cipher from RSA Security, Inc., as its encryption and decryption algorithm. ARC4 uses a pseudorandom number generator (PRNG) to generate a key stream that it exclusive-ORs (XORs) with a plaintext data stream to produce cipher text or to recover plaintext from a cipher text.

11.2.2.4.3 WEP seed construction

A WEP implementation shall construct a per-MPDU key, called a *seed*, by concatenating an encryption key to an IV.

For WEP-40, bits 0–39 of the WEP key correspond to bits 24–63 of the seed, and bits 0–23 of the IV correspond to bits 0–23 of the seed, respectively. The bit numbering conventions in 8.2.2 apply to the seed. The seed shall be the input to ARC4, in order to encrypt or decrypt the WEP Data and ICV fields.

NOTE—For WEP-104, bits 0–103 of the WEP key correspond to bits 24–127 of the seed, and bit 0–23 of the IV correspond to bits 0–23 of the seed, respectively.

The WEP implementation encapsulating an MPDU's plaintext data should select a new IV for every MPDU it WEP-protects. The IV selection algorithm is unspecified. The algorithm used to select the encryption key used to construct the seed is also unspecified.

The WEP implementation decapsulating an MPDU shall use the IV from the received MPDU's Init Vector subfield. See 11.2.2.4.5 for the specification of how the decapsulator selects the key to use to construct the per-MPDU key.

11.2.2.4.4 WEP MPDU cryptographic encapsulation

WEP shall apply three transformations to the plaintext MPDU to effect the WEP cryptographic encapsulation. WEP computes the ICV over the plaintext data and appends this after the MPDU data. WEP encrypts the MPDU plaintext data and ICV using ARC4 with a seed constructed as specified in 11.2.2.4.3. WEP encodes the IV and key identifier into the IV field, prepended to the encrypted Data field.

Figure 11-2 depicts the WEP cryptographic encapsulation process. The ICV shall be computed and appended to the plaintext data prior to encryption, but the IV encoding step may occur in any order convenient for the implementation.

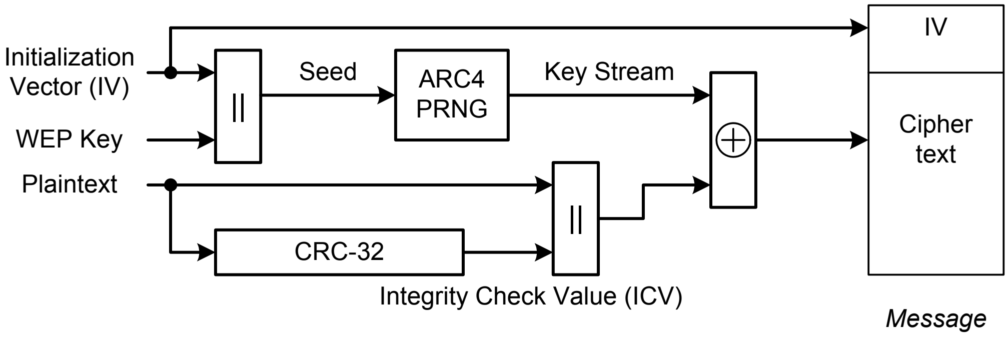


Figure 11-2—WEP encapsulation block diagram

11.2.2.4.5 WEP MPDU decapsulation

WEP shall apply three transformations to the WEP MPDU to decapsulate its payload. WEP extracts the IV and key identifier from the received MPDU. If a key-mapping key is present for the <TA,RA> pair, then this shall be used as the WEP key. Otherwise, the key identifier is extracted from the Key ID subfield of the WEP IV field in the received MPDU, identifying the default key to use.

WEP uses the constructed seed to decrypt the Data field of the WEP MPDU; this produces plaintext data and an ICV. Finally WEP recomputes the ICV and bit-wise compares it with the decrypted ICV from the MPDU. If the two are bit-wise identical, then WEP removes the IV and ICV from the MPDU, which is accepted as valid. If they differ in any bit position, WEP generates an error indication to MAC management. MSDUs with erroneous MPDUs (due to inability to decrypt) shall not be passed to LLC.

Figure 11-3 depicts a block diagram for WEP decapsulation. Unlike cryptographic encapsulation, the decapsulation steps shall be in the indicated order.

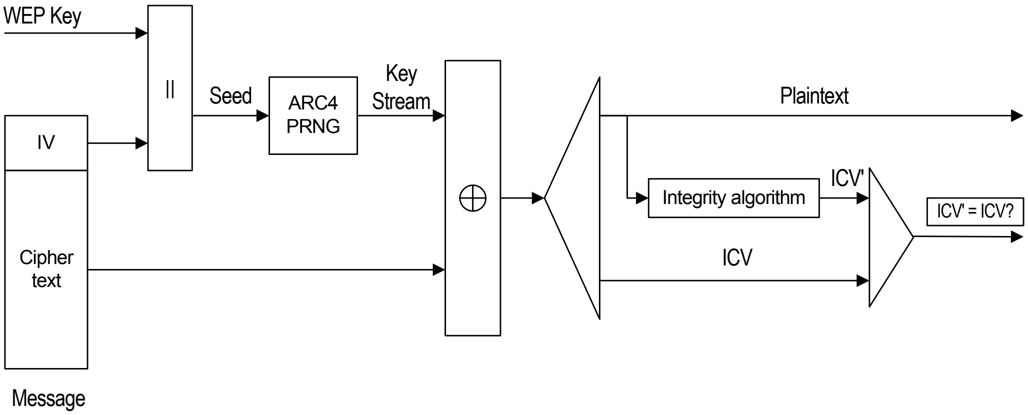


Figure 11-3—WEP decapsulation block diagram

11.2.3 Pre-RSNA authentication

11.2.3.1 Overview

In an ESS, a STA and an AP both complete an IEEE 802.11 authentication exchange prior to association. Such an exchange is optional in an independent BSS network.

All management frames of subtype Authentication shall be individually addressed, as IEEE 802.11 authentication is performed between pairs of STAs, i.e., group addressed authentication is not allowed. Management frames of subtype Deauthentication are advisory and may be sent as group addressed frames.

Shared Key authentication is deprecated and should not be implemented except for backward compatibility with pre-RSNA devices.

11.2.3.2 Open System authentication

11.2.3.2.1 General

Open System authentication is a null authentication algorithm. Any STA requesting Open System authentication may be authenticated if dot11AuthenticationAlgorithm at the recipient STA is Open System authentication. A STA may decline to authenticate with another requesting STA. Open System authentication is the default authentication algorithm for pre-RSNA equipment.

Open System authentication utilizes a two-message authentication transaction sequence. The first message asserts identity and requests authentication. The second message returns the authentication result. If the result is “successful,” the STAs shall be declared mutually authenticated.

In the description in 11.2.3.2.2 and 11.2.3.2.3, the STA initiating the authentication exchange is referred to as the *requester*, and the STA to which the initial frame in the exchange is addressed is referred to as the *responder*. The specific items in each of the messages described in the following subclauses are defined in 8.3.3.11, Table 8-28, and Table 8-29.

11.2.3.2.2 Open System authentication (first frame)

Upon receipt of an Open System MLME-AUTHENTICATE.request primitive, the requester shall perform the following procedure:

a) If one or more request parameters are invalid, issue an MLME-AUTHENTICATE.confirm primitive with ResultCode set to INVALID_PARAMETERS; else

b) Construct an Open System authentication request frame and transmit it to the responder.

11.2.3.2.3 Open System authentication (final frame)

Upon receipt of an authentication frame requesting Open System authentication, the responder may authenticate the requester using the following procedure:

a) Issue an MLME-AUTHENTICATE.indication primitive to inform the SME of the authentication request.

b) Construct and transmit an authentication response frame with the fields as defined in 8.3.3.6 and the status field as defined in 8.4.1.9.

If dot11AuthenticationAlgorithm does not include the value "Open System," the result code shall not take the value "successful."

11.2.3.3 Shared Key authentication

11.2.3.3.1 General

Shared Key authentication seeks to authenticate STAs as either a member of those who know a shared secret key or a member of those who do not.

Shared Key authentication may be used if WEP has been selected and shall not be used otherwise.

This mechanism uses a shared key delivered to participating STAs via a secure channel that is independent of IEEE Std 802.11. This shared key is set in a write-only MIB attribute with the intent to keep the key value internal to the STA.

A STA shall not initiate a Shared Key authentication exchange unless its dot11PrivacyOptionImplemented attribute is true.

In the description in 11.2.3.3.2 to 11.2.3.3.6, the STA initiating the authentication exchange is referred to as the *requester*, and the STA to which the initial frame in the exchange is addressed is referred to as the *responder.* The specific items in each of the messages described in the following subclauses are defined in 8.3.3.11, Table 8-28, and Table 8-29.

11.2.3.3.2 Shared Key authentication (first frame)

Upon receipt of a Shared Key MLME-AUTHENTICATE.request primitive, the requester shall perform the following procedure:

a) If one or more request parameters are invalid, issue an MLME-AUTHENTICATE.confirm primitive with ResultCode set to INVALID_PARAMETERS; else

b) Construct a Shared Key authentication request frame and transmit it to the responder.

11.2.3.3.3 Shared Key authentication (second frame)

Upon receipt of an authentication frame requesting Shared Key authentication, the responder may authenticate the requester using the procedure here and in the following two frames:

a) Issue an MLME-AUTHENTICATE.indication primitive to inform the SME of the authentication request.

b) Before sending the second frame in the Shared Key authentication sequence, the responder shall use WEP to generate a string of octets to be used as the authentication challenge text.

c) Construct and transmit to the requester an authentication response frame with the fields as defined in 8.3.3.6 and the status field as defined in 8.4.1.9.

If the status code is not "successful," this shall be the last frame of the transaction sequence; and the content of the challenge text field is unspecified.

If the status code is "successful," the following additional information items shall have valid contents:

- Authentication algorithm dependent information = The challenge text
- This authentication result shall be of fixed length of 128 octets. The field shall be filled with octets generated by the WEP PRNG. The actual value of the challenge field is unimportant, but the value shall not be a static value.

11.2.3.3.4 Shared Key authentication (third frame)

The requester shall copy the challenge text from the second frame into a third authentication frame. The third frame shall be transmitted to the responder after cryptographic encapsulation by WEP, as defined in 11.2.2, using the shared key.

11.2.3.3.5 Shared Key authentication (final frame)

The responder shall WEP-decapsulate the third frame as described in 11.2.2. If the WEP ICV check is successful, the responder shall compare the decrypted contents of the Challenge Text field with the challenge text sent in second frame. If they are the same, then the responder shall transmit an authentication frame to the requester with a successful status code in the final frame of the sequence. If the WEP ICV check fails or challenge text comparison fails, the responder shall respond with an unsuccessful status code in final frame.

11.2.3.3.6 Shared key MIB attributes

To transmit a management frame of subtype Authentication, with an Authentication Transaction Sequence Number field value of 2, the MAC shall operate according to the following decision tree:

```
if dot11PrivacyOptionImplemented is "false" then
    the MMPDU is transmitted with a sequence of 0 octets in the Challenge Text field and a status
    code value of 13
else
    the MMPDU is transmitted with a sequence of 128 octets generated using the WEP PRNG and
    a key whose value is unspecified and beyond the scope of this standard and a randomly chosen
    IV value (note that this is typically selected by the same mechanism for choosing IV values for
    transmitted data MPDUs) in the Challenge Text field and a status code value of 0 (the IV used
    is immaterial and is not transmitted). Note that there are cryptographic issues involved in the
    choice of key/IV for this process as the challenge text is sent unencrypted and, therefore,
    provides a known output sequence from the PRNG.
endif
```

To receive a management frame of subtype Authentication, with an Authentication Transaction Sequence Number field value of 2, the MAC shall operate according to the following decision tree:

```
if the Protected Frame subfield of the Frame Control field is 1 then
    respond with a status code value of 15
else
    if dot11PrivacyOptionImplemented is "true" then
```

```
        if there is a mapping in dot11WEPKeyMappings matching the MSDU’s TA then
            if that key is null then
                respond with a frame whose Authentication Transaction Sequence Number field is 3 that contains the appropriate authentication algorithm number, a status code value of 15, and no Challenge Text field, without encrypting the contents of the frame
            else
                respond with a frame whose Authentication Transaction Sequence Number field is 3 that contains the appropriate authentication algorithm number, a status code value of 0, and the identical Challenge Text field, encrypted using that key, and setting the Key ID subfield in the IV field to 0
            endif
        else
            if dot11WEPDefaultKeys[dot11WEPDefaultKeyID] is null then
                respond with a frame whose Authentication Transaction Sequence Number field is 3 that contains the appropriate authentication algorithm number, a status code value of 15, and no Challenge Text field, without encrypting the contents of the frame
            else
                respond with a frame whose Authentication Transaction Sequence Number field is 3 that contains the appropriate authentication algorithm number, a status code value of 0, and the identical Challenge Text field, WEP-encapsulating the frame under the key dot11WEPDefaultKeys[dot11WEPDefaultKeyID], and setting the Key ID subfield in the IV field to dot11WEPDefaultKeyID
            endif
        endif
    else
        respond with a frame whose Authentication Transaction Sequence Number field is 3 that contains the appropriate authentication algorithm number, a status code value of 13, and no Challenge Text field, without encrypting the contents of the frame
    endif
endif
```

When receiving a management frame of subtype Authentication, with an Authentication Transaction Sequence Number field value of 3, the MAC shall operate according to the following decision tree:

```
if the Protected Frame subfield of the Frame Control field is 0 then
    respond with a status code value of 15
else
    if dot11PrivacyOptionImplemented is “true” then
        if there is a mapping in dot11WEPKeyMappings matching the MSDU’s TA then
            if that key is null then
                respond with a frame whose Authentication Transaction Sequence Number field is 4 that contains the appropriate authentication algorithm number and a status code value of 15 without encrypting the contents of the frame
            else
                WEP-decapsulate with that key, incrementing dot11WEPICVErrorCount and responding with a status code value of 15 if the ICV check fails
            endif
        else
            if dot11WEPDefaultKeys[dot11WEPDefaultKeyID] is null then
                respond with a frame whose Authentication Transaction Sequence Number field is 4 that contains the appropriate authentication algorithm number and a status code value of 15 without encrypting the contents of the frame
            else
                WEP-decapsulate with dot11WEPDefaultKeys[dot11WEPDefault-KeyID], incrementing dot11WEPICVErrorCount and responding with a status code value of 15 if the ICV check fails
            endif
        endif
    else
        respond with a frame whose Authentication Transaction Sequence Number field is 4 that contains the appropriate authentication algorithm number and a status code value of 15
    endif
endif
```

The attribute dot11PrivacyInvoked shall not take the value of true if the attribute dot11PrivacyOption-Implemented is false. Setting dot11WEPKeyMappings to a value that includes more than dot11WEPKeyMappingLengthImplemented entries is illegal and shall have an implementation-specific effect on the operation of the data confidentiality service. Note that dot11WEPKeyMappings may contain from zero to dot11WEPKeyMappingLengthImplemented entries, inclusive.

The values of the attributes in the aPrivacygrp should not be changed during the authentication sequence, as unintended operation may result.

11.3 Authentication using a password

11.3.1 SAE overview

STAs, both AP STAs and non-AP STAs, may authenticate each other by proving possession of a password. Authentication protocols that employ passwords need to be resistant to off-line dictionary attacks.

Simultaneous authentication of equals (SAE) is a variant of *Dragonfly*, a password-authenticated key exchange based on a zero-knowledge proof. SAE is used by STAs to authenticate with a password; it has the following security properties:

- The successful termination of the protocol results in a PMK shared between the two STAs.
- An attacker is unable to determine either the password or the resulting PMK by passively observing an exchange or by interposing itself into the exchange by faithfully relaying messages between the two STAs.
- An attacker is unable to determine either the password or the resulting shared key by modifying, forging, or replaying frames to an honest, uncorrupted STA.
- An attacker is unable to make more than one guess at the password per attack. This implies that the attacker cannot make one attack and then go offline and make repeated guesses at the password until successful. In other words, SAE is resistant to dictionary attack.
- Compromise of a PMK from a previous run of the protocol does not provide any advantage to an adversary attempting to determine the password or the shared key from any other instance.
- Compromise of the password does not provide any advantage to an adversary in attempting to determine the PMK from the previous instance.

Unlike other authentication protocols SAE does not have a notion of an “initiator” and “responder” or of a “supplicant” and “authenticator.” The parties to the exchange are equals, with each side being able to initiate the protocol. Each side may initiate the protocol simultaneously such that each side views itself as the “initiator” for a particular run of the protocol. Such a peer-to-peer protocol may be used in a traditional client-server (or supplicant/authenticator) fashion but the converse does not hold. This requirement is necessary to address the unique nature of MBSSs.

The parties involved will be called STA-A and STA-B. They are identified by their MAC addresses, STA-A-MAC and STA-B-MAC, respectively. STAs begin the protocol when they discover a peer through Beacons and Probe Responses, or when they receive an IEEE 802.11 authentication frame indicating SAE authentication from a peer.

SAE is an RSNA authentication protocol and is selected according to 11.5.2.

SAE shall be implemented on all mesh STAs to facilitate and promote interoperability.

11.3.2 Assumptions on SAE

SAE uses various functions and data to accomplish its task and assumes certain properties about each function. These are as follows:

- H is an "extractor" function (see IETF RFC 5869) that concentrates potentially dispersed entropy from an input to create an output that is a cryptographically strong, pseudorandom key. This function takes as input a non-secret "salt" and a secret input and produces a fixed-length output.
- CN is a confirmation function that takes a secret key and data to confirm and bind to the exchange.
- A finite cyclic group is negotiated for which solving the discrete logarithm problem is computationally infeasible.

When used with AKMs 00-0F-AC:8 or 00-0F-AC:9 from Table 8-101, H is instantiated as HMAC-SHA256:

$$H(salt, ikm) = HMAC\text{-}SHA256(salt, ikm)$$

When used with AKMs 00-0F-AC:8 or 00-0F-AC:9 from Table 8-101, CN is instantiated as a function that takes a key and a sequence of data. Each piece of data is converted to an octet string and concatenated together before being passed, along with the key, to HMAC-SHA256:

$$CN(key, X, Y, Z, \ldots) = HMAC\text{-}SHA256(key, D2OS(X) \| D2OS(Y) \| D2OS(Z) \| \ldots)$$

where D2OS() represents the data to octet string conversion functions in 11.3.7.2.

Other instantiations of functions H and CN require creation of a new AKM identifier.

11.3.3 Representation of a password

Passwords are used in SAE to deterministically compute a secret element in the negotiated group, called a "password element." The input to this process needs to be in the form of a binary string. For the protocol to successfully terminate, it is necessary for each side to produce identical binary strings for a given password, even if that password is in character format. There is no canonical binary representation of a character and ambiguity exists when the password is a character string. To eliminate this ambiguity, a compliant STA shall represent a character-based password as an ASCII string. Representation of a character-based password in another character set or use of a password preprocessing technique (to map a character string to a binary string) may be agreed upon, in an out-of-band fashion, prior to beginning SAE. If the password is already in binary form (e.g., it is a binary preshared key) no character set representation is assumed. The binary representation of the password, after being transformed from a character representation or directly if it is already in binary form, is stored in the dot11RSNASAEPasswordValueTable. When a "password" is called for in the description of SAE that follows the credential from the dot11RSNASAEPasswordValueTable is used.

11.3.4 Finite cyclic groups

11.3.4.1 General

SAE uses discrete logarithm cryptography to achieve authentication and key agreement. Each party to the exchange derives ephemeral public and private keys with respect to a particular set of domain parameters that define a finite cyclic group. Groups may be based on either Finite Field Cryptography (FFC) or on Elliptic Curve Cryptography (ECC). Each component of a group is referred to as an "element." Groups are negotiated using an identifying number from a repository maintained by IANA as "Group Description" attributes for IETF RFC 2409 (IKE)[B17]. The repository maps an identifying number to a complete set of domain parameters for the particular group. For the purpose of interoperability, conformant STAs shall support group nineteen (19), an ECC group defined over a 256-bit prime order field.

More than one group may be configured on a STA for use with SAE by using the dot11RSNAConfigDLCGroup table. Configured groups are prioritized in ascending order of preference. If only one group is configured, it is, by definition, the most preferred group.

NOTE—The preference of one group over another is a local policy issue.

SAE uses three arithmetic operators defined for both FFC and ECC groups, an operation that takes two elements to produce a third element (called the "element operation"), an operation that takes an integer (called "scalar") and an element to produce a second element (called the "scalar operation"), and an operation that takes an element to produce a second element (called the "inverse operation"). The convention used here is to represent group elements in uppercase bold italic and scalar values in lowercase italic. The element operation takes two elements, $\boldsymbol{X}$ and $\boldsymbol{Y}$, to produce a third element, $\boldsymbol{Z}$, and is denoted $\boldsymbol{Z}$ = elem-op($\boldsymbol{X}$,$\boldsymbol{Y}$); the scalar operation takes a scalar, x, and an element, $\boldsymbol{Y}$, to produce a second element **Z** and is denoted $\boldsymbol{Z}$ = scalar-op(x,$\boldsymbol{Y}$); the inverse operation takes an element, $\boldsymbol{X}$, to produce a second element, $\boldsymbol{Z}$, and is denoted $\boldsymbol{Z}$ = inverse-op($\boldsymbol{X}$).

scalar-op(x,$\boldsymbol{Y}$) is defined as successive iterations of elem-op($\boldsymbol{Y}$, $\boldsymbol{Y}$). That is, it is possible to define scalar-op(1, $\boldsymbol{Y}$) = $\boldsymbol{Y}$ and for $x > 1$, scalar-op(x, $\boldsymbol{Y}$) = elem-op(scalar-op(x-1, $\boldsymbol{Y}$), $\boldsymbol{Y}$). The specific definition of elem-op(**X**, $\boldsymbol{Y}$) depends on the type of group, either ECC or FFC.

11.3.4.2 Elliptic curve cryptography (ECC) groups

11.3.4.2.1 ECC group definition

ECC groups used by SAE are defined by the sextuple (p, a, b, $\boldsymbol{G}$, r, h) where p is a prime number, a and b specify the elliptic curve defined by the equation, $y^2 = x^3 + ax + b$ modulo p, $\boldsymbol{G}$ is a generator (a base point on the elliptic curve), r is the prime order of $\boldsymbol{G}$, and h is the co-factor. Elements in ECC groups are the points on the elliptic curve defined by their coordinates—(x, y)—that satisfy the equation for the curve and the identity element, the so-called "point at infinity."

The IANA registry used to map negotiated numbers to group domain parameters includes some ECC groups defined over a characteristic 2 finite field and may include some ECC groups with a co-factor greater than one (1). These groups shall not be used with SAE. Only ECC groups defined over an odd prime finite field with a co-factor equal to one (1) shall be used with SAE.

The element operation in an ECC group is addition of two points on the curve resulting in a third point on the curve. For example, the point $\boldsymbol{X}$ is added to the point $\boldsymbol{Y}$ to produce the point $\boldsymbol{Z}$:

$$\boldsymbol{Z} = \boldsymbol{X} + \boldsymbol{Y} = \text{elem-op}(\boldsymbol{X},\boldsymbol{Y})$$

The scalar operation in an ECC group is multiplication of a point on the curve by a scalar resulting in a second point on the curve. For example, the point $\boldsymbol{Y}$ is multiplied by the scalar x to produce the point $\boldsymbol{Z}$:

$$\boldsymbol{Z} = x\boldsymbol{Y} = \text{scalar-op}(x, \boldsymbol{Y})$$

The inverse operation in an ECC group is inversion of a point on a curve resulting in a second point on the curve. A point on an elliptic curve is the inverse of a different point if their sum is the "point at infinity." In other words:

$$\text{elem-op}(\boldsymbol{X}, \text{inverse}(\boldsymbol{X})) = \text{“point at infinity”}$$

ECC groups make use of a mapping function, F, that maps a point (x, y) that satisfies the curve equation to its x-coordinate—i.e., if $\boldsymbol{P} = (x, y)$ then $\mathrm{F}(\boldsymbol{P}) = x$. Function F is not defined with the identity element as input.

NOTE—SAE protocol operations preclude function F from ever being called with the identity element, i.e., the "point at infinity."

11.3.4.2.2 Generation of the Password Element with ECC groups

The Password Element of an ECC group (***PWE***) shall be generated in a random hunt-and-peck fashion. The password and a counter, represented as a single octet and initially set to one (1), are used with the peer identities to generate a password seed. The password seed shall then be stretched using the key derivation function (KDF) from 11.6.1.7.2 to a length equal to the bit length of the prime number, p, from the elliptic curve domain parameters with the Label being the string "SAE Hunting and Pecking" and with the Context being the prime number. If the resulting password value is greater than or equal to the prime number, the counter shall be incremented, a new password seed shall be derived and the hunting-and-pecking shall continue. Otherwise, it shall be used as the x-coordinate of a candidate point (x, y) on the curve satisfying the curve equation, if such a point exists. If no solution exists, the counter shall be incremented, a new password-seed shall be derived and the hunting-and-pecking shall continue. Otherwise, there will be two possible solutions: (x, y) and $(x, p – y)$. The password seed shall be used to determine which one to use: if the least-significant bit (LSB) of the password seed is equal to that of y, the ***PWE*** shall be set to (x, y); otherwise, it shall be set to $(x, p – y)$.

In order to address the possibility of side-channel attacks that attempt to determine the number of interations of the "hunting-and-pecking" loop required for a given <password, STA-A-MAC, STA-B-MAC> tuple, implementations should perform at least k iterations regardless of whether ***PWE*** is discovered or not. The value k may be set to any non-negative value and should be set to a sufficiently large number to effectively guarantee the discovery of ***PWE*** in less than k iterations. If ***PWE*** is discovered in less than k iterations a random "password" can be used in subsequent iterations to further obfuscate the true cost of discovering ***PWE***.

NOTE—The probability that one requires more than n iterations of the "hunting and pecking" loop to find ***PWE*** is roughly $(r/2p)^n$, which rapidly approaches zero (0) as n increases.

Algorithmically this process is described as follows:

```
found = 0;
counter = 1
z = len(p)
base = password
do {
     pwd-seed = H(MAX(STA-A-MAC, STA-B-MAC) || MIN(STA-A-MAC, STA-B-MAC),
               base || counter)
     pwd-value = KDF-z(pwd-seed, “SAE Hunting and Pecking”, p)
    if (pwd-value < p)
    then
              x = pwd-value
              if the equation $y^2 = x^3 + ax + b$ modulo p has a solution y
              then
                     if (found = 0)
```

```
            then
              determine a solution, $y$, to be the equation $y^2 = x^3 + ax + b$ modulo $p$
              if LSB($pwd\text{-}seed$) = LSB($y$)
              then
                  $\boldsymbol{PWE} = (x, y)$
              else
                  $\boldsymbol{PWE} = (x, p - y)$
              fi
              $found = 1$
            else
              $base$ = new-random-number
            fi
        fi
    fi
    $counter = counter + 1$
} while (($counter <= k$) or ($found$=0))
```

11.3.4.3 Finite field cryptography (FFC) groups

11.3.4.3.1 FFC group definition

FFC groups used by SAE are defined by the triple (p, $\boldsymbol{G}$, r), where p is a prime number, $\boldsymbol{G}$ is a generator, and r is the prime order of $\boldsymbol{G}$ modulo p. An element, $\boldsymbol{B}$, in an FFC group satisfies $\boldsymbol{B} = \boldsymbol{G}^i$ modulo p for some integer i. This special property differentiates elements from scalars, even though both elements and scalars can be represented as non-negative integers less than the prime modulus p. The notation convention of 11.3.4 signifies this difference between an element and a scalar in an FFC group. The identity element for an FFC group is the value one (1) modulo p.

The element operation in an FFC group is modular multiplication of two elements of this group resulting in a third element of this group. For example, the element $\boldsymbol{X}$ is multiplied by the element $\boldsymbol{Y}$ to product the element $\boldsymbol{Z}$:

$$\boldsymbol{Z} = (\boldsymbol{XY}) \text{ modulo } p = \text{elem-op}(\boldsymbol{X},\boldsymbol{Y})$$

The scalar operation in an FFC group is modular exponentiation of an element of this group by a scalar resulting in a second element of this group. For example, the point $\boldsymbol{Y}$ is raised to the power x to produce the element $\boldsymbol{Z}$:

$$\boldsymbol{Z} = \boldsymbol{Y}^x \text{ modulo } p = \text{scalar-op}(x,\boldsymbol{Y})$$

Some FFC groups in the IANA repository are based on *safe primes*, i.e., a prime, p, of the form $p = 2q + 1$, where q is also a prime number. For these FFC groups, the group generated by $\boldsymbol{G}$ always has order $r = (p - 1)/2$ and thus is uniquely derived from context. For other FFC groups, the parameter r shall be explicitly stated as part of the domain parameters.

The inverse operation in a FFC group is modular inversion of an element of this group producing a second element in this group. An element $\boldsymbol{Z}$ is the inverse of a second element $\boldsymbol{X}$ of this group if their modular product is the identity element of the FFC group. In other words:

$$\text{elem-op}(\boldsymbol{X}, \text{inverse}(\boldsymbol{X})) = 1 \text{ modulo } p$$

In contrast to ECC groups, FFC groups do not need a mapping function that maps an element of the FFC group to an integer (since those elements are already non-negative integers less than the prime number, p).

However, for sake of uniform protocol definition, function F with FFC groups is defined as the identity function—i.e., if x is an element of the FFC group then $F(x) = x$.

11.3.4.3.2 Generation of the Password Element with FFC groups

The Password Element of an FFC group (***PWE***) shall be generated in a random hunt-and-peck fashion similar to the technique for an ECC group. The password and a counter, represented as a single octet and initially set to one (1), are used with the two peer identities to generate a password seed. The password seed shall then be stretched using the key derivation function (KDF) from 11.6.1.7.2 to a length equal to the bit length of the prime number, p, from the group domain parameters with the Label being the string "SAE Hunting and Pecking" and the Content being the prime number. If the resulting password value is greater than or equal to the prime number, the counter shall be incremented, a new password seed shall be derived, and the hunting-and-pecking shall continue. Otherwise, it shall be raised to the power $(p – 1) / r$ (where p is the prime number and r is the order) modulo the prime number to produce a candidate ***PWE***. If the candidate ***PWE*** is greater than one (1), the candidate ***PWE*** becomes the ***PWE***; otherwise, the counter shall be incremented, a new password seed shall be derived, and the hunting-and-pecking shall continue.

Algorithmically this process is described as follows:

```
found = 0;
counter = 1
z = len(p)
do {
    pwd-seed = H(MAX(STA-A-MAC, STA-B-MAC) || MIN(STA-A-MAC, STA-B-MAC),
            password || counter)
    pwd-value = KDF-z(pwd-seed, "SAE Hunting and Pecking", p)
    if (pwd-value < p)
    then
        PWE = $pwd\text{-}value^{(p-1)/r}$ modulo p
        if (PWE > 1)
        then
            found = 1
        fi
    fi
    counter = counter + 1
} while (found=0)
```

11.3.5 SAE protocol

11.3.5.1 Message exchanges

The protocol consists of two message exchanges, a commitment exchange and a confirmation exchange. The commitment exchange is used to force each party to the exchange to commit to a single guess of the password. The confirmation exchange is used to prove that the password guess was correct. Authentication frames are used to perform these exchanges (see 8.3.3.11 and 11.3.7.3). The rules for performing these exchanges are specified by the finite state machine in 11.3.8.

When a party has sent its message in the commit exchange it is said to have *committed* and when it has sent its message in the confirmation exchange it has *confirmed*. The following rules are ascribed to the protocol:

— A party may *commit* at any time

— A party *confirm*s after it has *committed* and its peer has *committed*

— A party *accept*s authentication after a peer has *confirmed*

— The protocol successfully *terminates* after each peer has *accepted*

11.3.5.2 PWE and secret generation

Prior to beginning the protocol message exchange, the secret element ***PWE*** and two secret values are generated. First, a group is selected, either the most preferred group if the STA is initiating SAE to a peer, or the group from a received Commit Message if the STA is responding to a peer. The ***PWE*** shall be generated for that group (according to 11.3.4.2.2 or 11.3.4.3.2, depending on whether the group is ECC or FFC, respectively) using the identities of the two STAs and the configured password.

After generation of the ***PWE***, each STA shall generate a secret value, *rand*, and a temporary secret value, *mask*, each of which shall be chosen randomly such that $1 < rand < r$ and $1 < mask < r$, where *r* is the (prime) order of the group. The values *rand* and *mask* shall be random numbers produced from a quality random number generator. These values shall never be reused on distinct protocol runs.

11.3.5.3 Construction of a Commit Message

A Commit Message consists of a scalar and an element that shall be produced using the ***PWE*** and secrets generated in 11.3.5.2, as follows:

commit-scalar = (*rand* + *mask*) modulo *r*
COMMIT-ELEMENT = inverse(scalar-op(*mask*, ***PWE***))

This message shall be transmitted to the peer as described in 11.3.7. The temporary secret *mask* may be destroyed at this point.

11.3.5.4 Processing of a peer’s Commit Message

Upon receipt of a peer’s Commit Message both the scalar and element shall be verified.

If the scalar value is greater than zero (0) and less than the order, *r*, of the negotiated group, scalar validation succeeds; otherwise, it fails. Element validation depends on the type of group. For FFC groups, the element shall be an integer greater than zero (0) and less than the prime number *p*, and the scalar operation of the element and the order of the group, *r*, shall equal one (1) modulo the prime number *p*. If either of these conditions does not hold, element validation fails; otherwise, it succeeds. For ECC groups, both the x- and y-coordinates of the element shall be non-negative integers less than the prime number *p*, and the two coordinates shall produce a valid point on the curve satisfying the group’s curve definition, not being equal to the “point at the infinity.” If either of those conditions does not hold, element validation fails; otherwise, element validation succeeds.

If either scalar validation or element validation fails, the STA shall reject the peer’s authentication. If both the scalar and element from the peer’s Commit Message are successfully validated, a shared secret element, *K*, shall be derived using the scalar and element (*peer-commit-scalar* and ***PEER-COMMIT-ELEMENT***, respectively) from the peer’s Commit Message and the STA’s secret value.

K = scalar-op(*rand*, (elem-op(scalar-op(*peer-commit-scalar*, ***PWE***),
PEER-COMMIT-ELEMENT)))

If the shared secret element, ***K***, is the identity element for the negotiated group (the value one for an FFC group or the point-at-infinity for an ECC group) the STA shall reject the peer’s authentication. Otherwise, a secret value, *k*, shall be computed as:

k = F(***K***)

The entropy of *k* shall then be extracted using H to produce *keyseed*. The key derivation function from 11.6.1.7.2 shall then be used to derive a key confirmation key, KCK, and a pairwise master key, PMK, from *keyseed*. When used with AKMs 8 or 9, the salt shall consist of thirty-two (32) octets of the value zero (0) (indicated below as <0>32) and both the KCK and PMK shall be 256-bits in length. Use of other AKMs require definition of the lengths of the salt, the KCK, and the PMK.

keyseed = H(<0>32, *k*)
KCK || *PMK* = KDF-512(*keyseed*, “SAE KCK and PMK”,
(*commit-scalar* + *peer-commit-scalar*) modulo *r*)

The PMK identifier is defined as follows:

PMKID = L((*commit-scalar* + *peer-commit-scalar*) modulo *r*, 0, 128)

11.3.5.5 Construction of a Confirm Message

A peer generates a Confirm Message by passing the KCK, the current value of the *send-confirm* counter (see 8.4.1.37), the scalar and element from the sent Commit Message, and the scalar and element from the received Commit Message to the confirmation function CN.

confirm = CN(KCK*, send-confirm, commit-scalar,* ***COMMIT-ELEMENT***, *peer-commit-scalar,*
PEER-COMMIT-ELEMENT)

The message shall be transmitted to the peer as described in 11.3.7.

11.3.5.6 Processing of a peer’s Confirm Message

Upon receipt of a peer’s Confirm Message a *verifier* is computed, which is the expected value of the peer’s confirmation, *peer-confirm*, extracted from the received Confirm Message. The *verifier* is computed by passing the KCK, the peer’s send-confirm counter from the received Confirm Message (see 8.4.1.37), the scalar and element from the received Commit Message, and scalar and element from the sent Commit Message to the confirmation function CN.

verifier = CN(KCK*, peer-send-confirm, peer-commit-scalar,* ***PEER-COMMIT-ELEMENT***,
commit-scalar, ***COMMIT-ELEMENT***)

If the *verifier* equals *peer-confirm,* the STA shall accept the peer’s authentication and set the lifetime of the PMK to the value dot11RSNAConfigPMKLifetime. If the *verifier* differs from the *peer-confirm,* the STA shall reject the peer’s authentication and destroy the PMK.

11.3.6 Anti-clogging tokens

A STA is required to do a considerable amount of work upon receipt of a Commit Message. This opens up the possibility of a distributed denial-of-service attack by flooding a STA with bogus Commit Messages from forged MAC addresses. To prevent this from happening, a STA shall maintain an *Open* counter in its SAE state machine indicating the number of open and unfinished protocol instances (see 11.3.5.1). When that counter hits or exceeds dot11RSNASAEAntiCloggingThreshold, the STA shall respond to each Commit Message with a rejection that includes an Anti-Clogging Token statelessly bound to the sender of the Commit Message. The sender of the Commit Message shall then include this Anti-Clogging Token in a subsequent Commit Message.

The Anti-Clogging Token is a variable-length value that statelessly binds the MAC address of the sender of a Commit Message. The length of the Anti-Clogging Token needs not be specified because the generation and processing of the Anti-Clogging Token is solely up to one peer. To the other peer in the SAE protocol, the Anti-Clogging Token is merely an opaque blob whose length is insignificant. It is suggested that an Anti-Clogging Token not exceed 256 octets.

NOTE—A suggested method for producing Anti-Clogging Tokens is to generate a random secret value each time the state machine variable hits dot11RSNASAEAntiCloggingThreshold and pass that secret and the MAC address of the sender of the Commit Message to the random function H to generate the token.

As long as the state machine variable *Open* is greater than or equal to dot11RSNASAEAntiCloggingThreshold all Commit Messages that do not include a valid Anti-Clogging Token shall be rejected with a request to repeat the Commit Message and include the token (see 11.3.5.1).

Since the Anti-Clogging Token is of fixed size and the size of the *peer-commit-scalar* and ***PEER-COMMIT-ELEMENT*** are inferred from the finite cyclic group being used, it is straightforward to determine whether a received Commit Message includes an Anti-Clogging Token or not.

Encoding of the Anti-Clogging Token and its placement with respect to the *peer-commit-scalar* and ***PEER-COMMIT-ELEMENT*** is described in 11.3.7.4.

11.3.7 Framing of SAE

11.3.7.1 General

Commit Messages and Confirm Messages are sent and received by a SAE protocol using IEEE 802.11 Authentication frames.

11.3.7.2 Data type conversion

11.3.7.2.1 General

This protocol requires elements in finite cyclic groups to be converted to octet strings prior to transmission and back again upon receipt. To convert an element into an octet string, the first step is to represent the element in integer format and then employ an integer-to-octet string conversion prior to transmission. To convert an octet string into an element requires an octet string to integer conversion and then representing the integer(s) as an element.

11.3.7.2.2 Integer to octet string conversion

An integer, x, shall be converted into an octet string of length m such that $2^{8m} > x$ by first representing x in its binary form and then converting the result to an octet-string.

Given x, m, represent x as a sequence of x_{m-i} base 2^8:

$$x = x_{m-1} \times 2^{8(m-1)} + x_{m-2} \times 2^{8(m-2)} + \ldots + x_1 \times 2^8 + x_0$$

then let the octet M_i have the value x_i for $0 \le i \le m-1$ and the octet string shall be $M_{m-1} \| M_{m-2} \| \ldots \| M_1 \| M_0$ where || symbolizes concatenation.

11.3.7.2.3 Octet string to integer conversion

An octet string shall be converted into an integer by viewing the octet string as the base 2^8 representation of the integer.

$$x = \sum_{i=1}^{m} 2^{8(m-i)} \times M_{m-i}$$

11.3.7.2.4 Element to octet string conversion

For ECC groups, each element, except the "point at infinity," is a point on the elliptic curve satisfying the curve equation and consists of two components: an x-coordinate and a y-coordinate. To convert this point to an octet string, each component shall be treated as an integer and converted into an octet string whose length is the smallest integer *m* such that $2^{8m} > p$, where *p* is the prime number specified by the elliptic curve domain parameters, according to 11.3.7.2.2. The point shall be represented as the concatenation of the x-coordinate and the y-coordinate, each represented as an octet string of length *m* octets, and is 2*m* octets long.

For FFC groups each element is a non-negative integer less than the prime number *p* specified by the FFC domain parameters. To convert this element into an octet string, it shall be treated directly as an integer and converted into an octet string whose length is the smallest integer *m* such that $2^{8m} > p$, where *p* is the prime number specified by the domain parameters, according to 11.3.7.2.2.

11.3.7.2.5 Octet string to element conversion

To convert an octet string into a point on an elliptic curve it is necessary to divide it into two octet strings of equal length *m*. If the length of the octet string does not evenly divide by two, conversion shall fail. Each octet string of length *m* shall be converted to an integer according to 11.3.7.2.3. The first octet string conversion produces an integer that becomes the x-coordinate of the point and the second octet string conversion produces an integer that becomes the y-coordinate of the point. If either integer equals zero (0) or is greater than or equal to *p*, the prime from the elliptic curve domain parameters, conversion shall fail. If the resulting (x, y) point does not satisfy the equation of the curve, or produces the "point at infinity," conversion shall fail.

To convert an octet string into an element in a prime modulus group the octet string shall be converted into an integer according to 11.3.7.2.3 and the integer shall be used directly as the group element.

11.3.7.3 Authentication transaction sequence number for SAE

A Commit Message shall use Authentication Transaction Sequence Number one (1). A Confirm Message shall use Authentication Transaction Sequence Number two (2).

11.3.7.4 Encoding and decoding of Commit Messages

A Commit Message shall be encoded as an IEEE 802.11 Authentication frame with an Authentication Algorithm of three (3), a Transaction Sequence Number of one (1) and a Status Code of zero (0). Nonzero status codes indicate a rejection of a peer's Commit Message and are described in 11.3.7.6.

A Commit Message shall consist of a Finite Cyclic Group field (8.4.1.42) indicating the desired group, a Scalar field (8.4.1.39) containing the scalar, and an Element field containing the element (8.4.1.40). If the Commit Message is in response to an Anti-Clogging Token request (see 11.3.7.6), the Anti-Clogging Token is present (see 8.4.1.38).

When transmitting a Commit Message, the scalar and element shall be converted to octet strings and placed in the Scalar field and Element field, respectively. The scalar shall be treated as an integer and converted into an octet string of length *m* such that $2^{8m} > r$, where *r* is the order of the group, according to 11.3.7.2.2, and the element shall be converted into (an) octet string(s) according to 11.3.7.2.4. When receiving a Commit Message the component octet strings in the Scalar field and Element field shall be converted into a scalar and element, respectively, according to 11.3.7.2.3 and 11.3.7.2.5, respectively.

11.3.7.5 Encoding and decoding of Confirm Messages

A Confirm Message shall be encoded as an IEEE 802.11 Authentication frame with an Authentication Algorithm of three (3), a Transaction Sequence Number of two (2) and a Status Code of zero (0). Nonzero status codes indicate rejection of a peer's Confirm Message and are described in 11.3.7.6.

A Confirm Message shall consist of a Send-Confirm field (8.4.1.37) and a Confirm field (8.4.1.41) containing the output of the random function as described in 11.3.5.5. When transmitting a Confirm Message the output of the random function shall be treated as an integer and converted into an octet string of length *m*, where *m* is the block size of the random function, according to 11.3.7.2.2 and placed in the Confirm field. When receiving a Confirm Message, the octet string in the Confirm field shall be converted into an integer representing the peer's Confirm according to 11.3.7.2.3.

11.3.7.6 Status codes

A Commit Message with a nonzero status code shall indicate that a peer rejects a previously sent Commit Message. An unsupported finite cyclic group is indicated with a status code of 77, "Authentication is rejected because the offered finite cyclic group is not supported." An Anti-Clogging Token is requested by transmitting a Commit Message with a status code of 76, "Anti-Clogging Token Requested," with the Anti-Clogging Token occupying the Token field of the Authentication frame.

A Confirm Message, with a nonzero status code, shall indicate that a peer rejects a previously sent Confirm Message. A Confirm Message that was not successfully verified is indicated with a status code of fifteen (15), "Authentication rejected; the response to the challenge failed."

11.3.8 SAE finite state machine

11.3.8.1 General

The protocol is instantiated by the finite state machine in Figure 11-4. Each instance of the protocol is identified by a tuple consisting of the local MAC address and the peer MAC address. The model in which SAE is defined consists of a parent process, managed by the SME, which receives messages, and dispatches them to the appropriate protocol instance, also managed by the SME. The parent process manages a database of protocol instances indexed by the peer identity. Protocol instances maintain state, receive events from the parent process, send events to itself, and output data.

NOTE—Figure 11-4 does not show all state machine transitions. A full description of the SAE finite state machine is in 11.3.8.6.2 to 11.3.8.6.6.

The parent process instantiates protocol instances upon receipt of SAE messages and initiation by SME. The parent process also maintains a counter of the number of protocol instances created.

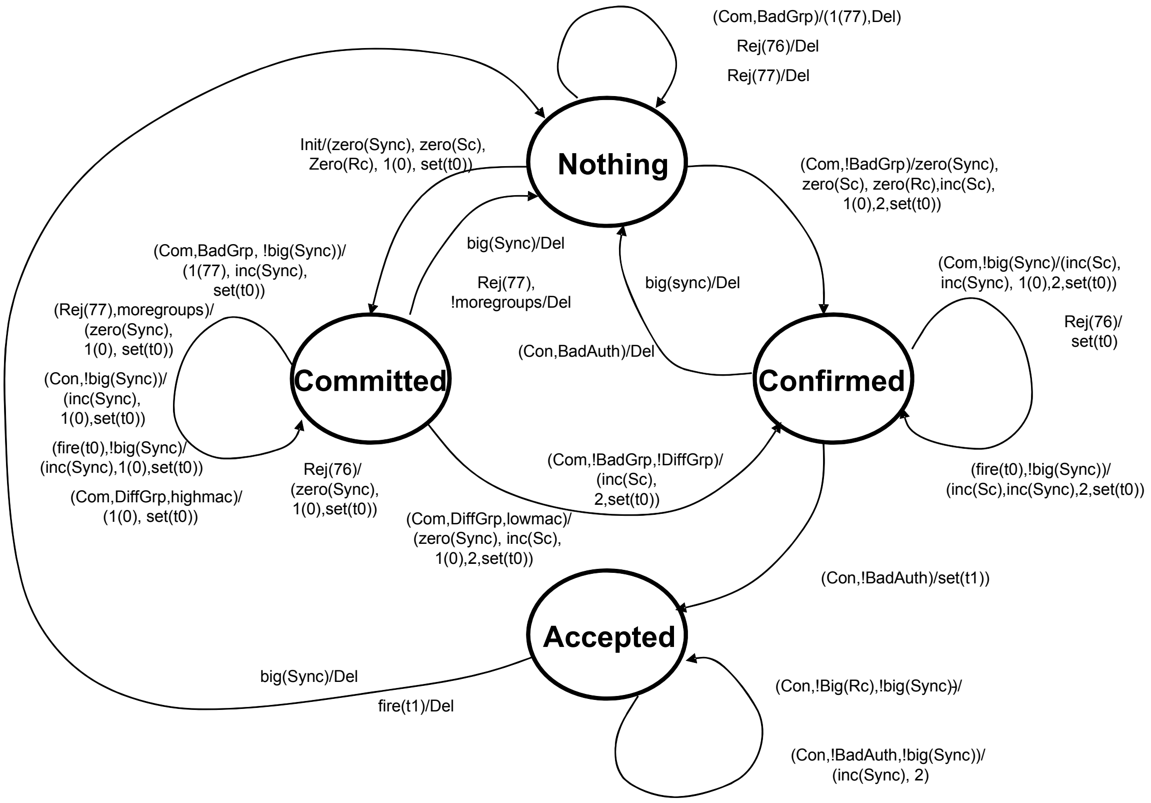


Figure 11-4—SAE finite state machine

11.3.8.2 States

11.3.8.2.1 Parent process states

The parent process is in a continuous quiescent state.

11.3.8.2.2 Protocol instance states

Each protocol instance is in one of the following four (4) states:

- *Nothing*—The *Nothing* state represents the initial state of a freshly allocated protocol instance or the terminal state of a soon-to-be deallocated protocol instance. Freshly created protocol instances will immediately transition out of *Nothing* state depending on the reason for their creation. Protocol instances that transition into *Nothing* state will immediately be destroyed with their state zeroed and returned to the memory pool.
- *Committed*—In the *Committed* state, the finite state machine has sent a Commit Message and is awaiting a Commit Message and a Confirm Message from the peer.
- *Confirmed*—In the *Confirmed* state, the finite state machine has sent both a Commit Message and a Confirm Message and received a Commit Message. It awaits a Confirm Message.
- *Accepted*—In the *Accepted* state, the protocol instance has both sent and received a Commit Message and a Confirm Message and the protocol instance has finished.

11.3.8.3 Events and output

11.3.8.3.1 Parent process events and output

The parent process receives events from three (3) sources: the SME, protocol instances, and received frames.

The SME signals the following events to the parent SAE process:

- *Initiate*—An *Initiate* event is used to instantiate a protocol instance to begin SAE with a designated peer.
- *Kill*—A *Kill* event is used to remove a protocol instance with a designated peer.

Protocol instances send the following events to the SAE parent process:

- *Fail*—The peer failed to be authenticated.
- *Auth*—The peer was successfully authenticated.
- *Del*—The protocol instance has had a fatal event.

Receipt of frames containing SAE messages signals the following events to the SAE parent process:

- *IEEE 802.11 Authentication frame with Transaction Sequence number 1*—This event indicates that a Commit Message has been received from a peer STA.
- *IEEE 802.11 Authentication frames with Transaction Sequence number 2*—This event indicates that a Confirm Message has been received from a peer STA.

The parent process generates IEEE 802.11 Authentication frames with Authentication transaction sequence 1 and a Status of 76 indicating rejection of an Authentication attempt because an Anti-Clogging Token is required.

11.3.8.3.2 Protocol instance events and output

The protocol instance receives events from the parent SAE process.

- *Com*—Indicates receipt of a Commit Message (Authentication transaction sequence number 1) with a status of zero (0).
- *Con*—Indicates receipt of a Confirm Message (Authentication transaction sequence number 2) with a status of zero (0).
- *Init*—Indicates that the protocol instance should begin negotiation with a specified peer.
- *Rej(N)*—Indicates receipt of a rejected Commit Message with status *N*.

In addition, protocol instances receive *fire(X)* events indicating expiry of timer *X*. Upon expiry of a timer and generation of a *fire()* event, the expired timer is not reset.

The protocol instance generates output from the following events:

- *1(N)*—Indicates generation of a Commit Message (Authentication transaction sequence number 1) with status *N*.
- *2*—Indicates generation of a Confirm Message (Authentication transaction sequence number 2).

11.3.8.4 Timers

The parent SAE process does not use timers. Each protocol instance can set timers that result in *fire()* events to be sent to itself. The following timers can be set:

- t0—A retransmission timer.

- t1—A PMK expiry timer.

Timers are set by the protocol instance issuing a *set()* for the particular timer.

11.3.8.5 Variables

11.3.8.5.1 Parent process variables

The parent SAE process maintains a counter, *Open*, which indicates the number of protocol instances in either *Committed* or *Confirmed* state. When the parent SAE process starts up, *Open* is set to zero (0).

The parent process maintains a database of protocol instances.

NOTE—Depending on how Anti-Clogging Tokens (see 11.3.6) are constructed, the parent SAE process might also maintain a random secret used for token creation.

11.3.8.5.2 Protocol instance variables

Each protocol instance maintains the following three variables:

- *Sync*—The number of state resynchronizations that have occurred.
- *Sc*—The number of Confirm messages that have been sent. This is the send-confirm counter used in the construction of Confirm messages (see 11.3.5.5).
- *Rc*—The received value of the *send-confirm* counter in the last received Confirm Message. In other words, this is the value of the peer's send-confirm counter.

Function *zero(X)* assigns the value zero (0) to the variable *X*, *inc(X)* increments the variable *X*, and *big(X)* indicates that the variable *X* has exceeded a maximum value.

In addition, protocol instances maintain the following six indicators that are not maintained as state variables but, instead, indicate the cause of certain behavior.

- *BadGrp*—The group specified in a Commit Message is not supported.
- *DiffGrp*—The group specified in a Commit Message is supported but differs from the one offered.
- *BadConf*—The contents of a confirm frame were incorrect.
- *highmac*—The peer identity is numerically less than the local identity.
- *lowmac*—The peer identity is numerically greater than the local identity.
- *moregroups*—There are finite cyclic groups in the configuration that have not been offered to the peer.

A negative indication is shown with an exclamation point (!)—e.g., "the group specified in a Commit Message is supported" would be !BadGrp, which is read as "not BadGrp."

11.3.8.6 Behavior of state machine

11.3.8.6.1 Parent process behavior

For any given peer identity, there shall be only one protocol instance in *Committed* or *Confirmed* state. Similarly, for any given peer identity, there shall be only one protocol instance in *Accepted* state.

The parent process creates protocol instances based upon different actions. Creating a protocol instance entails allocation of state necessary to maintain the protocol instance state machine, putting the protocol instance in *Nothing* state, incrementing the *Open* counter, and inserting the protocol instance into its database indexed by the MAC address of the peer with whom the protocol instance will communicate.

The parent process also destroys protocol instances by zeroing out the state of the protocol instance and returning it to the memory pool.

Upon receipt of an *Initiate* event, the parent process shall check whether there exists a protocol instance for the peer MAC address (from the *Init* event) in either *Committed* or *Confirmed* state. If there is, the *Initiate* event shall be ignored. Otherwise, a protocol instance shall be created, and an *Init* event shall be sent to the protocol instance.

Upon receipt of a *Kill* event, the parent process shall destroy all protocol instances indexed by the peer MAC address (from the *Kill* event) in its database. For each protocol instance in *Committed* or *Confirmed* state, the *Open* counter shall be decremented.

Upon receipt of a *Sync, Del,* or *Fail* event from a protocol instance, the parent process shall decrement the *Open* counter and destroys the protocol instance.

Upon receipt of an *Auth* event from a protocol instance, the parent process shall decrement the *Open* counter. If another protocol instance exists in the database indexed by the same peer identity as the protocol instance that sent the *Auth* event, the other protocol instance shall be destroyed.

Upon receipt of a Commit Message, the parent process checks whether a protocol instance for the peer MAC address exists in the database. If one does, and it is in either *Committed* state or *Confirmed* state the frame shall be passed to the protocol instance. If one does and it is in Authenticated state, the scalar in the received frame is checked against the *peer-scalar* used in authentication of the existing protocol instance (in Authenticated state). If it is identical, the frame shall be dropped. If not, the parent process checks the value of *Open*. If *Open* is greater than dot11RSNASAEAntiCloggingThreshold, the parent process shall check for the presence of an Anti-Clogging Token. If an Anti-Clogging Token exists and is correct, the parent process shall create a protocol instance. If the Anti-Clogging Token is incorrect, the frame shall be silently discarded. If Open is greater than dot11RSNASAEAntiCloggingThreshold and there is no Anti-Clogging Token in the received frame, the parent process shall construct a response as an IEEE 802.11 Authentication frame with Authentication sequence number one (1), Status code 76, and the body of the frame consisting of an Anti-Clogging Token (see 11.3.6). If *Open* is not greater than dot11RSNASAEAntiCloggingThreshold, the parent process shall create a protocol instance and the frame shall be sent to the protocol instance as a *Com* event.

Upon receipt of a Confirm Message, the parent process checks whether a protocol instance for the peer MAC address (as indicated by the SA in the received frame) exists in the database. If there is a single protocol instance, the frame shall be passed to it as a *Con* event. If there are two (2) protocol instances indexed by that peer MAC address, the frame shall be passed, as a *Con* event, to the protocol instance that is not in *Accepted* state. If there are no protocol instances indexed by that peer MAC address, the frame shall be dropped.

11.3.8.6.2 Protocol instance behavior—General

State machine behavior is illustrated in Figure 11-4. The protocol instance receives events from the parent process and from itself. It generates SAE messages that are transmitted to a peer and sends events to itself and the parent process.

The semantics of the state diagram are “occurrence/behavior” where “occurrence” is a comma-separated list of events and/or indicators, or the special symbol “—” indicating no occurrence; and, “behavior” is a comma-separated list of outputs and/or functions, or the special symbol “—” indicating no behavior.

When the state machine calls for the t0 (retransmission) timer to be set, it shall be set to the value of dot11RSNASAERetransPeriod. When the state machine calls for the t1 (key expiry) timer to be set, it shall be set to the value of dot11RSNAConfigPMKLifetime.

11.3.8.6.3 Protocol instance behavior - Nothing state

In *Nothing* state a protocol instance has just been allocated.

Upon receipt of an *Init* event, the protocol instance shall zero its *Sync* variable, *Rc*, and *Sc* variables, select a group from local configuration and generate the ***PWE*** and the secret values according to 11.3.5.2, generate a Commit Message (see 11.3.5.3), and set its t0 (retransmission) timer. The protocol instance transitions into *Committed* state.

Upon receipt of a *Com* event, the protocol instance shall check the Status of the Authentication frame. If the Status code is nonzero, the frame shall be silently discarded and a *Del* event shall be sent to the parent process. Otherwise, the frame shall be processed by first checking the finite cyclic group field to see if the requested group is supported. If not, *BadGrp* shall be set and the protocol instance shall construct and transmit a an Authentication frame with Status code 77 indicating rejection with the finite cyclic group field set to the rejected group, and shall send the parent process a *Del* event. If the group is supported, the protocol instance shall zero the *Sc* and *Rc* counters and it shall generate the ***PWE*** and the secret values according to 11.3.5.2. It shall then process the received Commit Message (see 11.3.5.4). If validation of the received Commit Message fails, the protocol instance shall send a Del event to the parent process; otherwise, it shall construct and transmit a Commit Message (see 11.3.5.3) followed by a Confirm Message (see 11.3.5.5). The *Sync* counter shall be set to zero and the t0 (retransmission) timer shall be set. The protocol instance transitions to *Confirmed* state.

NOTE—A protocol instance in *Nothing* state will never receive a Confirm Message due to state machine behavior of the parent process.

11.3.8.6.4 Protocol instance behavior - Committed state

In *Committed* state, a protocol instance has sent its peer a Commit Message but has yet to receive (and accept) anything.

Upon receipt of a *Com* event, the t0 (retransmission) timer shall be cancelled. Then the following is performed:

- The protocol instance shall check the Status code of the Authentication frame. If the Status code is 76, a new Commit Message shall be constructed with the Anti-Clogging Token from the received Authentication frame, and the *commit-scalar* and ***COMMIT-ELEMENT*** previously sent. The new Commit Message shall be transmitted to the peer, *Sync* shall be zeroed, and the t0 (retransmission) timer shall be set.
- If the Status code is 77, the protocol instance shall check the finite cyclic group field being rejected. If the rejected group does not match the last offered group the protocol instance shall silently discard the message and set the t0 (retransmission) timer. If the rejected group matches the last offered group, the protocol instance shall choose a different group and generate the ***PWE*** and the secret values according to 11.3.5.2; it then generates and transmits a new Commit Message to the peer, zeros *Sync*, sets the t0 (retransmission) timer, and remains in *Committed* state. If there are no other groups to choose, the protocol instance shall send a *Del* event to the parent process and transitions back to *Nothing* state.
- If the Status is some other nonzero value, the frame shall be silently discarded and the t0 (retransmission) timer shall be set.
- If the Status is zero, the finite cyclic group field is checked. If the group is not supported, *BadGrp* shall be set and the value of *Sync* shall be checked.
  - If *Sync* is greater than dot11RSNASAESync, the protocol instance shall send a *Del* event to the parent process and transitions back to *Nothing* state.
  - If *Sync* is not greater than dot11RSNASAESync, *Sync* shall be incremented, a Commit Message with Status code equal to 77 indicating rejection, and the Algorithm identifier set to

the rejected algorithm, shall be sent to the peer, the t0 (retransmission) timer shall be set and the protocol instance shall remain in *Committed* state.

- If the group is supported but does not match that used when the protocol instance constructed its Commit Message, *DiffGrp* shall be set and the local identity and peer identity shall be checked.
  - The mesh STA, with the numerically greater of the two MAC addresses, drops the received Commit Message, retransmits its last Commit Message, and shall set the t0 (retransmission) timer and remain in *Committed* state.
  - The mesh STA, with the numerically lesser of the two MAC addresses, zeros *Sync*, shall increment *Sc*, choose the group from the received Commit Message, generate new ***PWE*** and new secret values according to 11.3.5.2, process the received Commit Message according to 11.3.5.4, generate a new Commit Message and Confirm Message, and shall transmit the new Commit and Confirm to the peer. It shall then transition to *Confirmed* state.
- If the group is supported and matches that used when the protocol instance constructed its Commit Message, the protocol instance checks the *peer-commit-scalar* and ***PEER-COMMIT-ELEMENT*** from the message. If they match those sent as part of the protocol instance's own Commit Message, the frame shall be silently discarded (because it is evidence of a reflection attack) and the t0 (retransmission) timer shall be set. If the received element and scalar differ from the element and scalar offered, the received Commit Message shall be processed according to 11.3.5.4, the *Sc* counter shall be incremented (thereby setting its value to one), the protocol instance shall then construct a Confirm Message, transmit it to the peer, and set the t0 (retransmission) timer. It shall then transition to *Confirmed* state.

If the t0 (retransmission) timer fires, the value of the *Sync* counter is checked. If *Sync* is greater than dot11RSNASAESync, the protocol instance shall send a *Del* event to the parent process and transition back to *Nothing* state. If *Sync* is not greater than dot11RSNASAESync, the *Sync* counter shall be incremented, the last message sent shall be sent again, and the t0 (retransmission) timer shall be set.

Upon receipt of a *Con* event, the t0 (retransmission) timer shall be cancelled. Then the protocol instance checks the value of *Sync*. If it is greater than dot11RSNASAESync, the protocol instance shall send a *Del* event to the parent process and transition back to *Nothing* state. If *Sync* is not greater than dot11RSNASAESync, the protocol instance shall increment *Sync*, transmit the last Commit Message sent to the peer, and set the t0 (retransmission) timer.

11.3.8.6.5 Protocol instance behavior - Confirmed state

In *Confirmed* state, a protocol instance has sent its peer a Commit Message and Confirm Message. It has received a Commit Message from its peer.

Rejection frames received in Confirmed state shall be silently discarded.

Upon receipt of a *Com* event, the t0 (retransmission) timer shall be cancelled. If the Status is nonzero, the frame shall be silently discarded, the t0 (retransmission) timer set, and the protocol instance shall remain in the *Confirmed* state. If *Sync* is greater than dot11RSNASAESync, the protocol instance shall send the parent process a *Del* event and transitions back to *Nothing* state. If *Sync* is not greater than dot11RSNASAESync, the protocol instance shall verify that the finite cyclic group is the same as the previously received Commit frame. If not, the frame shall be silently discarded. If so, the protocol instance shall increment *Sync*, increment *Sc*, and transmit its Commit and Confirm (with the new *Sc* value) messages. It then shall set the t0 (retransmission) timer.

Upon receipt of a *Con* event, the t0 (retransmission) timer shall be cancelled and the Confirm Message shall be processed according to 11.3.5.6. If processing is successful and the Confirm Message has been verified, the *Rc* variable shall be set to the send-confirm portion of the frame, *Sc* shall be set to the value $2^{16} – 1$, the t1 (key expiry) timer shall be set, and the protocol instance shall transition to *Accepted* state.

If the t0 (retransmission) timer fires, the value of the *Sync* counter shall be checked. If *Sync* is greater than dot11RSNASAESync, the protocol instance shall send a *Del* event to the parent process and transition back to *Nothing* state. If *Sync* is not greater than dot11RSNASAESync, the *Sync* counter shall be incremented, *Sc* shall be incremented, and the protocol instance shall create a new Confirm (with the new *Sc* value) Message, transmit it to the peer, and set the t0 (retransmission) timer.

11.3.8.6.6 Protocol instance behavior - Accepted state

In *Accepted* state, a protocol instance has sent a Commit Message and a Confirm Message to its peer and received a Commit Message and Confirm Message from the peer. Unfortunately, there is no guarantee that the final Confirm Message sent by the STA was received by the peer.

Upon receipt of a *Con* event, the *Sync* counter shall be checked. If the value is greater than dot11RSNASAESync, the protocol instance shall send a *Del* event to the parent process and shall transition to *Nothing* state. If the value of *Sync* is not greater than dot11RSNASAESync, the value of send-confirm shall be checked. If the value is not greater than *Rc* or is equal to $2^{16} – 1$, the received frame shall be silently discarded. Otherwise, the Confirm portion of the frame shall be checked according to 11.3.5.6. If the verification fails, the received frame shall be silently discarded. If the verification succeeds, the *Rc* variable shall be set to the send-confirm portion of the frame, the *Sync* shall be incremented and a new Confirm Message shall be constructed (with *Sc* set to $2^{16} – 1$) and sent to the peer. The protocol instance shall remain in *Accepted* state.

If the t1 (key expiry) timer fires, the protocol instance shall send the parent process a *Del* event and transition to *Nothing* state.

11.4 RSNA confidentiality and integrity protocols

11.4.1 Overview

This standard defines two RSNA data confidentiality and integrity protocols: TKIP and CCMP. This standard defines one integrity protocol for management frames: BIP.

Implementation of TKIP is optional for an RSNA and used only for the protection of data frames. A design aim for TKIP was that the algorithm should be implementable within the capabilities of most devices supporting only WEP, so that many such devices would be field-upgradable by the supplier to support TKIP.

BIP is a mechanism that is used only when management frame protection is negotiated. BIP provides integrity protection for group addressed robust management frames. BIP is only used to protect management frames within the BSS.

11.4.2 Temporal Key Integrity Protocol (TKIP)

11.4.2.1 TKIP overview

11.4.2.1.1 General

The TKIP is a cipher suite enhancing WEP on pre-RSNA hardware. TKIP modifies WEP as follows:

a) A transmitter calculates a keyed cryptographic message integrity code (MIC) over the MSDU SA and DA, the MSDU priority (see 11.4.2.3), and the MSDU plaintext data. TKIP appends the computed MIC to the MSDU data prior to fragmentation into MPDUs. The receiver verifies the MIC after decryption, ICV checking, and defragmentation of the MPDUs into an MSDU and discards any received MSDUs with invalid MICs. TKIP's MIC provides a defense against forgery attacks.

b) Because of the design constraints of the TKIP MIC, it is still possible for an adversary to compromise message integrity; therefore, TKIP also implements countermeasures. The countermeasures bound the probability of a successful forgery and the amount of information an attacker can learn about a key.

c) TKIP uses a per-MPDU TKIP sequence counter (TSC) to sequence the MPDUs it sends. The receiver drops MPDUs received out of order, i.e., not received with increasing sequence numbers. This provides replay protection. TKIP encodes the TSC value from the sender to the receiver as a WEP IV and extended IV.

d) TKIP uses a cryptographic mixing function to combine a temporal key, the TA, and the TSC into the WEP seed. The receiver recovers the TSC from a received MPDU and utilizes the mixing function to compute the same WEP seed needed to correctly decrypt the MPDU. The key mixing function is designed to defeat weak-key attacks against the WEP key.

TKIP defines additional MIB variables; see Annex C.

11.4.2.1.2 TKIP cryptographic encapsulation

TKIP enhances the WEP cryptographic encapsulation with several additional functions, as depicted in Figure 11-5.

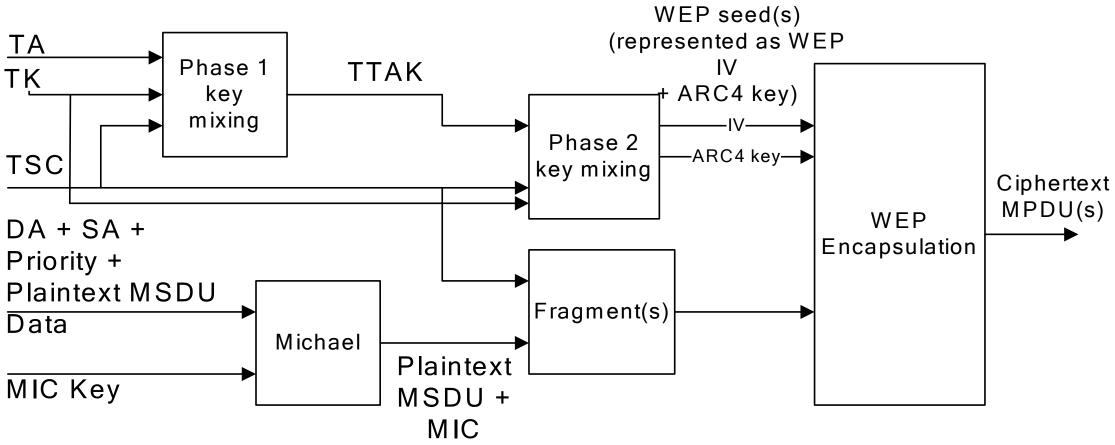


Figure 11-5—TKIP encapsulation block diagram

a) TKIP MIC computation protects the MSDU Data field and corresponding SA, DA, and Priority fields. The computation of the MIC is performed on the ordered concatenation of the SA, DA, Priority, and MSDU Data fields. The MIC is appended to the MSDU Data field. TKIP discards any MIC padding prior to appending the MIC.

b) If needed, IEEE Std 802.11 fragments the MSDU with MIC into one or more MPDUs. TKIP assigns a monotonically increasing TSC value to each MPDU, taking care that all the MPDUs generated from the same MSDU have the same value of extended IV (see 11.4.2.2).

c) For each MPDU, TKIP uses the key mixing function to compute the WEP seed.

d) TKIP represents the WEP seed as a WEP IV and ARC4 key and passes these with each MPDU to WEP for generation of the ICV (see 8.2.4.7), and for encryption of the plaintext MPDU, including all or part of the MIC, if present. WEP uses the WEP seed as a WEP default key, identified by a key identifier associated with the temporal key.

NOTE—When the TSC space is exhausted, the choices available to an implementation are to replace the temporal key with a new one or to end communications. Reuse of any TSC value compromises already sent traffic. Note that retransmitted MPDUs reuse the TSC without any compromise of security. The TSC is large enough, however, that TSC space exhaustion is not expected to be an issue.

In Figure 11-5, the TKIP-mixed transmit address and key (TTAK) denotes the intermediate key produced by Phase 1 of the TKIP mixing function (see 11.4.2.5).

11.4.2.1.3 TKIP decapsulation

TKIP enhances the WEP decapsulation process with the following additional steps:

a) Before WEP decapsulates a received MPDU, TKIP extracts the TSC sequence number and key identifier from the WEP IV and the extended IV. TKIP discards a received MPDU that violates the sequencing rules (see 11.4.2.6) and otherwise uses the mixing function to construct the WEP seed.

b) TKIP represents the WEP seed as a WEP IV and ARC4 key and passes these with the MPDU to WEP for decapsulation.

c) If WEP indicates the ICV check succeeded, the implementation reassembles the MPDU into an MSDU. If the MSDU defragmentation succeeds, the receiver verifies the TKIP MIC. If MSDU defragmentation fails, then the MSDU is discarded.

d) The MIC verification step recomputes the MIC over the MSDU SA, DA, Priority, and MSDU Data fields (but not the TKIP MIC field). The calculated TKIP MIC result is then compared bit-wise to the received MIC.

e) If the received and the locally computed MIC values are identical, the verification succeeds, and TKIP shall deliver the MSDU to the upper layer. If the two differ, then the verification fails; the receiver shall discard the MSDU and shall engage in appropriate countermeasures.

Figure 11-6 depicts this process.

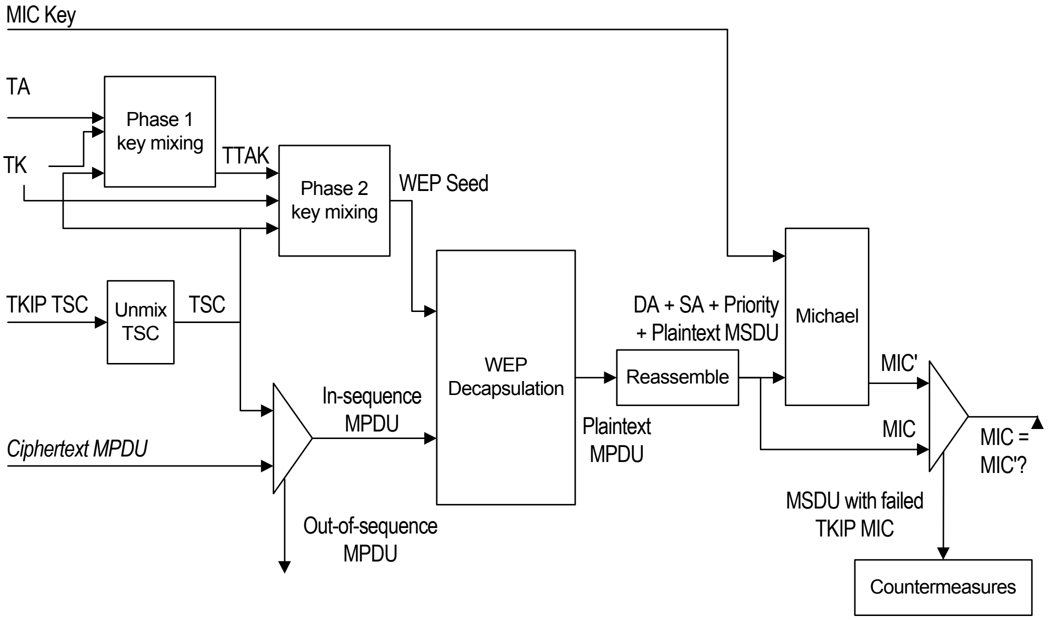


Figure 11-6—TKIP decapsulation block diagram

11.4.2.2 TKIP MPDU formats

TKIP reuses the pre-RSNA WEP MPDU format. It extends the MPDU by 4 octets to accommodate an extension to the WEP IV, denoted by the Extended IV field, and extends the MSDU format by 8 octets to accommodate the new MIC field. TKIP inserts the Extended IV field immediately after the WEP IV field and before the encrypted data. TKIP appends the MIC to the MSDU Data field; the MIC becomes part of the encrypted data.

Once the MIC is appended to the MSDU data, the added MIC octets are considered part of the MSDU for subsequent fragmentation.

Figure 11-7 depicts the layout of the encrypted MPDU when using TKIP. Note that the figure only depicts the case when the MSDU can be encapsulated in a single MPDU.

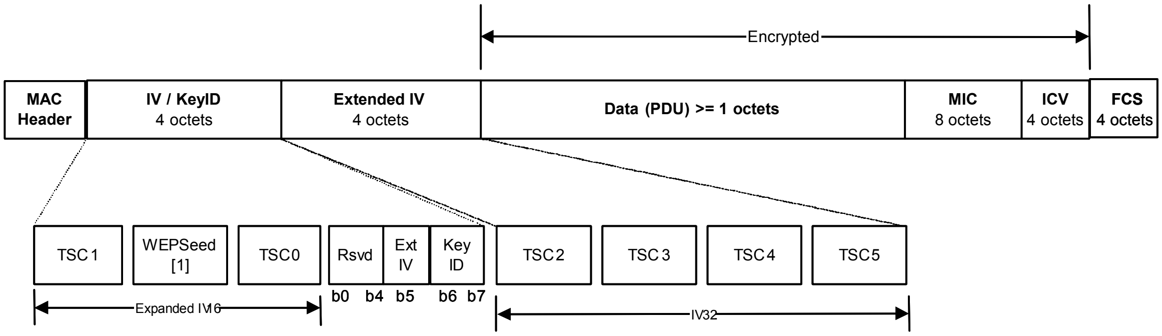


Figure 11-7—Construction of expanded TKIP MPDU

The ExtIV bit in the Key ID octet indicates the presence or absence of an extended IV. If the ExtIV bit is 0, only the nonextended IV is transferred. If the ExtIV bit is 1, an extended IV of 4 octets follows the original IV. For TKIP the ExtIV bit shall be set to 1, and the Extended IV field shall be supplied. The ExtIV bit shall be 0 for WEP frames. The Key ID field shall be set to the key index supplied by the MLME-SETKEYS.request primitive for the key used in cryptographic encapsulation of the frame.

TSC5 is the most significant octet of the TSC, and TSC0 is the least significant. Octets TSC0 and TSC1 form the IV sequence number and are used with the TKIP Phase 2 key mixing. Octets TSC2–TSC5 are used in the TKIP Phase 1 key hashing and are in the Extended IV field. When the lower 16-bit sequence number rolls over (0xFFFF → 0x0000), the extended IV value, i.e., the upper 32 bits of the entire 48-bit TSC, shall be incremented by 1.

NOTE—The rationale for this construction is as follows:

- Aligning on word boundaries eases implementation on legacy devices.
- Adding 4 octets of extended IV eliminates TSC exhaustion as a reason to rekey.
- Key ID octet changes. Bit 5 indicates that an extended IV is present. The receiver/transmitter interprets the 4 octets following the Key ID as the extended IV. The receiving/transmitting STA also uses the value of octets TSC0 and TSC1 to detect that the cached TTAK needs to be updated.

The Extended IV field shall not be encrypted.

WEPSeed[1] is not used to construct the TSC, but is set to (TSC1 | 0x20) & 0x7f.

TKIP shall encrypt all the MPDUs generated from one MSDU under the same temporal key.

11.4.2.3 TKIP MIC

11.4.2.3.1 General

Flaws in the IEEE 802.11 WEP design cause it to fail to meet its goal of protecting data traffic content from casual eavesdroppers. Among the most significant WEP flaws is the lack of a mechanism to defeat message forgeries and other active attacks. To defend against active attacks, TKIP includes a MIC, named Michael. This MIC offers only weak defenses against message forgeries, but it constitutes the best that can be achieved with the majority of legacy hardware. TKIP uses different MIC keys depending on the direction of the transfer as described in 11.7.1 and 11.7.2.

Annex M contains an implementation of the TKIP MIC. It also provides test vectors for the MIC.

11.4.2.3.2 Motivation for the TKIP MIC

Before defining the details of the MIC, it is useful to review the context in which this mechanism operates. Active attacks enabled by the original WEP design include the following:

- Bit-flipping attacks
- Data (payload) truncation, concatenation, and splicing
- Fragmentation attacks
- Iterative guessing attacks against the key
- Redirection by modifying the MPDU DA or RA field
- Impersonation attacks by modifying the MPDU SA or TA field

The MIC makes it more difficult for any of these attacks to succeed.

All of these attacks remain at the MPDU level with the TKIP MIC. The MIC, however, applies to the MSDU, so it blocks successful MPDU-level attacks. TKIP applies the MIC to the MSDU at the transmitter and verifies it at the MSDU level at the receiver. If a MIC check fails at the MSDU level, the implementation shall discard the MSDU and invoke countermeasures (see 11.4.2.4).

Informative Figure 11-8 depicts different peer layers communicating.

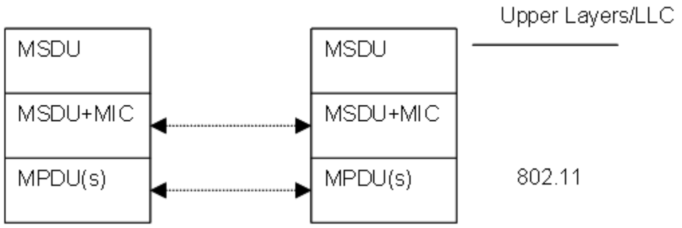


Figure 11-8—TKIP MIC relation to IEEE 802.11 processing (informative)

This figure depicts an architecture where the MIC is logically appended to the raw MSDU in response to the MA-UNITDATA.request primitive. The TKIP MIC is computed over

- The MSDU DA
- The MSDU SA
- The MSDU Priority
- The entire unencrypted MSDU data (payload)

The DA field, SA field, three reserved octets, and a 1-octet Priority field are used only for calculating the MIC. The Priority field refers to the priority parameter of the MA-UNITDATA.request primitive. The fields in Figure 11-9 are treated as an octet stream using the conventions described in 8.2.2.

6	6	1	3	M	1	1	1	1	1	1	1	1	octets
DA	SA	Priority	0	Data	M 0	M 1	M 2	M 3	M 4	M 5	M 6	M 7	

Figure 11-9—TKIP MIC processing format

TKIP appends the MIC at the end of the MSDU payload. The MIC is 8 octets in size for Michael. The IEEE 802.11 MAC then applies its normal processing to transmit this MSDU-with-MIC as a sequence of one or more MPDUs. In other words, the MSDU-with-MIC can be partitioned into one or more MPDUs, the WEP ICV is calculated over each MPDU, and the MIC can even be partitioned to lie in two MPDUs after fragmentation. The TKIP MIC augments, but does not replace, the WEP ICV. Because the TKIP MIC is a weak construction, TKIP protects the MIC with encryption, which makes TKIP MIC forgeries more difficult. The WEP ICV helps to prevent false detection of MIC failures that would cause countermeasures to be invoked.

The receiver reverses this procedure to reassemble the MSDU; and, after the MSDU has been logically reassembled, the IEEE 802.11 MAC verifies the MIC prior to delivery of the MSDU to upper layers. If the MIC validation succeeds, the MAC delivers the MSDU. If the MIC validation fails, the MAC shall discard the MSDU and invoke countermeasures (see 11.4.2.4).

NOTE—TKIP calculates the MIC over the MSDU rather than the MPDU, because doing so increases the implementation flexibility with preexisting WEP hardware.

It should be noted that a MIC alone cannot provide complete forgery protection, as it cannot defend against replay attacks. Therefore, TKIP provides replay detection by TSC sequencing and ICV validation. Furthermore, if TKIP is utilized with a GTK, an insider STA can masquerade as any other STA belonging to the group.

11.4.2.3.3 Definition of the TKIP MIC

Michael generates a 64-bit MIC. The Michael key consists of 64 bits, represented as an 8-octet sequence, $k_0...k_7$. This is converted to two 32-bit words, K_0 and K_1. Throughout this subclause, all conversions between octets and 32-bit words shall use the little endian conventions, given in 8.2.2.

Michael operates on each MSDU including the Priority field, 3 reserved octets, SA field, and DA field. An MSDU consists of octets $m_0...m_{n-1}$ where n is the number of MSDU octets, including SA, DA, Priority, and Data fields. The message is padded at the end with a single octet with value 0x5a, followed by between 4 and 7 zero octets. The number of zero octets is chosen so that the overall length of the padded MSDU is a multiple of four. The padding is not transmitted with the MSDU; it is used to simplify the computation over the final block. The MSDU is then converted to a sequence of 32-bit words $M_0 ...M_{N-1}$, where $N = \lceil (n+5)/4 \rceil$, and where $\lceil a \rceil$ means to round a up to the nearest integer. By construction, $M_{N-1} = 0$ and $M_{N-2} \neq 0$.

The MIC value is computed iteratively starting with the key value (K_0 and K_1) and applying a block function b for every message word, as shown in Figure 11-10. The algorithm loop runs a total of N times (i takes on the values 0 to N–1 inclusive), where N is as above, the number of 32-bit words composing the padded MSDU. The algorithm results in two words (l and r), which are converted to a sequence of 8 octets using the least-significant-octet-first convention:

- M0 = *l* & 0xff
- M1 = (*l*/0x100) & 0xff
- M2 = (*l*/0x10000) & 0xff
- M3 = (*l*/0x1000000) & 0xff
- M4 = *r* & 0xff

- M5 = (*r*/0x100) & 0xff
- M6 = (*r*/0x10000) & 0xff
- M7 = (*r*/0x1000000) & 0xff

This is the MIC value. The MIC value is appended to the MSDU as data to be sent.

```
Input: Key ($K_0$, $K_1$) and padded MSDU (represented as 32-bit words) $M_0 \ldots M_{N-1}$
Output: MIC value ($V_0$, $V_1$)
        MICHAEL(($K_0$, $K_1$) , ($M_0$, ... , $M_N$))
        $(l,r) \leftarrow (K_0, K_1)$
        for $i = 0$ to $N-1$ do
                $l \leftarrow l \oplus M_i$
                $(l,r) \leftarrow b(l,r)$
        return $(l,r)$
```

Figure 11-10—Michael message processing

Figure 11-11 defines the Michael block function *b*. It is a Feistel-type construction with alternating additions and XOR operations. It uses <<< to denote the rotate-left operator on 32-bit values, >>> for the rotate-right operator, and XSWAP for a function that swaps the position of the 2 least significant octets. It also uses the position of the two most significant octets in a word.

```
Input: $(l,r)$
Output (l,r)
b(L,R)
        $r \leftarrow r \oplus (l \lll 17)$
        $l \leftarrow (l + r) \bmod 2^{32}$
        $r \leftarrow r \oplus \mathrm{XSWAP}(l)$
        $l \leftarrow (l + r) \bmod 2^{32}$
        $r \leftarrow r \oplus (l \lll 3)$
        $l \leftarrow (l + r) \bmod 2^{32}$
        $r \leftarrow r \oplus (l \ggg 2)$
        $l \leftarrow (l + r) \bmod 2^{32}$
        return $(l,r)$
```

Figure 11-11—Michael block function

11.4.2.4 TKIP countermeasures procedures

11.4.2.4.1 General

The TKIP MIC trades off security in favor of implementability on pre-RSNA devices. Michael provides only weak protection against active attacks. A failure of the MIC in a received MSDU indicates a probable active attack. A successful attack against the MIC would mean an attacker could inject forged data frames and perform further effective attacks against the encryption key itself. If TKIP implementation detects a probable active attack, TKIP shall take countermeasures as specified in this subclause. These countermeasures accomplish the following goals:

- MIC failure events should be logged as a security-relevant matter. A MIC failure is an almost certain indication of an active attack and warrants a follow-up by the system administrator.
- The rate of MIC failures needs to be kept below two per minute. This implies that STAs and APs detecting two MIC failure events within 60 s need to disable all receptions using TKIP for a period of 60 s. The slowdown makes it difficult for an attacker to make a large number of forgery attempts in a short time.

- As an additional security feature, the PTK and, in the case of the Authenticator, the GTK should be changed.

Before verifying the MIC, the receiver shall check the FCS, ICV, and TSC for all related MPDUs. Any MPDU that has an invalid FCS, an incorrect ICV, or a TSC value that is less than or equal to the TSC replay counter shall be discarded before checking the MIC. This avoids unnecessary MIC failure events. Checking the TSC before the MIC makes countermeasure-based denial-of-service attacks harder to perform. While the FCS and ICV mechanisms are sufficient to detect noise, they are insufficient to detect active attacks. The FCS and ICV provide error detection, but not integrity protection.

A single counter or timer shall be used to log MIC failure events. These failure events are defined as follows:

- For an Authenticator:
  - Detection of a MIC failure on a received individually addressed frame.
  - Receipt of Michael MIC Failure Report frame.
- For a Supplicant:
  - Detection of a MIC failure on a received individually addressed or group addressed frame.
  - Attempt to transmit a Michael MIC Failure Report frame.

The number of MIC failures is accrued independent of the particular key context. Any single MIC failure, whether detected by the Supplicant or the Authenticator and whether resulting from a group MIC key failure or a pairwise MIC key failure, shall be treated as cause for a MIC failure event.

The Supplicant uses a single Michael MIC Failure Report frame to report a MIC failure event to the Authenticator. A Michael MIC Failure Report is an EAPOL-Key frame with the following Key Information field bits set to 1: MIC bit, Error bit, Request bit, Secure bit. The Supplicant protects this message with the current PTK; the Supplicant uses the KCK portion of the PTK to compute the IEEE 802.1X EAPOL MIC.

The MLME-MICHAELMICFAILURE.indication primitive is used by the IEEE 802.11 MAC to attempt to indicate a MIC failure to the local IEEE 802.1X Supplicant or Authenticator. MLME-EAPOL.request primitive is used by the Supplicant to send the EAPOL-Key frame containing the Michael MIC Failure Report. MLME-EAPOL.confirm primitive indicates to the Supplicant when the EAPOL-Key frame has been transmitted.

The first MIC failure shall be logged, and a timer initiated to enable enforcement of the countermeasures. If the MIC failure event is detected by the Supplicant, it shall also report the event to the AP by sending a Michael MIC Failure Report frame.

If a subsequent MIC failure occurs within 60 s of the most recent previous failure, then a STA whose IEEE 802.1X entity has acted as a Supplicant shall deauthenticate (as defined in 10.3.4.4) itself or deauthenticate all the STAs with a security association if its IEEE 802.1X entity acted as an Authenticator. For an IBSS STA, both Supplicant and Authenticator actions shall be taken. Furthermore, the device shall not receive or transmit any TKIP-encrypted data frames, and shall not receive or transmit any unencrypted data frames other than IEEE 802.1X messages, to or from any peer for a period of at least 60 s after it detects the second failure. If the device is an AP, it shall disallow new associations using TKIP during this 60 s period; at the end of the 60 s period, the AP shall resume normal operations and allow STAs to (re)associate. If the device is an IBSS STA, it shall disallow any new security associations using TKIP during this 60 s period. If the device is a Supplicant, it shall first send a Michael MIC Failure Report frame prior to revoking its PTKSA and deauthenticating itself.

The `aMICFailTime` attribute shall contain the `sysUpTime` value at the time the MIC failure was logged.

11.4.2.4.2 TKIP countermeasures for an Authenticator

The countermeasures used by an Authenticator are depicted in Figure 11-12 and described as follows:

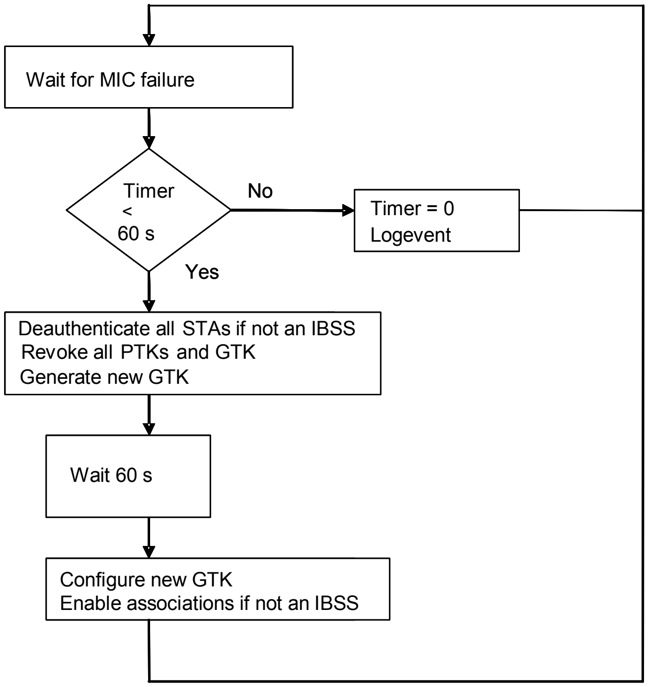


Figure 11-12—Authenticator MIC countermeasures

a) For an Authenticator's STA that receives a frame with a MIC error,
   1) Discard the frame.
   2) Increment the MIC failure counter, dot11RSNAStatsTKIPLocalMICFailures.
   3) Generate a MLME-MICHAELMICFAILURE.indication primitive.
b) For an Authenticator that receives a MLME-MICHAELMICFAILURE.indication primitive or a Michael MIC Failure Report frame,
   1) If it is a Michael MIC Failure Report frame, then increment dot11RSNAStatsTKIP-RemoteMICFailures.
   2) If this is the first MIC failure within the past 60 s, initialize the countermeasures timer.
   3) If less than 60 s have passed since the most recent previous MIC failure, the Authenticator shall deauthenticate and delete all PTKSAs for all STAs using TKIP. If the current GTKSA uses TKIP, that GTKSA shall be discarded, and a new GTKSA constructed, but not used for 60 s. The Authenticator shall refuse the construction of new PTKSAs using TKIP as one or more of the ciphers for 60 s. At the end of this period, the MIC failure counter and timer shall be reset, and creation of PTKSAs accepted as usual.
   4) If the Authenticator is using IEEE 802.1X authentication, the Authenticator shall transition the state of the IEEE 802.1X Authenticator state machine to the INITIALIZE state. This restarts the IEEE 802.1X state machine. If the Authenticator is instead using PSKs, this step is omitted.

Note that a Supplicant's STA may deauthenticate with a reason code of MIC failure if it is an ESS STA. The Authenticator shall not log the deauthenticate as a MIC failure event to prevent denial-of-service attacks through deauthentications. The Supplicant's STA reports the MIC failure event through the Michael MIC Failure Report frame so that the AP can log the event.

The requirement to deauthenticate all STAs using TKIP includes those using other pairwise ciphers if they are using TKIP as the group cipher.

11.4.2.4.3 TKIP countermeasures for a Supplicant

The countermeasures used by a Supplicant are depicted in Figure 11-13 and described as follows:

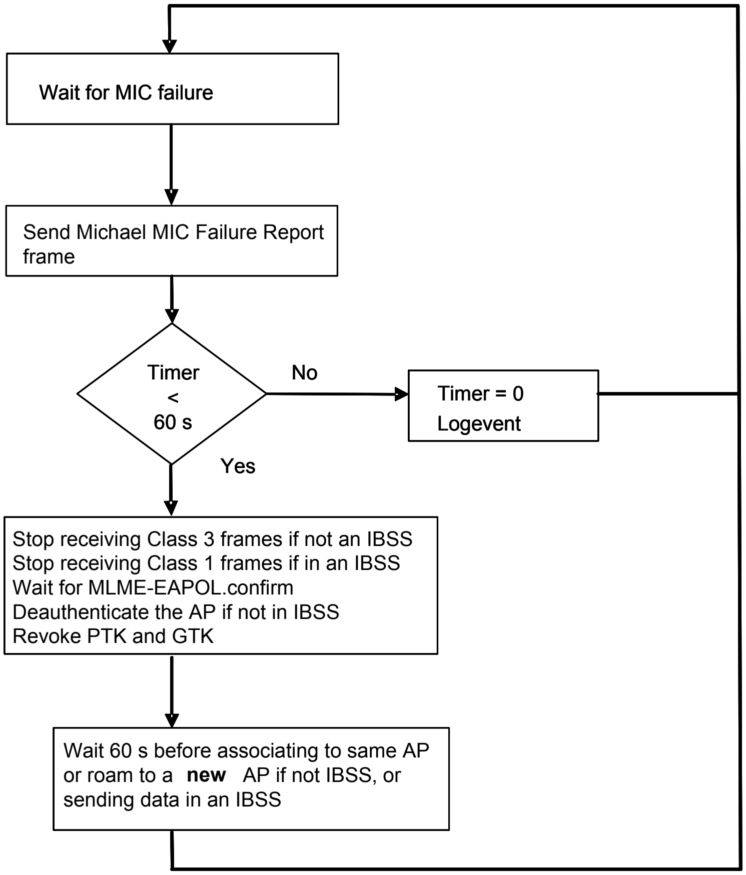

Wait for MIC failure
Send Michael MIC Failure Report frame
Timer < 60 s
No
Yes
Timer = 0
Logevent
Stop receiving Class 3 frames if not an IBSS
Stop receiving Class 1 frames if in an IBSS
Wait for MLME-EAPOL.confirm
Deauthenticate the AP if not in IBSS
Revoke PTK and GTK
Wait 60 s before associating to same AP or roam to a new AP if not IBSS, or sending data in an IBSS

Figure 11-13—Supplicant MIC countermeasures

a) For a Supplicant's STA that receives a frame with a MIC error,
   1) Increment the MIC failure counter, dot11RSNAStatsTKIPLocalMICFailures.
   2) Discard the offending frame.
   3) Generate a MLME-MICHAELMICFAILURE.indication primitive.

b) For a Supplicant that receives an MLME-MICHAELMICFAILURE.indication primitive from its STA,
   1) Send a Michael MIC Failure Report frame to the AP.
   2) If this is the first MIC failure within the past 60 s, initialize the countermeasures timer.
   3) If less than 60 s have passed since the most recent previous MIC failure, delete the PTKSA and GTKSA. Deauthenticate from the AP and wait for 60 s before (re)establishing a TKIP association with the same AP. A TKIP association is any IEEE 802.11 association that uses TKIP for its pairwise or group cipher suite.

c) If a STA receives a deauthenticate frame with the reason code "MIC failure," it is unable to verify that the frame has not been forged, since the frame does not contain a MIC. The STA may attempt association with this, or another, AP. If the frame was genuine, then it is probable that attempts to associate with the same AP requesting the use of TKIP will fail because the AP is conducting countermeasures.

11.4.2.5 TKIP mixing function

11.4.2.5.1 General

Annex M defines a C-language reference implementation of the TKIP mixing function. It also provides test vectors for the mixing function.

The mixing function has two phases. Phase 1 mixes the appropriate temporal key (pairwise or group) with the TA and TSC. A STA may cache the output of this phase to reuse with subsequent MPDUs associated with the same temporal key and TA. Phase 2 mixes the output of Phase 1 with the TSC and temporal key (TK) to produce the WEP seed, also called the *per-frame key*. The WEP seed may be precomputed before it is used. The two-phase process may be summarized as follows:

```
TTAK := Phase1 (TK, TA, TSC)
WEP seed := Phase2 (TTAK, TK, TSC)
```

11.4.2.5.2 S-Box

Both Phase 1 and Phase 2 rely on an S-box, defined in this subclause. The S-box substitutes one 16-bit value with another 16-bit value. This function may be implemented as a table look up.

NOTE—The S-box is a nonlinear substitution. The table look-up is be organized as either a single table with 65 536 entries and a 16-bit index (128K octets of table) or two tables with 256 entries and an 8-bit index (1024 octets for both tables). When the two smaller tables are used, the high-order octet is used to obtain a 16-bit value from one table, the low-order octet is used to obtain a 16-bit value from the other table, and the S-box output is the XOR (⊕) of the two 16-bit values. The second S-box table is an octet-swapped replica of the first.

```
#define _S_(v16)      (Sbox[0][Lo8(v16)] ^ Sbox[1][Hi8(v16)])

/* 2-byte by 2-byte subset of the full AES S-box table */
const u16b Sbox[2][256]=       /* Sbox for hash (can be in ROM)     */
{ {
   0xC6A5,0xF884,0xEE99,0xF68D,0xFF0D,0xD6BD,0xDEB1,0x9154,
   0x6050,0x0203,0xCEA9,0x567D,0xE719,0xB562,0x4DE6,0xEC9A,
   0x8F45,0x1F9D,0x8940,0xFA87,0xEF15,0xB2EB,0x8EC9,0xFB0B,
   0x41EC,0xB367,0x5FFD,0x45EA,0x23BF,0x53F7,0xE496,0x9B5B,
   0x75C2,0xE11C,0x3DAE,0x4C6A,0x6C5A,0x7E41,0xF502,0x834F,
   0x685C,0x51F4,0xD134,0xF908,0xE293,0xAB73,0x6253,0x2A3F,
   0x080C,0x9552,0x4665,0x9D5E,0x3028,0x37A1,0x0A0F,0x2FB5,
   0x0E09,0x2436,0x1B9B,0xDF3D,0xCD26,0x4E69,0x7FCD,0xEA9F,
   0x121B,0x1D9E,0x5874,0x342E,0x362D,0xDCB2,0xB4EE,0x5BFB,
   0xA4F6,0x764D,0xB761,0x7DCE,0x527B,0xDD3E,0x5E71,0x1397,
   0xA6F5,0xB968,0x0000,0xC12C,0x4060,0xE31F,0x79C8,0xB6ED,
   0xD4BE,0x8D46,0x67D9,0x724B,0x94DE,0x98D4,0xB0E8,0x854A,
   0xBB6B,0xC52A,0x4FE5,0xED16,0x86C5,0x9AD7,0x6655,0x1194,
   0x8ACF,0xE910,0x0406,0xFE81,0xA0F0,0x7844,0x25BA,0x4BE3,
   0xA2F3,0x5DFE,0x80C0,0x058A,0x3FAD,0x21BC,0x7048,0xF104,
   0x63DF,0x77C1,0xAF75,0x4263,0x2030,0xE51A,0xFD0E,0xBF6D,
   0x814C,0x1814,0x2635,0xC32F,0xBEE1,0x35A2,0x88CC,0x2E39,
   0x9357,0x55F2,0xFC82,0x7A47,0xC8AC,0xBAE7,0x322B,0xE695,
   0xC0A0,0x1998,0x9ED1,0xA37F,0x4466,0x547E,0x3BAB,0x0B83,
   0x8CCA,0xC729,0x6BD3,0x283C,0xA779,0xBCE2,0x161D,0xAD76,
   0xDB3B,0x6456,0x744E,0x141E,0x92DB,0x0C0A,0x486C,0xB8E4,
   0x9F5D,0xBD6E,0x43EF,0xC4A6,0x39A8,0x31A4,0xD337,0xF28B,
   0xD532,0x8B43,0x6E59,0xDAB7,0x018C,0xB164,0x9CD2,0x49E0,
   0xD8B4,0xACFA,0xF307,0xCF25,0xCAAF,0xF48E,0x47E9,0x1018,
   0x6FD5,0xF088,0x4A6F,0x5C72,0x3824,0x57F1,0x73C7,0x9751,
   0xCB23,0xA17C,0xE89C,0x3E21,0x96DD,0x61DC,0x0D86,0x0F85,
   0xE090,0x7C42,0x71C4,0xCCAA,0x90D8,0x0605,0xF701,0x1C12,
   0xC2A3,0x6A5F,0xAEF9,0x69D0,0x1791,0x9958,0x3A27,0x27B9,
   0xD938,0xEB13,0x2BB3,0x2233,0xD2BB,0xA970,0x0789,0x33A7,
   0x2DB6,0x3C22,0x1592,0xC920,0x8749,0xAAFF,0x5078,0xA57A,
   0x038F,0x59F8,0x0980,0x1A17,0x65DA,0xD731,0x84C6,0xD0B8,
   0x82C3,0x29B0,0x5A77,0x1E11,0x7BCB,0xA8FC,0x6DD6,0x2C3A,
  },
  {  /* second half of table is byte-reversed version of first! */
   0xA5C6,0x84F8,0x99EE,0x8DF6,0x0DFF,0xBDD6,0xB1DE,0x5491,
   0x5060,0x0302,0xA9CE,0x7D56,0x19E7,0x62B5,0xE64D,0x9AEC,
   0x458F,0x9D1F,0x4089,0x87FA,0x15EF,0xEBB2,0xC98E,0x0BFB,
   0xEC41,0x67B3,0xFD5F,0xEA45,0xBF23,0xF753,0x96E4,0x5B9B,
   0xC275,0x1CE1,0xAE3D,0x6A4C,0x5A6C,0x417E,0x02F5,0x4F83,
   0x5C68,0xF451,0x34D1,0x08F9,0x93E2,0x73AB,0x5362,0x3F2A,
   0x0C08,0x5295,0x6546,0x5E9D,0x2830,0xA137,0x0F0A,0xB52F,
   0x090E,0x3624,0x9B1B,0x3DDF,0x26CD,0x694E,0xCD7F,0x9FEA,
   0x1B12,0x9E1D,0x7458,0x2E34,0x2D36,0xB2DC,0xEEB4,0xFB5B,
   0xF6A4,0x4D76,0x61B7,0xCE7D,0x7B52,0x3EDD,0x715E,0x9713,
   0xF5A6,0x68B9,0x0000,0x2CC1,0x6040,0x1FE3,0xC879,0xEDB6,
   0xBED4,0x468D,0xD967,0x4B72,0xDE94,0xD498,0xE8B0,0x4A85,
   0x6BBB,0x2AC5,0xE54F,0x16ED,0xC586,0xD79A,0x5566,0x9411,
   0xCF8A,0x10E9,0x0604,0x81FE,0xF0A0,0x4478,0xBA25,0xE34B,
   0xF3A2,0xFE5D,0xC080,0x8A05,0xAD3F,0xBC21,0x4870,0x04F1,
   0xDF63,0xC177,0x75AF,0x6342,0x3020,0x1AE5,0x0EFD,0x6DBF,
   0x4C81,0x1418,0x3526,0x2FC3,0xE1BE,0xA235,0xCC88,0x392E,
   0x5793,0xF255,0x82FC,0x477A,0xACC8,0xE7BA,0x2B32,0x95E6,
   0xA0C0,0x9819,0xD19E,0x7FA3,0x6644,0x7E54,0xAB3B,0x830B,
   0xCA8C,0x29C7,0xD36B,0x3C28,0x79A7,0xE2BC,0x1D16,0x76AD,
   0x3BDB,0x5664,0x4E74,0x1E14,0xDB92,0x0A0C,0x6C48,0xE4B8,
   0x5D9F,0x6EBD,0xEF43,0xA6C4,0xA839,0xA431,0x37D3,0x8BF2,
   0x32D5,0x438B,0x596E,0xB7DA,0x8C01,0x64B1,0xD29C,0xE049,
   0xB4D8,0xFAAC,0x07F3,0x25CF,0xAFCA,0x8EF4,0xE947,0x1810,
   0xD56F,0x88F0,0x6F4A,0x725C,0x2438,0xF157,0xC773,0x5197,
   0x23CB,0x7CA1,0x9CE8,0x213E,0xDD96,0xDC61,0x860D,0x850F,
   0x90E0,0x427C,0xC471,0xAACC,0xD890,0x0506,0x01F7,0x121C,
   0xA3C2,0x5F6A,0xF9AE,0xD069,0x9117,0x5899,0x273A,0xB927,
   0x38D9,0x13EB,0xB32B,0x3322,0xBBD2,0x70A9,0x8907,0xA733,
   0xB62D,0x223C,0x9215,0x20C9,0x4987,0xFFAA,0x7850,0x7AA5,
   0x8F03,0xF859,0x8009,0x171A,0xDA65,0x31D7,0xC684,0xB8D0,
   0xC382,0xB029,0x775A,0x111E,0xCB7B,0xFCA8,0xD66D,0x3A2C,
  }
};
```

11.4.2.5.3 Phase 1 Definition

The inputs to Phase 1 of the temporal key mixing function shall be a temporal key (*TK*), the TA, and the TSC. The temporal key shall be 128 bits in length. Only the 32 MSBs of the TSC and all of the temporal key are used in Phase 1. The output, *TTAK*, shall be 80 bits in length and is represented by an array of 16-bit values: $TTAK_0$ $TTAK_1$ $TTAK_2$ $TTAK_3$ $TTAK_4$.

The description of the Phase 1 algorithm treats all of the following values as arrays of 8-bit values: $TA_0..TA_5$, $TK_0..TK_{15}$. The *TA* octet order is represented according to the conventions from 8.2.2, and the first 3 octets represent the OUI.

The XOR (⊕) operation, the bit-wise-and (&) operation, and the addition (+) operation are used in the Phase 1 specification. A loop counter, *i*, and an array index temporary variable, *j*, are also employed.

One function, *Mk*16, is used in the definition of Phase 1. The function *Mk*16 constructs a 16-bit value from two 8-bit inputs as $Mk16(X,Y) = (256 \cdot X)+Y$.

Two steps make up the Phase 1 algorithm. The first step initializes *TTAK* from *TSC* and *TA*. The second step uses an S-box to iteratively mix the keying material into the 80-bit *TTAK*. The second step sets the PHASE1_LOOP_COUNT to 8. See Figure 11-14.

```
Input: transmit address $TA0 \ldots TA5$, Temporal Key $TK0..TK15$, and $TSC0..TSC5$
Output: intermediate key $TTAK0..TTAK4$
PHASE1-KEY-MIXING($TA0 \ldots TA5$, $TK0..TK15$, $TSC0..TSC5$)
    PHASE1_STEP1:
    $TTAK0 \leftarrow MK16(TSC3, TSC2)$
    $TTAK1 \leftarrow MK16(TSC5, TSC4)$
    $TTAK2 \leftarrow Mk16(TA1,TA0)$
    $TTAK3 \leftarrow Mk16(TA3,TA2)$
    $TTAK4 \leftarrow Mk16(TA5,TA4)$
    PHASE1_STEP2:
    for $i$ = 0 to PHASE1_LOOP_COUNT-1
        $j \leftarrow 2 \cdot (i \,\&\, 1)$
        $TTAK0 \leftarrow TTAK0 + S[TTAK4 \oplus Mk16(TK1+j,TK0+j)]$
        $TTAK1 \leftarrow TTAK1 + S[TTAK0 \oplus Mk16(TK5+j,TK4+j)]$
        $TTAK2 \leftarrow TTAK2 + S[TTAK1 \oplus Mk16(TK9+j,TK8+j)]$
        $TTAK3 \leftarrow TTAK3 + S[TTAK2 \oplus Mk16(TK13+j,TK12+j)]$
        $TTAK4 \leftarrow TTAK4 + S[TTAK3 \oplus Mk16(TK1+j,TK0+j)] + i$
```

Figure 11-14—Phase 1 key mixing

NOTE 1—The TA is mixed into the temporal key in Phase 1 of the hash function. Implementations might achieve a significant performance improvement by caching the output of Phase 1. The Phase 1 output is the same for 2^{16} = 65 536 consecutive frames from the same temporal key and TA. Consider the simple case where a STA communicates only with an AP. The STA performs Phase 1 using its own address, and the *TTAK* is used to protect traffic sent to the AP. The STA performs Phase 1 using the AP address, and it is used to unwrap traffic received from the AP.

NOTE 2—The cached *TTAK* from Phase 1 needs to be updated when the lower 16 bits of the TSC wrap and the upper 32 bits need to be updated.

11.4.2.5.4 Phase 2 definition

The inputs to Phase 2 of the temporal key mixing function shall be the output of Phase 1 (*TTAK*) together with the temporal key and the TSC. The *TTAK* is 80-bits in length. Only the 16 LSBs of the TSC are used in Phase 2. The temporal key is 128 bits. The output is the WEP seed, which is a per-frame key, and is 128 bits in length. The constructed WEP seed has an internal structure conforming to the WEP specification. In other words, the first 24 bits of the WEP seed shall be transmitted in plaintext as the WEP IV. As such, these 24 bits are used to convey lower 16 bits of the TSC from the sender (encryptor) to the receiver (decryptor). The rest of the TSC shall be conveyed in the Extended IV field. The temporal key and *TTAK* values are represented as in Phase 1. The WEP seed is treated as an array of 8-bit values: $WEPSeed_0 \ldots WEPSeed_{15}$. The TSC shall be treated as an array of 8-bit values: TSC_0 TSC_1 TSC_2 TSC_3 TSC_4 TSC_5.

The pseudo-code specifying the Phase 2 mixing function employs one variable: *PPK*, which is 96 bits long. The *PPK* is represented as an array of 16-bit values: $PPK_0..PPK_5$. The pseudo-code also employs a loop counter, *i*. As detailed in this subclause, the mapping from the 16-bit *PPK* values to the 8-bit *WEPseed* values is explicitly little endian to match the endian architecture of the most common processors used for this application.

The XOR (⊕) operation, the addition (+) operation, the AND (&) operation, the OR (|) operation, and the right bit shift (>>) operation are used in the specification of Phase 2. See Figure 11-15.

```
Input: intermediate key $TTAK0 \ldots TTAK4$, $TK$, and TKIP sequence counter $TSC$
Output: WEP Seed $WEPSeed0 \ldots WEPSeed15$
PHASE2-KEY-MIXING($TTAK0 \ldots TTAK4$, $TK0 \ldots TK15$, $TSC0 \ldots TSC5$)
    PHASE2_STEP1:
        $PPK0 \leftarrow TTAK0$
        $PPK1 \leftarrow TTAK1$
        $PPK2 \leftarrow TTAK2$
        $PPK3 \leftarrow TTAK3$
        $PPK4 \leftarrow TTAK4$
        $PPK5 \leftarrow TTAK4 + \mathrm{Mk16}(\mathrm{TSC1}, \mathrm{TSC0})$
    PHASE2_STEP2:
        $PPK0 \leftarrow PPK0 + S[PPK5 \oplus Mk16(TK1,TK0)]$
        $PPK1 \leftarrow PPK1 + S[PPK0 \oplus Mk16(TK3,TK2)]$
        $PPK2 \leftarrow PPK2 + S[PPK1 \oplus Mk16(TK5,TK4)]$
        $PPK3 \leftarrow PPK3 + S[PPK2 \oplus Mk16(TK7,TK6)]$
        $PPK4 \leftarrow PPK4 + S[PPK3 \oplus Mk16(TK9,TK8)]$
        $PPK5 \leftarrow PPK5 + S[PPK4 \oplus Mk16(TK11,TK10)]$
        $PPK0 \leftarrow PPK0 + RotR1(PPK5 \oplus Mk16(TK13,TK12))$
        $PPK1 \leftarrow PPK1 + RotR1(PPK0 \oplus Mk16(TK15,TK14))$
        $PPK2 \leftarrow PPK2 + RotR1(PPK1)$
        $PPK3 \leftarrow PPK3 + RotR1(PPK2)$
        $PPK4 \leftarrow PPK4 + RotR1(PPK3)$
        $PPK5 \leftarrow PPK5 + RotR1(PPK4)$
    PHASE2_STEP3:
        $WEPSeed0 \leftarrow TSC1$
        $WEPSeed1 \leftarrow (\mathrm{TSC1} \mid \mathrm{0x20})\ \&\ \mathrm{0x7F}$
        $WEPSeed2 \leftarrow \mathrm{TSC0}$
        $WEPSeed3 \leftarrow Lo8((PPK5 \oplus Mk16(TK1,TK0)) >> 1)$
        for $i = 0$ to 5
            $WEPSeed4+(2 \cdot i) \leftarrow Lo8(PPKi)$
            $WEPSeed5+(2 \cdot i) \leftarrow Hi8(PPKi)$
        end
    return $WEPSeed0 \ldots WEPSeed15$
```

Figure 11-15—Phase 2 key mixing

The algorithm specification relies on four functions:

- The first function, *Lo*8, references the 8 LSBs of the 16-bit input value.
- The second function, *Hi*8, references the 8 MSBs of the 16-bit value.
- The third function, *RotR*1, rotates its 16-bit argument 1 bit to the right.
- The fourth function, *Mk*16, is already used in Phase 1, defined by $Mk16(X,Y) = (256 \cdot X)+Y$, and constructs a 16-bit output from two 8-bit inputs.

NOTE—The rotate and addition operations in STEP2 make Phase 2 particularly sensitive to the endian architecture of the processor, although the performance degradation due to running this algorithm on a big endian processor is expected to be minor.

Phase 2 comprises three steps:

- STEP1 makes a copy of *TTAK* and brings in the TSC.
- STEP2 is a 96-bit bijective mixing, employing an S-box.
- STEP3 brings in the last of the temporal key *TK* bits and assigns the 24-bit WEP IV value.

The WEP IV format carries 3 octets. STEP3 of Phase 2 determines the value of each of these three octets. The construction was selected to preclude the use of known ARC4 weak keys. The recipient can reconstruct the 16 LSBs of the TSC used by the originator by concatenating the third and first octets, ignoring the second octet. The remaining 32 bits of the TSC are obtained from the Extended IV field.

11.4.2.6 TKIP replay protection procedures

TKIP implementations shall use the TSC field to defend against replay attacks by implementing the following rules:

a) Each MPDU shall have a unique TKIP TSC value.

b) Each transmitter shall maintain a single TSC (48-bit counter) for each PTKSA, GTKSA, and STKSA.

c) The TSC shall be implemented as a 48-bit monotonically incrementing counter, initialized to 1 when the corresponding TKIP temporal key is initialized or refreshed.

d) The WEP IV format carries the 16 LSBs of the 48-bit TSC, as defined by the TKIP mixing function (Phase 2, STEP3). The remainder of the TSC is carried in the Extended IV field.

e) A receiver shall maintain a separate set of TKIP TSC replay counters for each PTKSA, GTKSA, and STKSA.

f) TKIP replay detection takes place after the MIC verification and any reordering required by ACK processing. Thus, a receiver shall delay advancing a TKIP TSC replay counter until an MSDU passes the MIC check, to prevent attackers from injecting MPDUs with valid ICVs and TSCs, but invalid MICs.

   NOTE—This works because if an attacker modifies the TSC, then the encryption key is modified and hence both the ICV and MIC decrypt incorrectly, causing the received MPDU to be dropped.

g) For each PTKSA, GTKSA, and STKSA, the receiver shall maintain a separate replay counter for each frame priority and shall use the TSC recovered from a received frame to detect replayed frames, subject to the limitations on the number of supported replay counters indicated in the RSN Capabilities field, as described in 8.4.2.27. A replayed frame occurs when the TSC extracted from a received frame is less than or equal to the current replay counter value for the frame’s priority. A transmitter shall not reorder frames with different priorities without ensuring that the receiver supports the required number of replay counters. The transmitter shall not reorder frames within a replay counter, but may reorder frames across replay counters. One possible reason for reordering frames is the IEEE 802.11 MSDU priority.

h) A receiver shall discard any MPDU that is received out of order and shall increment the value of dot11RSNAStatsTKIPReplays for this key.

i) For MSDUs sent using the Block Ack feature, reordering of received MSDUs according to the Block Ack receiver operation (described in 9.21.4) is performed prior to replay detection.

11.4.3 CTR with CBC-MAC Protocol (CCMP)

11.4.3.1 General

Subclause 11.4.3 specifies the CCMP, which provides data confidentiality, authentication, integrity, and replay protection. CCMP is mandatory for RSN compliance.

CCMP is based on the CCM of the AES encryption algorithm. CCM combines CTR for data confidentiality and CBC-MAC for authentication and integrity. CCM protects the integrity of both the MPDU Data field and selected portions of the IEEE 802.11 MPDU header.

The AES algorithm is defined in FIPS PUB 197-2001. All AES processing used within CCMP uses AES with a 128-bit key and a 128-bit block size.

CCM is defined in IETF RFC 3610. CCM is a generic mode that can be used with any block-oriented encryption algorithm. CCM has two parameters (*M* and *L*), and CCMP uses the following values for the CCM parameters:

— *M* = 8; indicating that the MIC is 8 octets.

— *L* = 2; indicating that the Length field is 2 octets, which is sufficient to hold the length of the largest possible IEEE 802.11 MPDU, expressed in octets.

CCM requires a fresh temporal key for every session. CCM also requires a unique nonce value for each frame protected by a given temporal key, and CCMP uses a 48-bit packet number (PN) for this purpose. Reuse of a PN with the same temporal key voids all security guarantees.

Annex M provides a test vector for CCM.

When CCMP is selected as the RSN pairwise cipher and management frame protection is negotiated, individually addressed robust management frames and the group addressed management frames that receive "Group Addressed Privacy" as indicated in Table 8-38 shall be protected with CCMP.

11.4.3.2 CCMP MPDU format

Figure 11-16 depicts the MPDU when using CCMP.

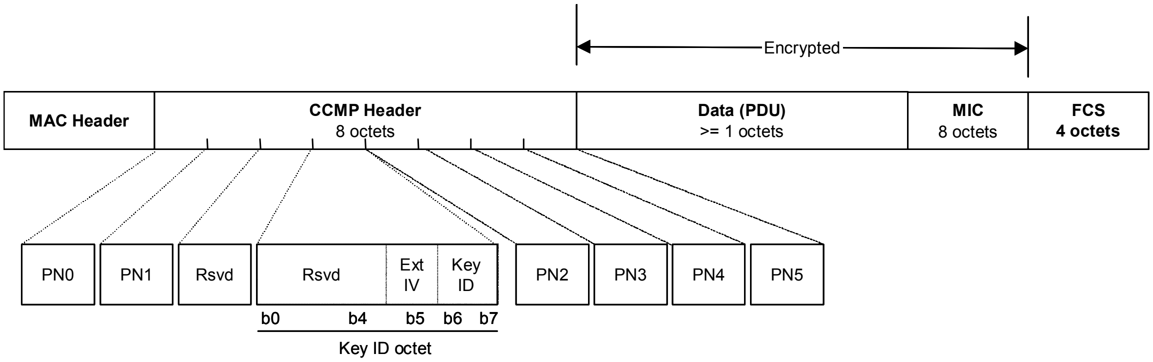


Figure 11-16—Expanded CCMP MPDU

CCMP processing expands the original MPDU size by 16 octets, 8 octets for the CCMP Header field and 8 octets for the MIC field. The CCMP Header field is constructed from the PN, ExtIV, and Key ID subfields. PN is a 48-bit PN represented as an array of 6 octets. PN5 is the most significant octet of the PN, and PN0 is the least significant. Note that CCMP does not use the WEP ICV.

The ExtIV subfield (bit 5) of the Key ID octet signals that the CCMP Header field extends the MPDU header by a total of 8 octets, compared to the 4 octets added to the MPDU header when WEP is used. The ExtIV bit (bit 5) is always set to 1 for CCMP.

Bits 6–7 of the Key ID octet are for the Key ID subfield.

The reserved bits shall be set to 0 and shall be ignored on reception.

11.4.3.3 CCMP cryptographic encapsulation

11.4.3.3.1 General

The CCMP cryptographic encapsulation process is depicted in Figure 11-17.

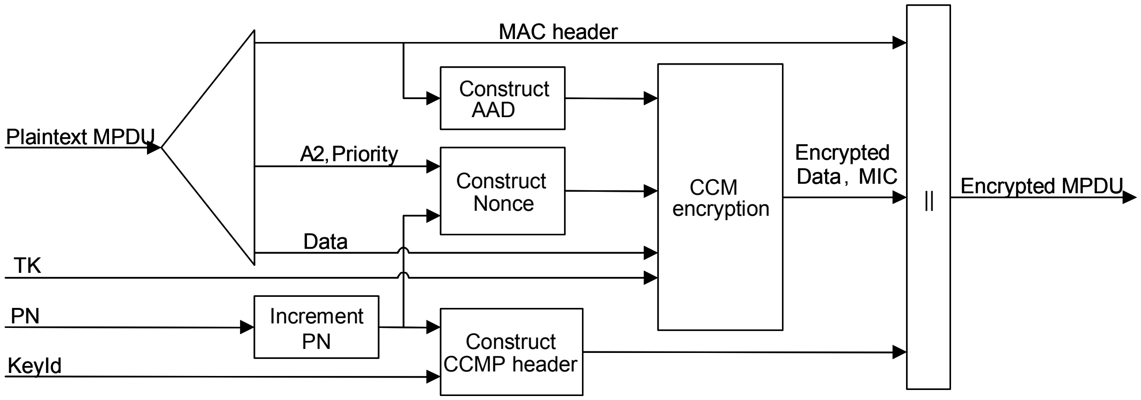


Figure 11-17—CCMP encapsulation block diagram

CCMP encrypts the payload of a plaintext MPDU and encapsulates the resulting cipher text using the following steps:

a) Increment the PN, to obtain a fresh PN for each MPDU, so that the PN never repeats for the same temporal key. Note that retransmitted MPDUs are not modified on retransmission.

b) Use the fields in the MPDU header to construct the additional authentication data (AAD) for CCM. The CCM algorithm provides integrity protection for the fields included in the AAD. MPDU header fields that may change when retransmitted are muted by being masked to 0 when calculating the AAD.

c) Construct the CCM Nonce block from the PN, A2, and the Priority field of the MPDU where A2 is MPDU Address 2.

d) Place the new PN and the key identifier into the 8-octet CCMP header.

e) Use the temporal key, AAD, nonce, and MPDU data to form the cipher text and MIC. This step is known as CCM originator processing.

f) Form the encrypted MPDU by combining the original MPDU header, the CCMP header, the encrypted data and MIC, as described in 11.4.3.2.

The CCM reference describes the processing of the key, nonce, AAD, and data to produce the encrypted output. See 11.4.3.3.2 to 11.4.3.3.6 for details of the creation of the AAD and nonce from the MPDU and the associated MPDU-specific processing.

11.4.3.3.2 PN processing

The PN is incremented by a positive number for each MPDU. The PN shall never repeat for a series of encrypted MPDUs using the same temporal key.

11.4.3.3.3 Construct AAD

The format of the AAD is shown in Figure 11-18.

	FC	A1	A2	A3	SC	A4	QC
Octets:	2	6	6	6	2	6	2

Figure 11-18—AAD construction

The length of the AAD varies depending on the presence or absence of the QC and A4 fields and is shown in Table 11-1.

Table 11-1—AAD length

QC field	A4 field	AAD length (octets)
Absent	Absent	22
Present	Absent	24
Absent	Present	28
Present	Present	30

The AAD is constructed from the MPDU header. The AAD does not include the header Duration field, because the Duration field value might change due to normal IEEE 802.11 operation (e.g., a rate change during retransmission). The AAD includes neither the Duration/ID field nor the HT Control field because the contents of these fields might change during normal operation (e.g., due to a rate change preceding retransmission). The HT Control field might also be inserted or removed during normal operation (e.g., retransmission of an A-MPDU where the original A-MPDU included an MRQ that has already generated a response). For similar reasons, several subfields in the Frame Control field are masked to 0. AAD construction is performed as follows:

a) FC – MPDU Frame Control field, with
   1) Subtype bits (bits 4 5 6) in a Data MPDU masked to 0
   2) Retry bit (bit 11) masked to 0
   3) Power Management bit (bit 12) masked to 0
   4) More Data bit (bit 13) masked to 0
   5) Protected Frame bit (bit 14) always set to 1
   6) Order bit (bit 15) as follows:
      i) Masked to 0 in all data MPDUs containing a QoS Control field
      ii) Unmasked otherwise

b) A1 – MPDU Address 1 field.

c) A2 – MPDU Address 2 field.

d) A3 – MPDU Address 3 field.

e) SC – MPDU Sequence Control field, with the Sequence Number subfield (bits 4–15 of the Sequence Control field) masked to 0. The Fragment Number subfield is not modified.

f) A4 – MPDU Address field, if present.

g) QC – QoS Control field, if present, a 2-octet field that includes the MSDU priority. The QC TID is used in the construction of the AAD. When both the STA and its peer have their SPP A-MSDU Capable fields equal to 1, bit 7 (the A-MSDU Present field) is used in the construction of the AAD. The remaining QC fields are masked to 0 for the AAD calculation (bits 4 to 6, bits 8 to 15, and bit 7 when either the STA or its peer has the SPP A-MSDU Capable field equal to 0).

11.4.3.3.4 Construct CCM nonce

The Nonce field occupies 13 octets, and its structure is shown in Figure 11-19. The structure of the Nonce Flags subfield of the Nonce field is shown in Figure 11-20.

	Nonce Flags	A2	PN
Octets:	1	6	6

Figure 11-19—Nonce construction

	B0 B3	B4	B5 B7
	Priority	Management	Reserved
Bits:	4	1	3

Figure 11-20—Nonce Flags subfield

The Nonce field has an internal structure of Nonce Flags || A2 || PN ("||" is concatenation), where

- The Priority subfield of the Nonce Flags field shall be set to the fixed value 0 when there is no QC field present in the MPDU header. When the QC field is present, bits 0 to 3 of the Priority subfield shall be set to the value of the QC TID (bits 0 to 3 of the QC field).
- When management frame protection is negotiated, the Management field of the Nonce Flags field shall be set to 1 if the Type field of the Frame Control field is 00 (Management frame); otherwise it is set to 0.
- Bits 5 to 7 of the Nonce Flags field are reserved and shall be set to 0 on transmission.
- MPDU address A2 field occupies octets 1–6. This shall be encoded with the octets ordered with A2 octet 0 at octet index 1 and A2 octet 5 at octet index 6.
- The PN field occupies octets 7–12. The octets of PN shall be ordered so that PN0 is at octet index 12 and PN5 is at octet index 7.

11.4.3.3.5 Construct CCMP header

The format of the 8-octet CCMP header is given in 11.4.3.2. The header encodes the PN, Key ID, and ExtIV field values used to encrypt the MPDU.

11.4.3.3.6 CCM originator processing

CCM is a generic authenticate-and-encrypt block cipher mode, and in this standard, CCM is used with the AES block cipher.

There are four inputs to CCM originator processing:

a) *Key:* the temporal key (16 octets).
b) *Nonce:* the nonce (13 octets) constructed as described in 11.4.3.3.4.
c) *Frame body:* the frame body of the MPDU.
d) *AAD:* the AAD (22–30 octets) constructed from the MPDU header as described in 11.4.3.3.3.

The CCM originator processing provides authentication and integrity of the frame body and the AAD as well as data confidentiality of the frame body. The output from the CCM originator processing consists of the encrypted data and 8 additional octets of encrypted MIC (see Figure 11-16).

A CCMP protected individually addressed robust management frame shall be protected with the TK.

11.4.3.4 CCMP decapsulation

11.4.3.4.1 General

Figure 11-21 depicts the CCMP decapsulation process.

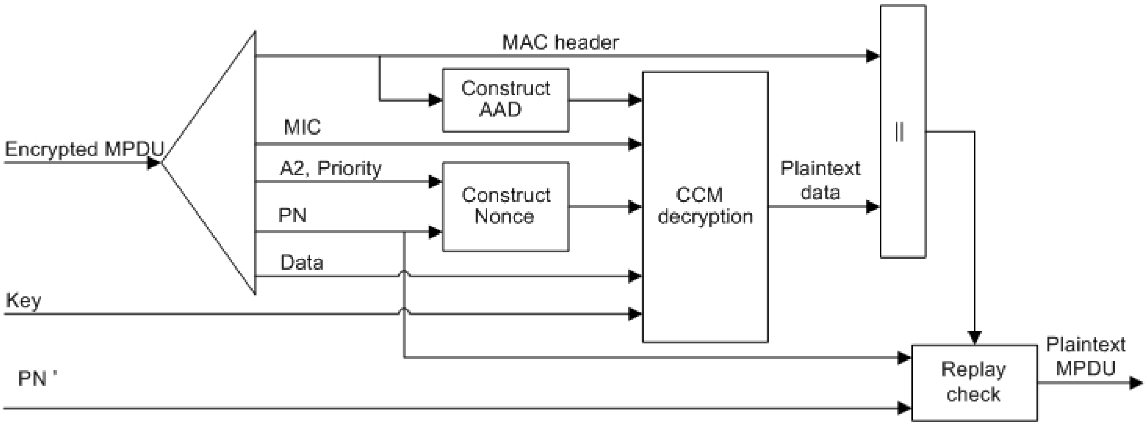


Figure 11-21—CCMP decapsulation block diagram

CCMP decrypts the payload of a cipher text MPDU and decapsulates a plaintext MPDU using the following steps:

a) The encrypted MPDU is parsed to construct the AAD and nonce values.
b) The AAD is formed from the MPDU header of the encrypted MPDU.
c) The Nonce value is constructed from the A2, PN, and Nonce Flags fields.
d) The MIC is extracted for use in the CCM integrity checking.
e) The CCM recipient processing uses the temporal key, AAD, nonce, MIC, and MPDU cipher text data to recover the MPDU plaintext data as well as to check the integrity of the AAD and MPDU plaintext data.
f) The received MPDU header and the MPDU plaintext data from the CCM recipient processing are concatenated to form a plaintext MPDU.
g) The decryption processing prevents replay of MPDUs by validating that the PN in the MPDU is greater than the replay counter maintained for the session.

See 11.4.3.4.2 to 11.4.3.4.4 for details of this processing.

When the received frame is a CCMP protected individually addressed robust management frame, contents of the MMPDU body after protection is removed shall be delivered to the SME via the MLME primitive designated for that management frame rather than through the MA-UNITDATA.indication primitive.

11.4.3.4.2 CCM recipient processing

CCM recipient processing uses the same parameters as CCM originator processing. A CCMP protected individually addressed robust management frame shall use the same TK as a Data MPDU.

There are four inputs to CCM recipient processing:

— *Key:* the temporal key (16 octets).

— *Nonce:* the nonce (13 octets) constructed as described in 11.4.3.3.4.

— *Encrypted frame body:* the encrypted frame body from the received MPDU. The encrypted frame body includes an 8-octet MIC.

— *AAD:* the AAD (22–30 octets) that is the canonical MPDU header as described in 11.4.3.3.3.

The CCM recipient processing checks the authentication and integrity of the frame body and the AAD as well as decrypting the frame body. The plaintext is returned only if the MIC check is successful.

There is one output from error-free CCM recipient processing:

— *Frame body:* the plaintext frame body, which is 8 octets smaller than the encrypted frame body.

11.4.3.4.3 Decrypted CCMP MPDU

The decapsulation process succeeds when the calculated MIC matches the MIC value obtained from decrypting the received encrypted MPDU. The original MPDU header is concatenated with the plaintext data resulting from the successful CCM recipient processing to create the plaintext MPDU.

11.4.3.4.4 PN and replay detection

To effect replay detection, the receiver extracts the PN from the CCMP header. See 11.4.3.2 for a description of how the PN is encoded in the CCMP header. The following processing rules are used to detect replay:

a) The PN values sequentially number each MPDU.

b) Each transmitter shall maintain a single PN (48-bit counter) for each PTKSA, GTKSA, and STKSA.

c) The PN shall be implemented as a 48-bit monotonically incrementing non-negative integer, initialized to 1 when the corresponding temporal key is initialized or refreshed.

d) A receiver shall maintain a separate set of PN replay counters for each PTKSA, GTKSA, and STKSA. The receiver initializes these replay counters to 0 when it resets the temporal key for a peer. The replay counter is set to the PN value of accepted CCMP MPDUs.

e) For each PTKSA, GTKSA, and STKSA, the recipient shall maintain a separate replay counter for each IEEE 802.11 MSDU or A-MSDU priority and shall use the PN recovered from a received frame to detect replayed frames, subject to the limitation of the number of supported replay counters indicated in the RSN Capabilities field (see 8.4.2.27). A replayed frame occurs when the PN extracted from a received frame is less than or equal to the current replay counter value for the frame's MSDU or A-MSDU priority and frame type. A transmitter shall not use IEEE 802.11 MSDU or A-MSDU priorities without ensuring that the receiver supports the required number of replay counters. The transmitter shall not reorder frames within a replay counter, but may reorder frames across replay counters. One possible reason for reordering frames is the IEEE 802.11 MSDU or A-MSDU priority.

f) If dot11RSNAProtectedManagementFramesActivated is true, the recipient shall maintain a single replay counter for received individually addressed robust management frames and shall use the PN from the received frame to detect replays. A replayed frame occurs when the PN from the frame is less than or equal to the current management frame replay counter value. The transmitter shall preserve the order of protected robust management frames sent to the same DA.

g) The receiver shall discard MSDUs, A-MSDUs, and MMPDUs whose constituent MPDU PN values are not sequential. A receiver shall discard any MPDU that is received with its PN less than or equal to the replay counter. When discarding a frame, the receiver shall increment by 1 the value of dot11RSNAStatsCCMPReplays for data frames or dot11RSNAStatsRobustMgmtCCMPReplays for robust management frames.

h) For MSDUs or A-MSDUs sent using the Block Ack feature, reordering of received MSDUs or A-MSDUs according to the Block Ack receiver operation (described in 9.21.4) is performed prior to replay detection.

11.4.4 Broadcast/Multicast Integrity Protocol (BIP)

11.4.4.1 BIP overview

BIP provides data integrity and replay protection for group addressed robust management frames after successful establishment of an IGTKSA (see 11.5.1.1.9).

BIP provides data integrity and replay protection, using AES-128 in CMAC Mode. NIST SP 800-38B defines the CMAC algorithm. All BIP processing uses AES with a 128-bit integrity key and a 128-bit block size, and a CMAC TLen value of 128 (16 octets). The CMAC output is truncated to 64 bits:

MIC = L(CMAC Output, 0, 64)

Where L is defined in 11.6.1.

BIP uses the IGTK to compute the MMPDU MIC. The authenticator shall distribute one new IGTK and IGTK PN (IPN) whenever it distributes a new GTK. The IGTK is identified by the MAC address of the transmitting STA plus an IGTK identifier that is encoded in the MME Key ID field.

11.4.4.2 BIP MMPDU format

The Management MIC element shall follow all of the other elements in the management frame body but precede the FCS. See 8.4.2.57 for the format of the Management MIC element. Figure 11-22 shows the BIP MMPDU.

IEEE 802.11 Header	Management Frame Body including MME	FCS

Figure 11-22—BIP Encapsulation

11.4.4.3 BIP AAD construction

The BIP Additional Authentication Data (AAD) shall be constructed from the MPDU header. The Duration field in the AAD shall be masked to 0. The AAD construction shall use a copy of the IEEE 802.11 header without the SC field for the MPDU, with the following exceptions:

a) FC—MPDU Frame Control field, with:
   1) Retry bit (bit 11) masked to 0
   2) Power Management bit (bit 12) masked to 0
   3) More Data bit (bit 13) masked to 0
b) A1—MPDU Address 1 field.
c) A2—MPDU Address 2 field.
d) A3—MPDU Address 3 field.

Figure 11-23 depicts the format of the AAD. The length of the AAD is 20 octets.

	FC	A1	A2	A3
Octets:	2	6	6	6

Figure 11-23—BIP AAD Construction

11.4.4.4 BIP replay protection

The MME Sequence Number field represents a sequence number whose length is 6 octets.

When management frame protection is negotiated, the receiver shall maintain a 48-bit replay counter for each IGTK. The receiver shall set the receive replay counter to the value of the IPN in the IGTK KDE provided by the Authenticator in either the 4-Way Handshake, FT 4-Way Handshake, FT Handshake, or Group Key Handshake. The transmitter may reinitialize the sequence counter when the IGTK is refreshed. See 11.4.4.5 and 11.4.4.6 for per packet BIP processing.

NOTE—When the IPN space is exhausted, the choices available to an implementation are to replace the IGTK or to end communications.

11.4.4.5 BIP transmission

When a STA transmits a protected group addressed robust management frame, it shall

a) Select the IGTK currently active for transmission of frames to the intended group of recipients and construct the MME (see 8.4.2.57) with the MIC field masked to 0 and the KeyID field set to the corresponding IGTK KeyID value. The transmitter shall insert a monotonically increasing non-negative integer into the MME IPN field.

b) Compute AAD as specified in 11.4.4.3.

c) Compute AES-128-CMAC over the concatenation of (AAD || Management Frame Body including MME), and insert the 64-bit output into the MME MIC field.

d) Compose the frame as the IEEE 802.11 header, management frame body, including MME, and FCS. The MME shall appear last in the frame body.

e) Transmit the frame.

11.4.4.6 BIP reception

When a STA with management frame protection negotiated receives a group addressed robust management frame protected by BIP, it shall

a) Identify the appropriate IGTK key and associated state based on the MME KeyID field. If no such IGTK exists, silently drop the frame.

b) Perform replay protection on the received frame. The receiver shall interpret the MME IPN field as a 48-bit unsigned integer. It shall compare this MME IPN integer value to the value of the receive replay counter for the IGTK identified by the MME Key ID field. If the integer value from the received MME IPN field is less than or equal to the replay counter value for this IGTK, the receiver shall discard the frame and increment the dot11RSNAStatsCMACReplays counter by 1. The receiver shall extract and save the received MIC value, and compute the AES-128-CMAC over the concatenation of (AAD || Management Frame Body including MME) with the MIC field masked to 0 in the MME. If the result does not match the received MIC value, then the receiver shall discard the frame and increment the dot11RSNAStatsCMACICVErrors counter by 1.

c) If the replay protection succeeds, compute AAD for this management frame, as specified in 11.4.4.3.

d) Extract and save the received MIC value, and compute the AES-128-CMAC over the concatenation of (AAD || Management Frame Body || MME) with the MIC field masked to 0 in the MME. If the result does not match the received MIC value, then the receiver shall discard the frame and increment the dot11RSNAStatsCMACICVErrors counter by 1.

e) Update the replay counter for the IGTK identified by the MME Key ID field with the integer value of the MME IPN field.

If management frame protection is negotiated, group addressed robust management frames that are received without BIP protection shall be discarded.

11.5 RSNA security association management

11.5.1 Security associations

11.5.1.1 Security association definitions

11.5.1.1.1 General

IEEE Std 802.11 uses the notion of a security association to describe secure operation. Secure communications are possible only within the context of a security association, as this is the context providing the state—cryptographic keys, counters, sequence spaces, etc.—needed for correct operation of the IEEE 802.11 cipher suites.

A security association is a set of policy(ies) and key(s) used to protect information. The information in the security association is stored by each party of the security association, needs to be consistent among all parties, and needs to have an identity. The identity is a compact name of the key and other bits of security association information to fit into a table index or an MPDU. The following types of security associations are supported by an RSN STA:

— PMKSA: A result of a successful IEEE 802.lX exchange, SAE authentication, preshared PMK information, or PMK cached via some other mechanism.

— PMK-R0 security association: A result of a successful FT initial mobility domain association.

— PMK-R1 security association: A result of a successful FT initial mobility domain association or FT authentication sequence.

— Mesh PMKSA: A result of successful completion of the active authentication protocol.

— PTKSA: A result of a successful 4-Way Handshake, FT 4-Way Handshake, or FT authentication sequence.

— Mesh TKSA: A result of a successful authenticated mesh peering exchange (AMPE).

— GTKSA: A result of a successful Group Key Handshake, 4-Way Handshake, FT 4-Way Handshake, or FT authentication sequence.

— IGTKSA: A result of a successful Group Key Handshake, successful 4-Way Handshake, FT 4-Way Handshake, or the Reassociation Response message of the Fast BSS Transition protocol.

— Mesh GTKSA: A result of a successful AMPE or mesh group key handshake.

— SMKSA: A result of a successful initial SMK Handshake.

— STKSA: A result of a successful 4-Way STK Handshake following the initial SMK Handshake or subsequent rekeying.

In order to set up a security association to a peer STA, a SME that does not know the security policy of the peer should send a Probe Request frame to the peer STA to find its security policy before setting up a security association to the peer STA.

In order to set up a security association to a peer STA, a STA that received a 4-Way Handshake but does not know the security policy of the peer should send a Probe Request frame to the peer STA to find its security policy before setting up a security association to the peer STA.

11.5.1.1.2 PMKSA

When the PMKSA is the result of a successful IEEE 802.1X authentication, it is derived from the EAP authentication and authorization parameters provided by the AS. When the PMKSA is the result of a successful SAE authentication, it is generated as a result of the successful completion of the SAE exchange. This security association is bidirectional. In other words, both parties use the information in the security association for both sending and receiving. The PMKSA is created by the Supplicant's SME when the EAP authentication completes successfully or the PSK is configured. The PMKSA is created by the Authenticator's SME when the PMK is created from the keying information transferred from the AS, when IEEE 802.1X authentication is utilized, when the SAE exchange successfully completes, or when the PSK is configured. The PMKSA is used to create the PTKSA. PMKSAs are cached for up to their lifetimes. The PMKSA consists of the following elements:

— PMKID, as defined in 11.6.1.3. The PMKID identifies the security association.
— Authenticator's or peer's MAC address.
— PMK.
— Lifetime, as defined in 11.6.1.3.
— AKMP.
— All authorization parameters specified by the AS or local configuration. This might include parameters such as the STA's authorized SSID.

11.5.1.1.3 PMK-R0 security association

The PMK-R0 security association is the result of a successful completion of the IEEE 802.1X authentication, SAE authentication, or use of PSK during the FT initial mobility domain association. This security association is bidirectional. It consists of the following elements:

— SSID
— MDID
— PMK-R0
— R0KH-ID
— PMKR0Name
— S0KH-ID
— PMK-R0 lifetime
— Pairwise cipher suite selector
— All authorization parameters specified by the AS or local configuration

11.5.1.1.4 PMK-R1 security association

The PMK-R1 security association is the result of

— A successful completion of the IEEE 802.1X authentication, SAE authentication, or use of PSK during the FT initial mobility domain association or
— A successful completion of the authentication phase in the fast BSS transition to the target AP

This security association is bidirectional. It consists of the following elements:

— SSID
— MDID

- PMK-R1
- PMK-R1 lifetime
- PMKR1Name
- R1KH-ID
- R0KH-ID
- PMKR0Name
- S0KH-ID
- S1KH-ID
- Pairwise cipher suite selector
- All authorization parameters specified by the AS or local configuration

11.5.1.1.5 Mesh PMKSA

The mesh PMKSA is the result of successful completion of the active authentication protocol. This security association is bidirectional. The two authenticated parties use the information in the security association for both sending and receiving. The mesh PMKSA is created by the Mesh STA's SME when the active authentication protocol completes successfully with the peer mesh STA. The mesh PMKSA is used to create the mesh TKSA. Mesh PMKSAs are cached for up to their lifetimes. Mesh PMKSAs contain the following elements, and are identified by their PMKID.

- PMKID, as defined in 11.3.5.4
- Mesh STA's MAC address
- Peer mesh STA's MAC address
- PMK
- AEK, as defined in 13.5.7
- Lifetime, as defined in 11.6.1.3
- Selected AKM suite (see 8.4.2.27.3)

11.5.1.1.6 PTKSA

The PTKSA is a result of the 4-Way Handshake, FT 4-Way Handshake, FT Protocol, or FT Resource Request Protocol. This security association is also bidirectional. PTKSAs are cached for the life of the PMKSA or PMK-R1 security association. Because the PTKSA is tied to the PMKSA or to a PMK-R1 security association, it only has the additional information from the 4-Way Handshake. For the PTKSA derived as a result of the 4-Way Handshake, there shall be only one PTKSA with the same Supplicant and Authenticator MAC addresses. For the PTKSA derived as a result of an initial mobility domain association or fast BSS transition, there shall be only one PTKSA with the same STA's MAC address and BSSID.

During the 4-Way Handshake defined in 11.6.6.5 and the FT 4-Way Handshake defined in 12.4.2, there is state created between Message 1 and Message 3 of the Handshake. This does not create a PTKSA until Message 3 is validated by the Supplicant and Message 4 is validated by the Authenticator.

During the FT authentication sequence defined in 12.8, the PTKSA is validated when Message 3 is validated by the R1KH and Message 4 is validated by the S1KH.

The PTKSA consists of the following elements:

- PTK
- Pairwise cipher suite selector
- Supplicant MAC address or STA's MAC address
- Authenticator MAC address or BSSID

- Key ID
- If FT key hierarchy is used,
  - R1KH-ID
  - S1KH-ID
  - PTKName

11.5.1.1.7 Mesh TKSA

The mesh TKSA is a result of the AMPE. This security association is also bidirectional. The mesh TKSA shall be deleted when the lifetime expires. The mesh TKSA contains the following elements:

- MTK, as defined in 13.5.7
- PMKID
- local mesh STA MAC address
- peer mesh STA MAC address
- local Link ID
- peer Link ID
- local nonce
- peer nonce
- Lifetime
- Pairwise cipher suite selector

11.5.1.1.8 GTKSA

The GTKSA results from a successful 4-Way Handshake, FT 4-Way Handshake, FT Protocol, FT Resource Request Protocol or the Group Key Handshake and is unidirectional. In an infrastructure BSS, there is one GTKSA, used exclusively for encrypting group addressed MPDUs that are transmitted by the AP and for decrypting group addressed transmissions that are received by the STAs. In an IBSS each STA defines its own GTKSA, which is used to encrypt its group addressed transmissions, and stores a separate GTKSA for each peer STA so that encrypted group addressed traffic received from other STAs may be decrypted. A GTKSA is created by the Supplicant's SME when Message 3 of the 4-Way Handshake is received or when Message 1 of the Group Key Handshake is received. The GTKSA is created by the Authenticator's SME when the SME changes the GTK and has sent the GTK to all STAs with which it has a PTKSA. A GTKSA consists of the following elements:

- Direction vector (whether the GTK is used for transmit or receive).
- Group cipher suite selector.
- GTK.
- Authenticator MAC address.
- Key ID.
- All authorization parameters specified by local configuration. This might include parameters such as the STA's authorized SSID.

When the GTK is used to encrypt individually addressed traffic (the selectable cipher suite is "Use group key"), the GTKSA is bidirectional.

11.5.1.1.9 IGTKSA

When management frame protection is enabled, a non-AP STA's SME creates an IGTKSA when it receives a valid Message 3 of the 4-Way Handshake or FT 4-Way Handshake, the Reassociation Response message of the Fast BSS Transition protocol with a status code indicating success, a Mesh Peering Open Message of the Authenticated Mesh Peering Exchange (AMPE) protocol, or a valid Message 1 of the Group Key Handshake. The Authenticator's SME creates an IGTKSA when it establishes or changes the IGTK with all STAs to which it has a valid PTKSA or MTKSA.

An IGTKSA consists of the following elements:

- Direction vector (whether the IGTK is used for transmit or receive)
- KeyID
- IGTK
- Authenticator MAC address

11.5.1.1.10 Mesh GTKSA

The mesh GTKSA results from a successful AMPE or mesh group key handshake, and is unidirectional. In an MBSS, each mesh STA defines its own "transmit mesh GTKSA," which is used to encrypt its group addressed transmissions. Also, each mesh STA stores a separate "receive mesh GTKSA" for each peer mesh STA so that encrypted group addressed traffic received from the peer mesh STAs may be decrypted.

A transmit mesh GTKSA is created by a mesh STA after the SME has changed the mesh GTK (MGTK) and the new MGTK has been sent to all peer mesh STAs. A receive mesh GTKSA is created by a mesh STA after successfully completing the AMPE in which a wrapped MGTK has been received, or after receiving a valid Message 1 of the mesh group key handshake. The receive mesh GTKSA shall be deleted when the lifetime expires or a new receive mesh GTKSA is created with the same Key ID for the same MGTK source mesh STA. See 13.6.1.

The MGTK and the GTK shall be independently selected from a uniform distribution. The MGTK source mesh STA MAC address in the mesh GTKSA shall not be the same as the Authenticator MAC address in the GTKSA.

NOTE—The use of a distinct Transmit MGTK and ESS GTK with identical transmit MAC addresses is precluded by limitations on key rollover and reception by STAs in an ESS (see 13.11.5 for collocated mesh STA rules). If the distinct MGTKs were to use different Key IDs, then rollover would be impossible. Since the Key ID 0 is reserved for individually addressed frame transmission, there are only three available Key IDs, and the different MGTKs would contend for the single remaining Key ID upon rollover. If the distinct MGTKs were to use the same Key IDs, then STAs would incorrectly attempt to decrypt mesh broadcast traffic using the ESS GTK, causing error counters (such as dot11RSNAStatsCCMPDecryptErrors) to continuously increment. (See 11.8.2.6 for a description of the procedure for receiving encrypted frames.)

The mesh GTKSA contains the following:

- MGTK
- MGTK source mesh STA MAC address (mesh STA that uses this GTK to encrypt transmissions)
- Group Cipher Suite Selector
- Lifetime
- Direction vector (whether this is a receive mesh GTKSA or transmit mesh GTKSA)
- Key Index

11.5.1.1.11 SMKSA

An SMKSA is the result of a successful SMK Handshake by the initiator STA (described in 11.6.8). It is derived from parameters provided by the STAs and AP. This security association is bidirectional between the initiator and the peer STA. In other words, both parties use the information in the security association for both sending and receiving. The SMKSA is created as a result of a successful SMK Handshake (see 11.6.8). The SMKSA is used to create the STKSA. The SMKSA consists of the following elements:

- SMKID, as defined in 11.6.8. The SMKID identifies the security association.

- BSSID
- Initiator MAC address
- Peer MAC address
- SMK
- Lifetime, as defined in 11.6.8.
- Pairwise cipher suite selector list, as proposed by initiator STA
- Pairwise cipher suite selector, as selected by peer STA

11.5.1.1.12 STKSA

The STKSA is a result of successful completion of the 4-Way STK Handshake. This security association is bidirectional between the initiator and the peer STAs. The STKSA is used to create session keys to protect this STSL. STKSAs are cached for the life of the SMKSA or until the STSL ends, whichever comes first. There shall be only one STKSA with the same initiator STA and peer MAC addresses at any one time. STKSA is created as a result of PeerKey Handshake (see 11.6.8). The STKSA consists of the following elements:

- STK
- Pairwise cipher suite selector
- Initiator MAC address
- Peer MAC address
- Key ID

11.5.1.2 TPKSA

The TPKSA results from a successful completion of the TDLS Peer Key Handshake. This security association is bidirectional between the TDLS initiator STA and the TDLS responder STA. The TPKSA is used to create session keys to protect this TDLS session. The TPKSA is cached per the lifetime indicated in the TDLS Peer Key Handshake or until the TDLS direct link is torn down, whichever comes first.

The TPKSA consist of the following:

- MAC addresses of the TDLS initiator STA and the TDLS responder STA
- Pairwise cipher suite selector
- TPK Lifetime
- TPK
- Link Identifier

11.5.1.3 Security association life cycle

11.5.1.3.1 General

A STA can operate in either an ESS or in an IBSS, and a security association has a distinct life cycle for each.

11.5.1.3.2 Security association in an ESS

In an ESS there are two cases:

- Initial contact between the STA and the ESS
- Roaming by the STA within the ESS

A STA and AP establish an initial security association via the following steps:

a) The STA selects an authorized ESS by selecting among APs that advertise an appropriate SSID.

b) The STA then performs IEEE 802.11 authentication followed by association to the chosen AP. Confirmation of security parameters takes place during association. A STA performing IEEE 802.1X authentication uses Open System authentication. A STA performing secure password-based, or PSK, authentication uses SAE authentication.

NOTE 1—It is possible for more than one PMKSA to exist. As an example, a second PMKSA might come into existence through PMKSA caching. A STA might leave the ESS and flush its cache. Before its PMKSA expires in the AP's cache, the STA returns to the ESS and establishes a second PMKSA from the AP's perspective.

NOTE 2—An attack altering the security parameters is detected by the key derivation procedure.

NOTE 3—IEEE 802.11 Open System authentication provides no security, but is included to maintain backward compatibility with the IEEE 802.11 state machine (see 10.3).

c) SAE authentication provides mutual authentication and derivation of a PMK. If Open System authentication is chosen instead, the Authenticator or the Supplicant initiates IEEE 802.1X authentication. The EAP method used by IEEE Std 802.1X-2004 needs to support mutual authentication, as the STA needs assurance that the AP is a legitimate AP.

NOTE 1—Prior to the completion of IEEE 802.1X authentication and the installation of keys, the IEEE 802.1X Controlled Port in the AP blocks all data frames. The IEEE 802.1X Controlled Port returns to the unauthorized state and blocks all data frames before invocation of an MLME-DELETEKEYS.request primitive. The IEEE 802.1X Uncontrolled Port allows IEEE 802.1X frames to pass between the Supplicant and Authenticator. Although IEEE Std 802.1X-2004 does not require a Supplicant Controlled Port, this standard assumes that the Supplicant has a Controlled Port in order to provide the needed level of security. Supplicants without a Controlled Port compromise RSN security and are not used.

NOTE 2—Any secure network cannot support promiscuous association, e.g., an unsecured operation of IEEE Std 802.11. A trust relationship is needed between the STA and the AS of the targeted SSID prior to association and secure operation, in order for the association to be trustworthy. The reason is that an attacker can deploy a rogue AP just as easily as a legitimate network provider can deploy a legitimate AP, so some sort of prior relationship is necessary to establish credentials between the ESS and the STA.

d) The last step is key management. The authentication process, whether SAE authentication utilizing IEEE 802.11 authentication frames or IEEE 802.1X authentication utilizing data frames post association, creates cryptographic keys shared between the cryptographic endpoints—the AP and STA, or the IEEE 802.1X AS and the STA, when using SAE or IEEE 802.1X, respectively. When using IEEE 802.1X, the AS transfers these keys to the AP, and the AP and STA uses one of the key confirmation handshakes, e.g., the 4-Way Handshake or FT 4-Way Handshake, to complete security association establishment. When using SAE authentication there is no AS and therefore no key transfer; the 4-way Handshake is performed directly between the AP and STA. The key confirmation handshake indicates when the link has been secured by the keys and is ready to allow normal data traffic and protected robust management frames.

When FT is not enabled, a STA roaming within an ESS establishes a new PMKSA by one of the four schemes:

— In the case of (re)association followed by IEEE 802.1X or PSK authentication, the STA repeats the same actions as for an initial contact association, but its Supplicant also deletes the PTKSA when it roams from the old AP. The Supplicant also deletes the PTKSA when it disassociates/ deauthenticates from all BSSIDs in the ESS.

— In the case of SAE authentication followed by (re)association, the STA repeats the same actions as for initial contact association, but the non-AP STA also deletes the PTKSA when it roams from the old AP. Note that a STA can take advantage of the fact that it can perform SAE authentication to multiple APs while maintaining a single association with one AP, and then use any of the PMKSAs created during authentication to effect a fast BSS transition.

— A STA (AP) can retain PMKs for APs (STAs) in the ESS to which it has previously performed a full IEEE 802.1X authentication or SAE authentication. If a STA wishes to roam to an AP for which it has cached one or more PMKSAs, it can include one or more PMKIDs in the RSNE of its (Re)Association Request frame. An AP that has retained the PMK for one or more of the PMKIDs can proceed with the 4-Way Handshake. The AP shall include the PMKID of the selected PMK in Message 1 of the 4-Way Handshake. If none of the PMKIDs of the cached PMKSAs matches any of the supplied PMKIDs, or if the AKM of the cached PMKSA differs from that offered in the (Re)Association Request, or if the PMK in the cached PMKSA is no longer valid, then the Authenticator, in the case of Open System authentication, shall perform another IEEE 802.1X authentication and, in the case of SAE authentication, shall transmit a Deauthentication frame to the STA. Similarly, if the STA fails to send a PMKID, the STA and AP need to perform a full IEEE 802.1X authentication.

— A STA already associated with the ESS can request its IEEE 802.1X Supplicant to authenticate with a new AP before associating to that new AP. The normal operation of the DS via the old AP provides the communication between the STA and the new AP. The SME delays reassociation with the new AP until IEEE 802.1X authentication completes via the DS. If IEEE 802.1X authentication completes successfully, then PMKSAs shared between the new AP and the STA are cached, thereby enabling the possible usage of reassociation without requiring a subsequent full IEEE 802.1X authentication procedure.

The MLME-DELETEKEYS.request primitive destroys the temporal keys established for the security association so that they cannot be used to protect subsequent IEEE 802.11 traffic. An SME uses this primitive when it deletes a PTKSA, GTKSA, or IGTKSA.

11.5.1.3.3 Security association in an IBSS

In an IBSS utilizing IEEE 802.11 Open System authentication and IEEE 802.1X, when a STA's SME establishes a security association with a peer STA, it creates both an IEEE 802.1X Supplicant and Authenticator for the peer. A STA in such an IBSS might also receive IEEE 802.1X messages from a previously unknown MAC address.

In an IBSS utilizing IEEE 802.11 SAE authentication, a STA creates a security association for a peer upon successful SAE authentication.

Any STA within an IBSS may decline to form a security association with a STA joining the IBSS. An attempt to form a security association may also fail because, for example, the peer uses a different PSK or password from what the STA expects.

In an IBSS each STA defines its own group key, i.e., GTK, to secure its group addressed transmissions. Each STA shall use either the 4-Way Handshake or the Group Key Handshake to distribute its transmit GTK to its new peer STA. When the STA generates a new GTK, it also uses the Group Key Handshake to distribute the new GTK to each established peer.

11.5.1.3.4 Security association in an MBSS

In order to create a secure peering, mesh STAs first authenticate each other and create a mesh PMKSA. This can be done using either SAE or IEEE 802.1X. Mesh STAs shall support SAE authentication (see 11.3). Optionally, mesh STAs may support IEEE 802.1X authentication (see 4.10).

When dot11MeshActiveAuthenticationProtocol is sae (1), the scanning mesh STA shall initiate SAE to the candidate mesh STA. If SAE terminates unsuccessfully, the scanning mesh STA shall terminate the peering establishment procedure. Otherwise, the PMK that results from successful SAE authentication shall be used to create a mesh PMKSA.

When dot11MeshActiveAuthenticationProtocol is ieee8021x (2), then the scanning mesh STA shall initiate the MPM protocol to establish a peering. If the MPM protocol fails then the scanning mesh STA shall terminate the peering establishment procedure. Otherwise, IEEE 802.1X authentication shall be performed between the two peers according to the following:

a) If only one mesh STA has the Connected to AS field set to 1, that STA shall act as the IEEE 802.1X authenticator and the other STA shall act as the IEEE 802.1X supplicant;
b) If both mesh STAs have the Connected to AS field set to 1, then the mesh STA with the higher MAC address shall act as the IEEE 802.1X authenticator and the other mesh STA shall act as the IEEE 802.1X supplicant (see 11.6.1 for MAC address comparison).

If IEEE 802.1X authentication fails, the peering establishment procedure shall be terminated and the peering established between the two mesh STAs shall be closed. Otherwise, the peering established between the two mesh STAs shall be closed and a mesh PMKSA shall be created using the PMK that resulted from the successful IEEE 802.1X authentication.

11.5.2 RSNA selection

A STA prepared to establish RSNAs shall advertise its capabilities by including the RSNE in Beacon and Probe Response messages. The included RSNE shall specify all the authentication and cipher suites enabled by the STA’s policy. A STA shall not advertise any authentication or cipher suite that is not enabled.

The SME shall utilize the MLME-SCAN.request primitive to identify neighboring STAs that assert robust security and advertise an SSID identifying an authorized ESS or IBSS. A STA may decline to communicate with STAs that fail to advertise an RSNE in their Beacon and Probe Response frames or that do not advertise an authorized SSID. A STA may also decline to communicate with other STAs that do not advertise authorized authentication and cipher suites within their RSNEs.

A STA shall advertise the same RSNE in both its Beacon and Probe Response frames.

NOTE—Whether a STA with robust security enabled attempts to communicate with a STA that does not include the RSNE is a matter of policy.

A STA shall observe the following rules when processing an RSNE:

- A STA shall advertise the highest version it supports.
- A STA shall request the highest Version field value it supports that is less than or equal to the version advertised by the peer STA.
- Two peer STAs without overlapping supported Version field values shall not use RSNA methods to secure their communication.
- A STA shall ignore suite selectors that it does not recognize.

11.5.3 RSNA policy selection in an ESS

RSNA policy selection in an ESS utilizes the normal IEEE 802.11 association procedure. RSNA policy selection is performed by the associating STA. The STA does this by including an RSNE in its (Re)Association Requests.

In an RSN, an AP shall not associate with pre-RSNA STAs, i.e., with STAs that fail to include the RSNE in the Association or Reassociation Request frame.

An SME initiating an association shall insert an RSNE into its (Re)Association Request via the MLME-ASSOCIATE.request or MLME-REASSOCIATE.request primitive, when the targeted AP indicates RSNA support. The initiating STA’s RSNE shall include one authentication and pairwise cipher suite from among those advertised by the targeted AP in its Beacon and Probe Response frames. It shall also specify the group cipher suite specified by the targeted AP. If at least one RSNE field from the AP's RSNE fails to overlap with any value the STA supports, the STA shall decline to associate with that AP. An HT STA shall eliminate TKIP as a choice for the pairwise cipher suite if CCMP is advertised by the AP or if the AP included an HT Capabilities element in its Beacon and Probe Response frames. The elimination of TKIP as a choice for the pairwise cipher suite may result in a lack of overlap of the remaining pairwise cipher suite choices, in which case the STA shall decline to create an RSN association with that AP.

If an RSNA-capable AP receives a (Re)Association Request including an RSNE and if it chooses to accept the association as a secure association, then it shall use the authentication and pairwise cipher suites in the (Re)Association Request, unless the AP includes an optional second RSNE in Message 3 of the 4-Way Handshake. If the second RSNE is supplied in Message 3, then the pairwise cipher suite used by the security association, if established, shall be the pairwise cipher from the second RSNE.

In order to accommodate local security policy, a STA may choose not to associate with an AP that does not support any pairwise cipher suites. An AP indicates that it does not support any pairwise keys by advertising "Use group key" as the pairwise cipher suite selector.

NOTE—When an ESS uses PSKs, STAs negotiate a pairwise cipher. However, any STA in the ESS can derive the pairwise keys of any other that uses the same PSK by capturing the first two messages of the 4-Way Handshake. This provides malicious insiders with the ability to eavesdrop as well as the ability to establish a man-in-the-middle attack.

An RSNA-enabled AP shall use Table 11-2 and the values of the Management Frame Protection Capable (MFPC) and Management Frame Protection Required (MFPR) bits advertised in the RSNEs to determine if it may associate with a non-AP STA. An RSNA enabled non-AP STA shall use Table 11-2 and the values of the Management Frame Protection Capable and Management Frame Protection Required bits advertised in the RSNEs to determine if it may associate with an AP. Management frame protection is enabled when dot11RSNAProtectedManagementFramesActivated is set to 1. Management frame protection is negotiated when an AP and non-AP STA set the Management Frame Protection Capable field to 1 in their respective RSNEs in the (re)association procedure, and both parties confirm the Management Frame Protection Capable bit set to 1 in the 4-Way Handshake, FT 4-Way Handshake, or the FT Fast BSS Transition protocol.

Table 11-2—Robust management frame selection in an ESS

AP MFPC	AP MFPR	STA MFPC	STA MFPR	AP Action	STA Action
0	0	0	0	The AP may associate with the STA	The STA may associate with the AP
1	0	0	0	The AP may associate with the STA	The STA may associate with the AP
1	0 or 1	1	0 or 1	The AP may associate with the STA	The STA may associate with the AP
1	1	0	0	The AP shall reject associations from the STA with the Status Code "Robust management frame policy violation"	The STA shall not associate with the AP
0	0	1	1	No action	The STA shall not try to associate with the AP
0	0	1	0	The AP may associate with the STA	The STA may associate with the AP

Table 11-2—Robust management frame selection in an ESS *(continued)*

AP MFPC	AP MFPR	STA MFPC	STA MFPR	AP Action	STA Action
1	0 or 1	0	1	The STA advertises an invalid setting. The AP shall reject associations from the STA with the Status Code "Robust management frame policy violation"	The STA shall not try to associate with the AP
0	1	1	0 or 1	No action	The AP advertises an invalid setting. The STA shall not try to associate with the AP

11.5.4 TSN policy selection in an ESS

In a TSN, an RSN STA shall include the RSNE in its (Re)Association Requests.

An RSNA-capable AP configured to operate in a TSN shall include the RSNE and may associate with both RSNA and pre-RSNA STAs. In other words, an RSNA-capable AP shall respond to an associating STA that includes the RSNE just as in an RSN.

If an AP operating within a TSN receives a (Re)Association Request without an RSNE, its IEEE 802.1X Controlled Port shall initially be blocked. The SME shall unblock the IEEE 802.1X Controlled Port when WEP has been enabled.

11.5.5 RSNA policy selection in an IBSS and for DLS

In an IBSS all STAs use a single group cipher suite, and all STAs support a common subset of pairwise cipher suites. However, the SMEs of any pair of non-HT STAs may negotiate to use any common pairwise cipher suite they both support. Each STA shall include the group cipher suite and its list of pairwise cipher suites in its Beacon and Probe Response messages. Two STAs shall not establish a PMKSA unless they have advertised the same group cipher suite. Similarly, the two STAs shall not establish a PMKSA if the STAs have advertised disjoint sets of pairwise cipher suites.

An HT STA that is in an IBSS or that is transmitting frames through a direct link shall eliminate TKIP as a choice for the pairwise cipher suite if CCMP is advertised by the other STA or if the other STA included an HT Capabilities element in any of its Beacon, Probe Response, DLS Request, or DLS Response messages.

NOTE—The elimination of TKIP as a choice for the pairwise cipher suite might result in a lack of overlap of the remaining pairwise cipher suite choices, in which case the STAs do not exchange encrypted frames.

In order to set up a security association with a peer STA, the SME of an IBSS STA that does not know the peer's policy needs first to obtain the peer's security policy using a Probe Request frame. The SME entities of the two STAs select the pairwise cipher suites using one of the 4-Way Handshakes. The SMEs of each pair of STAs within an IBSS may use the EAPOL-Key 4-Way Handshake to select a pairwise cipher suite. As specified in 11.6.2, Message 2 and Message 3 of the 4-Way Handshake convey an RSNE. The Message 2 RSNE includes the selected pairwise cipher suite, and Message 3 includes the RSNE that the STA would send in a Probe Response frame.

If the 4-Way Handshake is successfully completed, then the pair of STAs shall use the pairwise cipher suite specified in Message 2 of the 4-Way Handshake initiated by the Authenticator STA with the higher MAC address (see 11.6.1).

The SME shall check that the group cipher suite and AKMP match those in the Beacon and Probe Response frames for the IBSS.

NOTE 1—The RSNEs in Message 2 and Message 3 are not the same as in the Beacon frame. The group cipher and AKMP are the same, but the pairwise ciphers might differ because Beacon frames from different STAs might advertise different pairwise ciphers. Thus, STAs in an IBSS use the same AKM suite and group cipher, while different pairwise ciphers might be used between STA pairs.

NOTE 2—When an IBSS network uses PSKs, STAs can negotiate a pairwise cipher. However, any STA in the IBSS can derive the PTKs of any other that uses the same PSK by capturing the first two messages of the 4-Way Handshake. This provides malicious insiders with the ability to eavesdrop as well as the ability to establish a man-in-the-middle attack.

To establish a connection with a peer STA, an RSNA enabled STA that implements management frame protection shall use Table 11-3 and the MFPC and MFPR values advertised in the RSNEs exchanged in the 4-Way Handshake initiated by the Authenticator of the STA with the larger MAC address to determine if the communication is allowed. Management frame protection is enabled when dot11RSNAProtectedManagementFramesActivated is set to 1. The STAs negotiate protection of management frames when the both STAs set the Management Frame Protection Capable subfield to 1 during the 4-Way Handshake.

Table 11-3—Robust management frame selection in an IBSS

MFPC	MFPR	Peer STA MFPC	Peer STA MFPR	STA Action
0	0	0	0	The STA may exchange data with the peer STA.
1	0	0	0	The STA may exchange data with the peer STA.
1	0 or 1	1	0 or 1	The STA may exchange data with the peer STA.
1	1	0	0	The STA shall not exchange data with the peer STA and shall reject security association attempts from the peer STA with the Reason Code “Robust management frame policy violation.”
0	0	1	1	The STA shall not exchange data with the peer STA and shall reject security association attempts from the peer STA with the Reason Code “Robust management frame policy violation.”
0	0	1	0	The STA may establish a security association with the peer STA.
1	0 or 1	0	1	The STA shall not establish a security association with the peer STA and shall reject security association attempts from the peer STA with the Status Code “Robust management frame policy violation” because the peer STA is advertising an invalid setting. The STA shall not exchange data with the peer STA.
0	1	1	0 or 1	The peer STA shall not establish a security association with the peer STA and shall reject security association attempts from the STA with the Status Code “Robust management frame policy violation” because the STA is advertising an invalid setting.

11.5.6 TSN policy selection in an IBSS

Pre-RSNA STAs generate Beacon and Probe Response frames without an RSNE and ignore the RSNE because it is unknown to them. This allows an RSNA STA to identify the pre-RSNA STAs from which it has received Beacon and Probe Response frames.

If an RSNA STA's SME instead identifies a possible IBSS member on the basis of a received group addressed message, via MLME-PROTECTEDFRAMEDROPPED.indication primitive, it cannot identify the peer's security policy directly. The SME might attempt to obtain the peer STA's security policy via a Probe Request frame.

11.5.7 RSNA policy selection in an MBSS

RSNA policy is advertised in Beacon frames and Probe Response frames. A mesh STA identifies a candidate peer by parsing its neigbor STA's Beacon frames and Probe Response frames (see 13.2).

All mesh STAs in an MBSS use the same group cipher suite. Mesh STAs establish authenticated peerings with each other using the AMPE protocol (see 13.5). In AMPE, mesh STAs negotiate a pairwise cipher suite, and establish a pairwise MTKSA, to protect individually addressed frames and state a group cipher suite and establish an MGTKSA to process incoming group addressed frames from a peer.

The AMPE performs key confirmation of a secret, authenticated, and shared PMK derived by an authenticated key management protocol (see 11.3) and derives pairwise symmetric keys.

11.5.8 RSN management of the IEEE 802.1X Controlled Port

When the policy selection process chooses IEEE 802.1X authentication, this standard assumes that IEEE 802.1X Supplicants and Authenticators exchange protocol information via the IEEE 802.1X Uncontrolled port. The IEEE 802.1X Controlled Port is blocked from passing general data traffic between the STAs until an IEEE 802.1X authentication procedure completes successfully over the IEEE 802.1X Uncontrolled Port. The security of an RSNA depends on this assumption being true.

In an ESS, the STA indicates the IEEE 802.11 link is available by invoking the MLME-ASSOCIATE.confirm or MLME-REASSOCIATE.confirm primitive. This signals the Supplicant that the MAC has transitioned from the disabled to enabled state. At this point, the Supplicant's Controlled Port is blocked, and communication of all non-IEEE-802.1X MSDUs sent or received via the port is not authorized.

In an ESS, the AP indicates that the IEEE 802.11 link is available by invoking the MLME-ASSOCIATE.indication or MLME-REASSOCIATE.indication primitive. At this point the Authenticator's Controlled Port corresponding to the STA's association is blocked, and communication of all non-IEEE-802.1X MSDUs sent or received via the Controlled Port is not authorized.

In an IBSS the STA shall block all IEEE 802.1X ports at initialization. Communication of all non-IEEE-802.1X MSDUs sent or received via the Controlled Port is not authorized.

This standard assumes each Controlled Port remains blocked until the IEEE 802.1X state variables portValid and keyDone both become true. This assumption means that the IEEE 802.1X Controlled Port discards MSDUs sent across the IEEE 802.11 channel prior to the installation of cryptographic keys into the MAC. This protects the STA's host from forged MSDUs written to the channel while it is still being initialized.

The MAC does not distinguish between MSDUs for the Controlled Port, and MSDUs for the Uncontrolled Port. In other words, IEEE 802.1X EAPOL frames are encrypted only after invocation of the MLME-SETPROTECTION.request primitive.

This standard assumes that IEEE Std 802.1X-2004 does not block the Controlled Port when authentication is triggered through reauthentication. During IEEE 802.1X reauthentication, an existing RSNA can protect all MSDUs exchanged between the STAs. Blocking MSDUs is not required during reauthentication over an RSNA.

11.5.9 RSNA authentication in an ESS

11.5.9.1 General

When establishing an RSNA in a non-FT environment or during an FT initial mobility domain association, a STA shall use IEEE 802.11 SAE authentication or Open System authentication prior to (re)association.

SAE authentication is initiated when a STA's MLME-SCAN.confirm primitive finds another AP within the current ESS that advertises support for SAE in its RSNE.

IEEE 802.1X authentication is initiated by any one of the following mechanisms:

- If a STA negotiates to use IEEE 802.1X authentication during (re)association, the STA's management entity may respond to the MLME-ASSOCIATE.confirm (or indication) or MLME-REASSOCIATE.confirm (or indication) primitive by requesting the Supplicant (or Authenticator) to initiate IEEE 802.1X authentication. Thus, in this case, authentication is driven by the STA's decision to associate and the AP's decision to accept the association.
- If a STA's MLME-SCAN.confirm primitive finds another AP within the current ESS, a STA may signal its Supplicant to use IEEE Std 802.1X-2004 to preauthenticate with that AP.

  NOTE—A roaming STA's IEEE 802.1X Supplicant can initiate preauthentication by sending an EAPOL-Start message via its old AP, through the DS, to a new AP.

- If a STA receives an IEEE 802.1X message, it delivers this to its Supplicant or Authenticator, which may initiate a new IEEE 802.1X authentication.

11.5.9.2 Preauthentication and RSNA key management

Preauthentication allows a STA to perform RSN authentication with an AP prior to attempting (re)association. This might reduce the time that the IEEE 802.1X port is not valid.

A STA shall not use preauthentication except when pairwise keys are employed. A STA shall not use preauthentication within the same mobility domain if AKM suite type 00-0F-AC:3 or 00-0F-AC:4 is used in the current association. Preauthentication shall not be used unless the new AP advertises the preauthentication capability in the RSNE.

When preauthentication is used, then:

a) Authentication is independent of roaming.

b) The Supplicant may authenticate with multiple APs at a time.

NOTE—Preauthentication might be useful as a performance enhancement, as reassociation does not include the protocol overhead of a full reauthentication when it is used.

Preauthentication uses the IEEE 802.1X protocol and state machines with EtherType 88-C7, rather than the EtherType 88-8E. Only IEEE 802.1X frame types EAP-Packet and EAPOL-Start are valid for preauthentication.

NOTE—Some IEEE 802.1X Authenticators might not bridge IEEE 802.1X frames, as suggested in C.1.1 of IEEE Std 802.1X-2004. Preauthentication uses a distinct EtherType to enable such devices to bridge preauthentication frames.

A Supplicant may initiate preauthentication when it has completed the 4-Way Handshake and configured the required temporal keys. To effect preauthentication, the Supplicant sends an IEEE 802.1X EAPOL-Start message with the DA being the BSSID of a targeted AP and the RA being the BSSID of the AP with which it is associated. The target AP shall use a BSSID equal to the MAC address of its Authenticator. As preauthentication frames do not use the IEEE 802.1X EAPOL EtherType field, the AP with which the STA is currently associated need not apply any special handling. The AP and the MAC in the STA shall handle these frames in the same way as other frames with arbitrary EtherType field values that require distribution via the DS.

An AP's Authenticator that receives an EAPOL-Start message via the DS may initiate IEEE 802.1X authentication to the STA via the DS. The DS forwards this message to the AP with which the STA is associated.

The result of preauthentication may be a PMKSA, if the IEEE 802.1X authentication completes successfully. The AKM shall be set to 00-0F-AC:1 in the PMKSA that results from preauthentication. If preauthentication produces a PMKSA, then, when the Supplicant's STA associates with the preauthenticated AP, the Supplicant may use the PMKSA with the 4-Way Handshake.

Successful completion of EAP authentication over IEEE 802.1X establishes a PMKSA at the Supplicant. The Authenticator has the PMKSA when the AS completes the authentication, passes the keying information (the master session key (MSK), a portion of which is the PMK) to the Authenticator, and the Authenticator creates a PMKSA using the PMK. The PMKSA is inserted into the PMKSA cache. Therefore, if the Supplicant and Authenticator lose synchronization with respect to the PMKSA, the 4-Way Handshake fails. In such circumstances, dot11RSNA4WayHandshakeFailures shall be incremented.

A Supplicant may initiate preauthentication with any AP within its present ESS with preauthentication enabled regardless of whether the targeted AP is within radio range.

Even if a STA has preauthenticated, it is still possible that it may have to undergo a full IEEE 802.1X authentication, as the AP's Authenticator may have purged its PMKSA due to, for example, unavailability of resources, delay in the STA associating, etc.

11.5.9.3 Cached PMKSAs and RSNA key management

In a non-FT environment, a STA might retain PMKSAs it establishes as a result of previous authentication. The PMKSA cannot be changed while cached. The PMK in the PMKSA is used with the 4-Way Handshake to establish fresh PTKs.

If a STA in an ESS has determined it has a valid PMKSA with an AP to which it is about to (re)associate, it includes the PMKID for the PMKSA in the RSNE in the (Re)Association Request. Upon receipt of a (Re)Association Request with one or more PMKIDs, an AP checks whether its Authenticator has retained a PMK for the PMKIDs, whether the AKM in the cached PMKSA matches the AKM in the (Re)Association Request, and whether the PMK is still valid; and if so, it shall assert possession of that PMK by beginning the 4-Way Handshake after association has completed. If the Authenticator does not have a PMK for the PMKIDs in the (Re)Association Request, its behavior depends on how the STA performed IEEE 802.11 authentication. If the STA performed SAE authentication, then the AP STA shall send a Deauthentication frame. If the STA performed Open System authentication, it begins a full IEEE 802.1X authentication after association has completed.

If both sides assert possession of a cached PMKSA, but the 4-Way Handshake fails, both sides may delete the cached PMKSA for the selected PMKID.

If a STA roams to an AP with which it is preauthenticating and the STA does not have a PMKSA for that AP, the STA needs to initiate a full IEEE 802.1X EAP authentication.

11.5.10 RSNA authentication in an IBSS

When authentication is used in an IBSS, it is driven by each STA wishing to establish communications. The management entity of this STA chooses a set of STAs with which it might need to authenticate and then may cause the MAC to send an IEEE 802.11 Open System authentication message to each targeted STA.

Candidate STAs can be identified from Beacon frames, Probe Response frames, and data frames from the same BSSID. Before communicating with STAs identified from data frames, the security policy of the STAs may be obtained, e.g., by sending a Probe Request frame to the STA and obtaining a Probe Response frame. Targeted STAs that wish to respond may return an IEEE 802.11 Open System authentication message to the initiating STA.

When IEEE 802.1X authentication is used, the STA management entity requests its local IEEE 802.1X entity to create a Supplicant PAE for the peer STA. The Supplicant PAE initiates the authentication to the peer STA by sending an EAPOL-Start message to the peer. The STA management entity also requests its local IEEE 802.1X entity to create an Authenticator PAE for the peer STA on receipt of the EAPOL-Start message. The Authenticator initiates authentication to the peer STA by sending an EAP-Request message or, if PSK mode is in effect, Message 1 of the 4-Way Handshake.

Upon initial authentication between any pair of STAs, data frames, other than IEEE 802.1X messages, are not allowed to flow between the pair of STAs until both STAs in each pair of STAs have successfully completed AKM and have provided the supplied encryption keys.

Upon the initiation of an IEEE 802.1X reauthentication by any STA of a pair of STAs, data frames continue to flow between the STAs while authentication completes. Upon a timeout or failure in the authentication process, the Authenticator of the STA initiating the reauthentication shall cause a Deauthentication message to be sent to the Supplicant of the STA targeted for reauthentication. The Deauthentication message causes both STAs in the pair of STAs to follow the deauthentication procedure defined in 10.3.4.4 and 10.3.4.5.

The IEEE 802.1X reauthentication timers in each STA are independent. If the reauthentication timer of the STA with the higher MAC address (see 11.6.1 for MAC comparison) triggers the reauthentication via its Authenticator, its Supplicant shall send an EAPOL-Start message to the authenticator of the STA with the lower MAC address (see 11.6.1 for MAC comparison) to trigger reauthentication on the other STA. This process keeps the pair of STAs in a consistent state with respect to derivation of fresh temporal keys upon an IEEE 802.1X reauthentication.

When it receives an MLME-AUTHENTICATE.indication primitive due to an Open System authentication request, the IEEE 802.11 management entity on a targeted STA shall, if it intends to set up a security association with the peer STA, request its Authenticator to begin IEEE 802.1X authentication, i.e., to send an EAP-Request/Identity message or Message 1 of the 4-Way Handshake to the Supplicant.

The EAPOL-Key frame is used to exchange information between the Supplicant and the Authenticator to negotiate a fresh PTK. The 4-Way Handshake produces a single PTK from the PMK. The 4-Way Handshake and Group Key Handshake use the PTK to protect the GTK as it is transferred to the receiving STA.

Password or PSK authentication may also be used in an IBSS. When a single password or PSK is shared among the IBSS STAs, an SAE capable STA wishing to establish communication with a STA that advertises support for SAE in Beacon and Probe Response frames invokes SAE authentication, and upon successful conclusion of SAE, sends 4-Way Handshake Message 1 to the target STA. If the STA does not support SAE authentication or the target STA does not advertise support for SAE in Beacon and Probe Response frames, the STA may use the PSK as a PMK and initiate the 4-Way Handshake by sending a 4-Way Handshake Message 1 to the target STA. In either case, the targeted STA responds to Message 1 with Message 2 of the 4-Way Handshake and begins its 4-Way Handshake by sending Message 1 to the initiating STA. The two 4-Way Handshakes establish PTKSAs and GTKSAs to be used between the initiating STA and the targeted STA. PSK PMKIDs have security vulnerabilities when used with low-entropy keys and should be used only after taking this into account.

The model for security in an IBSS is not general. In particular, it assumes the following:

a) The sets of use cases for which the authentication procedures described in this subclause are valid are as follows:
   1) Password or PSK-based authentication using SAE to perform mutual authentication and generation of a shared PMK.
   2) An alternate form of PSK-based authentication, typically managed by the pass-phrase hash method as described in M.4. This method has security vulnerabilities and should only be used when SAE authentication is not possible.
   3) EAP-based authentication, using credentials that have been issued and preinstalled on the STAs within a common administrative domain, such as a single organization
b) All of the STAs are in direct radio communication. In particular, there is no routing, bridging, or forwarding of traffic by a third STA to effect communication. This assumption is made, because the model makes no provision to protect IBSS topology information from tampering by one of the members.

11.5.11 RSNA authentication in an MBSS

When establishing an RSNA in an MBSS, a mesh STA shall use IEEE 802.11 SAE authentication (see 11.3), or optionally IEEE 802.1X authentication, prior to establishment of an authenticated peering. An RSNA security association, called a Mesh PMKSA, is created upon successful completion of authentication.

Authentication using IEEE 802.11 SAE authentication is based on a password. A password is required to be shared between two mesh STAs in order to successfully complete authentication. This password can be pairwise – each pair of mesh STAs in an MBSS has a unique password – or it can be shared-all mesh STAs in the MBSS share the same password.

Due to the security properties of IEEE 802.11 SAE authentication, an adversary has no greater possibility in determining a shared password than in determining a pairwise password. The potential for misuse, though, is greater if a shared password becomes known to an adversary because an unlimited number of mesh STAs under the control of the adversary can be added to the MBSS.

11.5.12 RSNA key management in an ESS

When the IEEE 802.1X authentication completes successfully, this standard assumes that the STA's IEEE 802.1X Supplicant and the IEEE 802.1X AS share a secret, called a PMK. In a non-FT environment, the AS transfers the PMK, within the MSK, to the AP, using a technique that is outside the scope of this standard; the derivation of the PMK from the MSK is EAP-method-specific. With the PMK in place, the AP initiates a key confirmation handshake with the STA. The key confirmation handshake sets the IEEE 802.1X state variable portValid (as described in IEEE Std 802.1X-2004) to TRUE.

When SAE authentication completes, both STAs share a PMK. With this PMK in place, the AP initiates the key confirmation handshake with the STA.

The key confirmation handshake is implemented by the 4-Way Handshake. The purposes of the 4-Way Handshake are as follows:

a) Confirm the existence of the PMK at the peer.
b) Ensure that the security association keys are fresh.
c) Synchronize the installation of temporal keys into the MAC.
d) Transfer the GTK from the Authenticator to the Supplicant.
e) Confirm the selection of cipher suites.

NOTE 1—It is possible to forge message 1 of the 4-Way Handshake. However, the forgery attempt is detected in the failure of the 4-Way Handshake.

NOTE 2—Neither the AP nor the STA can use the PMK for any purpose but the one specified herein without compromising the key. If the AP uses it for another purpose, then the STA can masquerade as the AP; similarly if the STA reuses the PMK in another context, then the AP can masquerade as the STA.

The Supplicant and Authenticator signal the completion of key management by utilizing the MLME-SETKEYS.request primitive to configure the agreed-upon temporal pairwise key into the IEEE 802.11 MAC and by calling the MLME-SETPROTECTION.request primitive to enable its use.

A second key exchange, the Group Key Handshake, is also defined. It distributes a subsequent GTK. The AP's Authenticator can use the Group Key Handshake to update the GTK at the Supplicant. The Group Key Handshake uses the EAPOL-Key frames for this exchange. When it completes, the Supplicant can use the MLME-SETKEYS.request primitive to configure the GTK into the IEEE 802.11 MAC.

11.5.13 RSNA key management in an IBSS

To establish a security association between two STAs in an IBSS, each STA's SME has an accompanying IEEE 802.1X Authenticator and Supplicant. Each SME initiates the 4-Way Handshake from the Authenticator to the peer STA's Supplicant (see 11.5.10). Two separate 4-Way Handshakes are conducted.

The 4-Way Handshake is used to negotiate the pairwise cipher suites, as described in 11.5.5. The IEEE 802.11 SME configures the temporal key portion of the PTK into the IEEE 802.11 MAC. Each Authenticator uses the KCK and KEK portions of the PTK negotiated by the exchange it initiates to distribute its own GTK and if management frame protection is enabled, its own IGTK. Each Authenticator generates its own GTK and if management frame protection is enabled, its own IGTK, and uses either the 4-Way Handshake or the Group Key Handshake to transfer the GTK and if management frame protection is negotiated, the IGTK, to other STAs with whom it has completed a 4-Way Handshake. The pairwise key used between any two STAs shall be the pairwise key from the 4-Way Handshake initiated by the STA with the highest MAC address.

A STA joining an IBSS is required to adopt the security configuration of the IBSS, which includes the group cipher suite, pairwise cipher suite, AKMP, and if management frame protection is enabled, Group Management Cipher Suite (see 11.5.5). The STA shall not set up a security association with any STA having a different security configuration. The Beacon and Probe Response frames of the various STAs within an IBSS need to reflect a consistent security policy, as the beacon initiation rotates among the STAs.

A STA joining an IBSS shall support and advertise in the Beacon frame the security configuration of the IBSS, which includes the group cipher suite, advertised pairwise cipher suite, AKMP, and if management frame protection is enabled, Group Management Cipher Suite (see 11.5.5). The STA may use the Probe Request frame to discover the security policy of a STA, including additional individual cipher suites the STA supports. A STA shall ignore Beacon frames that advertise a different security policy. If enabled, management frame protection shall only be used as a required feature (MFPR) in an IBSS.

11.5.14 RSNA key management in an MBSS

Upon successful completion of the AMPE, a secure mesh peering is established between two mesh STAs. This secure mesh peering includes a mesh PMKSA and a mesh TKSA. Multiple mesh TKSAs may be created using a single mesh PMKSA (a limit to that number is a policy decision outside the scope of this standard).

A mesh TKSA is logically a child of the mesh PMKSA. A mesh TKSA shall be destroyed if the corresponding mesh PMKSA, which was used by the AMPE to create it, is destroyed. Mesh PMKSAs are limited by their lifetime (see 11.6.1.3).

11.5.15 RSNA security association termination

When a non-AP STA's SME receives a successful MLME-ASSOCIATE.confirm or MLME-REASSOCIATE.confirm primitive that is not part of a fast BSS transition or receives or invokes an MLME Disassociation or Deauthentication primitive, it deletes some security associations. Similarly, when an AP's SME

- receives an MLME-ASSOCIATE.indication or MLME-REASSOCIATE.indication primitive from a STA that has not negotiated management frame protection, or
- receives an MLME-ASSOCIATE.indication or MLME-REASSOCIATE.indication primitive from a STA that has negotiated management frame protection that a) has resulted in an MLME Association or Reassociation Response that is successful, and b) is not part of a fast BSS transition, or
- receives or invokes an MLME Disassociation or Deauthentication primitive,

it deletes some security associations. In the case of an ESS, the non-AP STA's SME shall delete the PTKSA, GTKSA, IGTKSA, SMKSA, any TPKSA, and any STKSA, and the AP's SME shall delete the PTKSA and invoke an STSL application teardown procedure for any of its STKSAs. An example of an STSL application teardown procedure is described in 10.7.4. In the case of an IBSS, the SME shall delete the PTKSA and the receive GTKSA and IGTKSA. Once the security associations have been deleted, the SME then invokes MLME-DELETEKEYS.request primitive to delete all temporal keys associated with the deleted security associations.

If a STA loses key state synchronization, it can apply the following rules to recover:

a) Any protected frame(s) received shall be discarded, and MLME-PROTECTEDFRAMEDROPPED.indication primitive is invoked.
b) If the STA is RSNA-enabled and has joined an IBSS, the SME shall execute the authentication procedure as described in 10.3.4.2.
c) If the STA is RSNA-enabled and has joined an ESS, the SME shall execute the deauthentication procedures as described in 10.3.4.4. However, if the STA has initiated the RSN security association, but has not yet invoked the MLME-SETPROTECTION.request primitive, then no additional action is required.

NOTE 1—There is a race condition between when MLME-SETPROTECTION.request primitive is invoked on the Supplicant and when it is invoked on the Authenticator. During this time, the STA might receive an MPDU that it is unable to decrypt; and the MPDU is discarded without a deauthentication occurring.

NOTE 2—Because the IEEE 802.11 null data MPDU does not derive from an MA-UNITDATA.request primitive, it is not protected.

If the selected AKMP fails between a STA and an AP that are associated, then both the STA and the AP shall invoke the MAC deauthentication procedure described in 10.3.4.4.

If the SMK Handshake fails between a pair of associated STAs and AP, then the STAs and the AP shall invoke an STSL application teardown procedure.

11.5.16 Protection of robust management frames

This subclause defines rules that shall be followed by STAs that implement Management Frame protection and have dot11RSNAEnable equal to true.

A STA with dot11RSNAProtectedManagementFramesActivated equal to false shall transmit and receive unprotected individually addressed robust management frames to and from any associated STA and shall discard protected individually addressed robust management frames received from any associated STA.

A STA with dot11RSNAProtectedManagementFramesActivated equal to true and dot11RSNAUnprotectedManagementFramesAllowed equal to true shall transmit and receive unprotected individually addressed robust management frames to and from any associated STA that advertised MFPC = 0 and shall discard protected individually addressed robust management frames received from any associated STA that advertised MFPC = 0.

A STA with dot11RSNAProtectedManagementFramesActivated equal to true and dot11RSNAUnprotectedManagementFramesAllowed equal to true shall transmit and receive protected individually addressed robust management frames to and from any associated STA that advertised MFPC = 1, shall discard unprotected individually addressed robust Action frames received from any STA that advertised MFPC = 1, and shall discard unprotected individually addressed Disassociation and Deauthentication frames received from a STA that advertised MFPC = 1 after the PTK and IGTK have been installed. The receiver shall process unprotected individually addressed Disassociation and Deauthentication frames before the PTK and IGTK are installed.

A STA with dot11RSNAProtectedManagementFramesActivated equal to true and dot11RSNAUnprotectedManagementFramesAllowed equal to false shall transmit and receive protected individually addressed robust Action frames to and from any STA, shall not transmit unprotected individually addressed robust Action frames to any STA, and shall discard unprotected individually addressed robust Action frames received from a STA after the PTK and IGTK have been installed. The receiver shall process unprotected individually addressed Disassociation and Deauthentication frames before the PTK and IGTK are installed.

A STA with dot11RSNAProtectedManagementFramesActivated equal to true shall protect transmitted group addressed robust management frames using the Group Management Cipher suite.

A STA with dot11RSNAProtectedManagementFramesActivated equal to true shall discard group addressed robust management frames received from any associated STA that advertised MFPC = 1 if the frames are unprotected or if a matching IGTK is not available.

A STA with dot11RSNAProtectedManagementFramesActivated equal to true and dot11RSNAUnprotectedManagementFramesAllowed equal to false shall discard received group addressed robust management frames that are unprotected or for which a matching IGTK is not available.

A STA with dot11RSNAProtectedManagementFramesActivated equal to false shall transmit group addressed robust management frames unprotected and shall ignore the protection on received group addressed robust management frames.

NOTE—BIP does not provide protection against forgery by associated and authenticated STAs. A STA that has left the group can successfully forge management frames until the IGTK is updated.

Protection of group addressed robust management frames shall be provided by a service in the MLME as described in 10.13.

Robust management frame protection cannot be applied until the PTK and IGTK has been established with the STA. A STA shall not transmit robust Action frames until it has installed the PTK for the peer STA, or in the case of group addressed frames, has installed the IGTK. The STA shall discard any robust Action frames received before the PTK and IGTK are installed. Action frames with “No” in the “Robust” column in Table 8-38 shall not be protected.

11.5.17 Robust management frame selection procedure

A STA with dot11RSNAProtectedManagementFramesActivated equal to true shall negotiate robust management frame protection with a STA that advertised MFPC = 1.

When a Public Action frame is transmitted for which a Protected Dual of Public Action frame is defined, (see 8.5.11), which variant (i.e., protected or not protected) is used depends on the setting of the "Protected" parameter of the corresponding MLME .request or .confirm primitive. Where there is no such parameter, the protected variant is used when management frame protection has been negotiated.

11.6 Keys and key distribution

11.6.1 Key hierarchy

11.6.1.1 General

RSNA defines the following key hierarchies:

a) Pairwise key hierarchy, to protect individually addressed traffic

b) GTK, a hierarchy consisting of a single key to protect group addressed traffic

NOTE—Pairwise key support with enhanced data cryptographic encapsulation mechanisms allows a receiving STA to detect MAC address spoofing and data forgery. The RSNA architecture binds the transmit and receive addresses to the pairwise key. If an attacker creates an MPDU with the spoofed TA, then the decapsulation procedure at the receiver generates an error. GTKs do not have this property.

c) Integrity GTK (IGTK), a hierarchy consisting of a single key to provide integrity protection for group addressed robust management frames

The description of the key hierarchies uses the following two functions:

— L(*Str*, *F*, *L*)From *Str* starting from the left, extract bits *F* to *F*+*L*–1, using the IEEE 802.11 bit conventions from 8.2.2.

— PRF-*n* Pseudorandom function producing *n* bits of output, defined in 11.6.1.2.

In an ESS, the IEEE 802.1X Authenticator MAC address (AA) and the AP’s BSSID are the same, and the Supplicant’s MAC address (SPA) and the STA’s MAC address are equal. For the purposes of comparison, the MAC address is encoded as 6 octets, taken to represent an unsigned binary number. The first octet of the MAC address shall be used as the most significant octet. The bit numbering conventions in 8.2.2 shall be used within each octet. This results in a 48-bit unsigned integer labelled b0 (least significant) to b47 (most significant), where the I/G bit is in b40.

An RSNA STA shall support at least one pairwise key for any <TA,RA> pair for use with enhanced data cryptographic encapsulation mechanisms. The <TA,RA> identifies the pairwise key, which does not correspond to any WEP key identifier.

In a a mixed environment, an AP may simultaneously communicate with some STAs using WEP with shared WEP keys and to STAs using enhanced data cryptographic encapsulation mechanisms with pairwise keys. The STAs running WEP use default keys 0–3 for shared WEP keys; the important point here is that WEP can still use WEP default key 0. The AP might be configured to use the WEP key in WEP default key 0 for WEP; if the AP is configured in this way, STAs that cannot support WEP default key 0 simultaneously with a TKIP pairwise key shall specify the No Pairwise subfield in the RSN Capabilities field. If an AP is configured to use WEP default key 0 as a WEP key and a “No Pairwise” STA associates, the AP shall not set the Install bit in the 4-Way Handshake. In other words, the STA does not install a pairwise temporal key and instead uses WEP default key 0 for all traffic.

NOTE—The behavior of “No Pairwise” STAs is only intended to support the migration of WEP to RSNA.

TKIP STAs in a mixed environment are expected to support a single pairwise key either by using a key mapping key or by mapping to default key 0. The AP uses a pairwise key for individually addressed traffic between the AP and the STA. If a key mapping key is available, the <RA,TA> pair identifies the key; if there is no key mapping key, then the default key 0 is used because the key index in the message is 0.

A STA that cannot support TKIP keys and WEP default key 0 simultaneously advertises this deficiency by setting the No Pairwise subfield in the RSNE it sends in the (Re)Association Request to the AP. In response, the AP sets the Install bit to 0 in Message 3 of the 4-Way Handshake to notify the STA not to install the pairwise key. The AP instead sends the WEP shared key to the STA to be plumbed as the WEP default key 0; this key is then used with WEP to send and receive individually addressed traffic between the AP and the STA.

The TKIP STA that has this limitation might not know that it will be forced to use WEP for all transmissions until it has associated with the AP and been given the keys to use. (The STA cannot know that the AP has been configured to use WEP default key 0 for WEP communication.) If this does not satisfy the security policy configured at the STA, the STA’s only recourse is to disassociate and try a different AP.

STAs using enhanced data cryptographic encapsulation mechanisms in a TSN shall support pairwise keys and WEP default key 0 simultaneously. It is invalid for the STA to negotiate the No Pairwise subfield when an enhanced data cryptographic encapsulation mechanism other than TKIP is one of the configured ciphers.

11.6.1.2 PRF

A PRF is used in a number of places in this standard. Depending on its use, it may need to output 128 bits, 192 bits, 256 bits, 384 bits, or 512 bits. This subclause defines five functions:

- PRF-128, which outputs 128 bits
- PRF-192, which outputs 192 bits
- PRF-256, which outputs 256 bits
- PRF-384, which outputs 384 bits
- PRF-512, which outputs 512 bits

In the following, *K* is a key; *A* is a unique label for each different purpose of the PRF; *B* is a variable-length string; *Y* is a single octet containing 0; *X* is a single octet containing the loop parameter *i*; and || denotes concatenation:

$$\text{H-SHA-1}(K, A, B, X) \leftarrow \text{HMAC-SHA-1}(K, A \,\|\, Y \,\|\, B \,\|\, X)$$

```
PRF(K, A, B, Len)
    for i ← 0 to (Len+159)/160 do
        R ← R || H-SHA-1(K, A, B, i)
    return L(R, 0, Len)
```

$\text{PRF-128}(K, A, B) = \text{PRF}(K, A, B, 128)$
$\text{PRF-192}(K, A, B) = \text{PRF}(K, A, B, 192)$
$\text{PRF-256}(K, A, B) = \text{PRF}(K, A, B, 256)$
$\text{PRF-384}(K, A, B) = \text{PRF}(K, A, B, 384)$
$\text{PRF-512}(K, A, B) = \text{PRF}(K, A, B, 512)$

When the negotiated AKM is 00-0F-AC:5 or 00-0F-AC:6, the KDF specified in 11.6.1.7.2 shall be used instead of the PRF construction defined here. In this case, A is used as the KDF label and B as the KDF Context and the PRF functions are defined as follows:

PRF-128(K, A, B) = KDF-128(K, A, B)
PRF-192(K, A, B) = KDF-192(K, A, B)
PRF-256(K, A, B) = KDF-256(K, A, B)

PRF-384(K, A, B) = KDF-384(K, A, B)
PRF-512(K, A, B) = KDF-512(K, A, B

11.6.1.3 Pairwise key hierarchy

Except when preauthentication is used, the pairwise key hierarchy utilizes PRF-384 or PRF-512 to derive session-specific keys from a PMK, as depicted in Figure 11-24. The PMK shall be 256 bits. The pairwise key hierarchy takes a PMK and generates a PTK. The PTK is partitioned into KCK, KEK, and temporal keys, which are used by the MAC to protect individually addressed communication between the Authenticator's and Supplicant's respective STAs. PTKs are used between a single Supplicant and a single Authenticator.

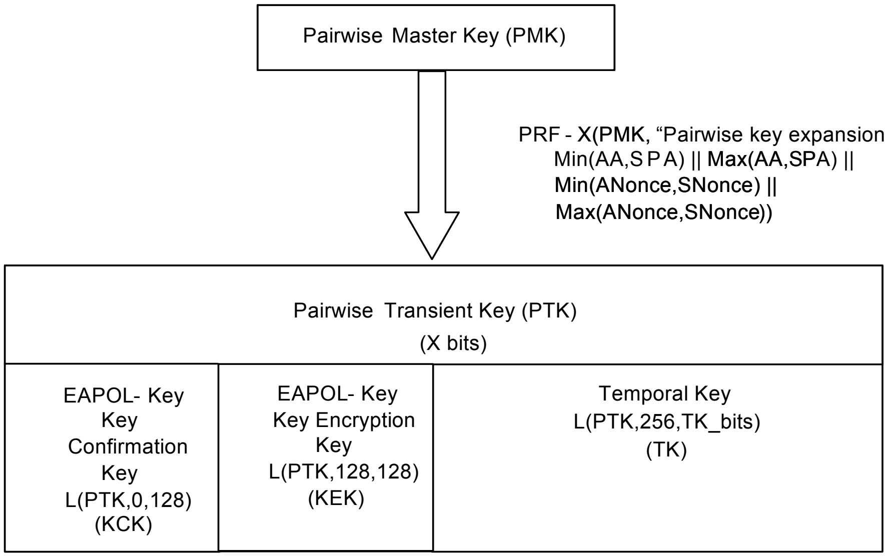

Pairwise Master Key (PMK)
PRF - X(PMK, "Pairwise key expansion", Min(AA,S P A) || Max(AA,SPA) || Min(ANonce,SNonce) || Max(ANonce,SNonce))
Pairwise Transient Key (PTK) (X bits)
EAPOL- Key Key Confirmation Key L(PTK,0,128) (KCK)
EAPOL- Key Key Encryption Key L(PTK,128,128) (KEK)
Temporal Key L(PTK,256,TK_bits) (TK)

Figure 11-24—Pairwise key hierarchy

When not using a PSK, the PMK is derived from the MSK. The PMK shall be computed as the first 256 bits (bits 0–255) of the MSK: PMK ← L(MSK, 0, 256). When this derivation is used, the MSK needs to consist of at least 256 bits.

The PTK shall not be used longer than the PMK lifetime as determined by the minimum of the PMK lifetime indicated by the AS, e.g., Session-Timeout + dot1xAuthTxPeriod or from dot11RSNAConfigPMK-Lifetime. When RADIUS is used and the Session-Timeout attribute is not in the RADIUS Accept message, and if the key lifetime is not otherwise specified, then the PMK lifetime is infinite.

NOTE 1—If the protocol between the Authenticator (or AP) and AS is RADIUS, then the MS-MPPE-Recv-Key attribute (vendor-id = 17; see Section 2.4.3 in IETF RFC 2548-1999 [B30]) is available to be used to transport the PMK to the AP.

NOTE 2—When reauthenticating and changing the pairwise key, a race condition might occur. If a frame is received while MLME-SETKEYS.request primitive is being processed, the received frame might be decrypted with one key and the MIC checked with a different key. Two possible options to avoid this race condition are as follows: the frame might be checked against the old MIC key, and the received frames might be queued while the keys are changed.

NOTE 3—If the AKMP is RSNA-PSK, then a 256-bit PSK might be configured into the STA and AP or a pass-phrase might be configured into the Supplicant or Authenticator. The method used to configure the PSK is outside this standard, but one method is via user interaction. If a pass-phrase is configured, then a 256-bit key is derived and used as the PSK. In any RSNA-PSK method, the PSK is used directly as the PMK. Implementations might support different PSKs for each pair of communicating STAs.

Here, the following assumptions apply:

- SNonce is a random or pseudorandom value contributed by the Supplicant; its value is taken when a PTK is instantiated and is sent to the PTK Authenticator.
- ANonce is a random or pseudorandom value contributed by the Authenticator.
- The PTK shall be derived from the PMK by

  PTK ← PRF-X(PMK, "Pairwise key expansion", Min(AA,SPA) || Max(AA,SPA) ||
  Min(ANonce,SNonce) || Max(ANonce,SNonce))

  where X = 256 + TK_bits. The value of TK_bits is cipher-suite dependent and is defined in Table 11-4. The Min and Max operations for IEEE 802 addresses are with the address converted to a positive integer treating the first transmitted octet as the most significant octet of the integer. The Min and Max operations for nonces are with the nonces treated as positive integers converted as specified in 8.2.2.

  NOTE—The Authenticator and Supplicant normally derive a PTK only once per association. A Supplicant or an Authenticator use the 4-Way Handshake to derive a new PTK. Both the Authenticator and Supplicant create a new nonce value for each 4-Way Handshake instance.
- The KCK shall be computed as the first 128 bits (bits 0–127) of the PTK:

  KCK ← L(PTK, 0, 128)

  The KCK is used by IEEE Std 802.1X-2004 to provided data origin authenticity in the 4-Way Handshake and Group Key Handshake messages.
- The KEK shall be computed as bits 128–255 of the PTK:

  KEK ← L(PTK, 128, 128)

  The KEK is used by the EAPOL-Key frames to provide data confidentiality in the 4-Way Handshake and Group Key Handshake messages.
- The temporal key (TK) shall be computed as bits 256 to (255 + TK_bits) of the PTK:

  TK ← L(PTK, 256, TK_bits)

The EAPOL-Key state machines (see 11.6.10 and 11.6.11) use the MLME-SETKEYS.request primitive to configure the temporal key into the STA. The STA uses the temporal key with the pairwise cipher suite; interpretation of this value is cipher-suite-specific.

A PMK identifier is defined as

PMKID = HMAC-SHA1-128(PMK, "PMK Name" || AA || SPA)

Here, HMAC-SHA1-128 is the first 128 bits of the HMAC-SHA1 of its argument list.

When the negotiated AKM is 00-0F-AC:5 or 00-0F-AC:6, HMAC-SHA-256 is used to calculate the PMKID, and the PMK identifier is defined as

PMKID = Truncate-128(HMAC-SHA-256(PMK, "PMK Name" || AA || SPA))

NOTE—When the PMKID is calculated for the PMKSA as part of RSN preauthentication, the AKM has not yet been negotiated. In this case, the HMAC-SHA1-128 based derivation is used for the PMKID calculation.

11.6.1.4 Group key hierarchy

The group temporal key (GTK) shall be a random number. The following is an example method for deriving a random GTK. Any other pseudorandom function, such as that specified in 11.6.1.2, could also be used.

A group master key (GMK) is an auxiliary key that may be used to derive a GTK at a time interval configured into the AP to reduce the exposure of data if the GMK is ever compromised.

The Authenticator might update the GTK for a number of reasons:

a) The Authenticator might change the GTK on disassociation or deauthentication of a STA.
b) An event within the SME might trigger a Group Key Handshake.

Figure 11-25 depicts an example of a relationship among the keys of the group key hierarchy. In this model, the group key hierarchy takes a GMK and generates a GTK. The GTK is partitioned into temporal keys used by the MAC to protect group addressed communication. GTKs are used between a single Authenticator and all Supplicants authenticated to that Authenticator. The Authenticator derives new GTKs when it needs to update the GTKs.

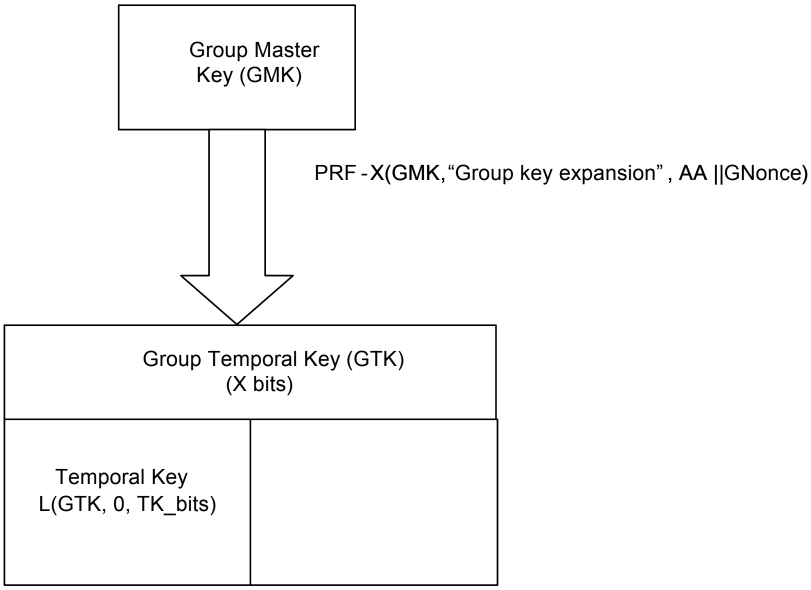


Figure 11-25—Group key hierarchy (informative)

In this example, the following assumptions apply:

a) Group nonce (GNonce) is a random or pseudorandom value contributed by the IEEE 802.1X Authenticator.
b) The GTK is derived from the GMK by
c) GTK ← PRF-X(GMK, "Group key expansion" || AA || GNonce)
d) X = 256 + TK_bits. The value of TK_bits is cipher-suite dependent and is defined in Table 11-4. AA is represented as an IEEE 802 address and GNonce as a bit string as defined in 8.2.2.
e) The temporal key (TK) is bits 0 to (TK_bits – 1) of the GTK:
TK ← L(GTK, 0, TK_bits)
f) The EAPOL-Key state machines (see 11.6.10 and 11.6.11) configure the temporal key into IEEE Std 802.11 via the MLME-SETKEYS.request primitive, and IEEE Std 802.11 uses this key. Its interpretation is cipher-suite-specific.

11.6.1.5 Integrity group key hierarchy

The Authenticator shall select the IGTK as a random value each time it is generated.

The Authenticator may update the IGTK for any reason, including:

a) The disassociation or deauthentication of a STA.
b) An event within the SME that triggers a Group Key Handshake.

The IGTK is configured via the MLME-SETKEYS.request primitive; see 6.3.19. IGTK configuration is described in the EAPOL-Key state machines; see 11.6.10 and 11.6.11.

The IPN is used to provide replay protection.

11.6.1.6 PeerKey key hierarchy

The station-to-station key hierarchy utilizes PRF-384 or PRF-512 to derive session-specific keys from an SMK, as depicted in Figure 11-26. The SMK shall be 256 bits. The pairwise key hierarchy takes an SMK and generates an STK. The STK is partitioned into SKCK, SKEK, and temporal keys, which are used by the MAC to protect individually addressed communication between the initiator and peer STAs. STKs are used between a single initiator STA and a single peer STA.

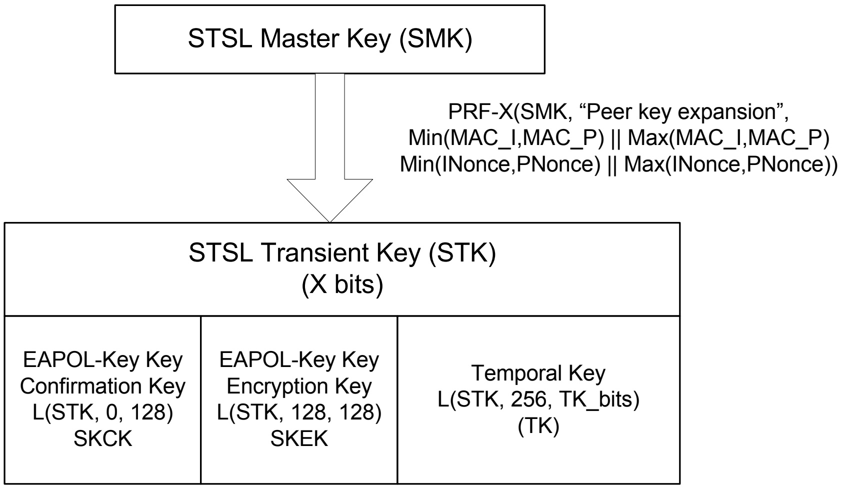

STSL Master Key (SMK)
PRF-X(SMK, “Peer key expansion”,
Min(MAC_I,MAC_P) || Max(MAC_I,MAC_P)
Min(INonce,PNonce) || Max(INonce,PNonce))
STSL Transient Key (STK)
(X bits)
EAPOL-Key Key Confirmation Key L(STK, 0, 128) SKCK
EAPOL-Key Key Encryption Key L(STK, 128, 128) SKEK
Temporal Key L(STK, 256, TK_bits) (TK)

Figure 11-26—PeerKey hierarchy

The following apply and are depicted in Figure 11-26:

a) INonce is a random or pseudorandom value contributed by the initiator STA.

b) PNonce is a random or pseudorandom value contributed by the peer STA.

c) The STK shall be derived from the SMK by

   STK ← PRF-X(SMK, "Peer key expansion", Min(MAC_I,MAC_P) || Max(MAC_I,MAC_P) || Min(INonce,PNonce) || Max(INonce,PNonce))

   where X = 256 + TK_bits. The value of TK_bits is cipher-suite dependent and is defined in Table 11-4. The Min and Max operations for IEEE 802 addresses are with the address converted to a positive integer treating the first transmitted octet as the most significant octet of the integer. The Min and Max operations for nonces are with the nonces treated as positive integers converted as specified in 8.2.2.

d) The SKCK shall be computed as the first 128 bits (bits 0–127) of the STK:

   SKCK ← L(STK, 0, 128)

   The SKCK is used to provide data origin authenticity in the 4-Way STK Handshake.

e) The SKEK shall be computed as bits 128–255 of the STK:

SKEK ← L(STK, 128, 128)

The SKEK is used by the EAPOL-Key frames to provide confidentiality in the 4-Way STK Handshake.

f) The temporal key (TK) shall be computed as bits 256 to (255 + TK_bits) of the STK:

TK ← L(STK, 256, TK_bits)

The EAPOL-Key state machines (see 11.6.10 and 11.6.11) use the MLME-SETKEYS.request primitive to configure the temporal key into the STA. The STA uses the temporal key with the pairwise cipher suite; interpretation of this value is cipher-suite-specific.

A SMK identifier is defined as

SMKID = HMAC-SHA1-128(SMK, "SMK Name" || PNonce || MAC_P || INonce || MAC_I)

Here, HMAC-SHA1-128 is the first 128 bits of the HMAC-SHA1 of its argument list.

When the negotiated AKM is 00-0F-AC:5 or 00-0F-AC:6, HMAC-SHA-256 is used to calculate the SMKID, and an SMK identifier is defined as

SMKID = Truncate-128(HMAC-SHA-256(SMK, "SMK Name" || PNonce || MAC_P || INonce || MAC_I))

11.6.1.7 FT key hierarchy

11.6.1.7.1 Overview

This subclause describes the FT key hierarchy and its supporting architecture. The FT key hierarchy is designed to allow a STA to make fast BSS transitions between APs without the need to perform an SAE or IEEE 802.1X authentication at every AP within the mobility domain.

The FT key hierarchy can be used with SAE, IEEE 802.1X authentication, or PSK authentication.

A three-level key hierarchy provides key separation between the key holders. The FT key hierarchy for the Authenticator is shown in Figure 11-27. An identical key hierarchy exists for the Supplicant, and identical functions are performed by the corresponding S0KH and S1KH.

The FT key hierarchy shown in Figure 11-27 consists of three levels whose keys are derived using the key derivation function (KDF) described in 11.6.1.7.2 as follows:

a) PMK-R0 – the first-level key of the FT key hierarchy. This key is derived as a function of the master session key (MSK) or PSK. It is stored by the PMK-R0 key holders, R0KH and S0KH.
b) PMK-R1 – the second-level key of the FT key hierarchy. This key is mutually derived by the S0KH and R0KH.
c) PTK – the third-level key of the FT key hierarchy that defines the IEEE 802.11 and IEEE 802.1X protection keys. The PTK is mutually derived by the PMK-R1 key holders, R1KH and S1KH.

As shown in Figure 11-27, the R0KH computes the PMK-R0 from the key obtained from SAE authentication (for the purposes of FT this key is identified as the Master PMK, or MPMK), from the PSK, or from the MSK resulting (per IETF RFC 3748-2004 [B38]) from a successful IEEE 802.1X authentication between the AS and the Supplicant. Upon a successful authentication, the R0KH shall delete any prior PMK-R0 security association for this mobility domain pertaining to this S0KH. The R0KH shall also delete all PMK-R1 security associations derived from that prior PMK-R0 security association. The PMK-R1s are generated by the R0KH and are assumed to be delivered from the R0KH to the R1KHs within the same mobility domain. The PMK-R1s are used for PTK generation. Upon receiving a new PMK-R1 for an S0KH, an R1KH deletes the prior PMK-R1 security association and PTKSAs derived from the prior PMK-R1.

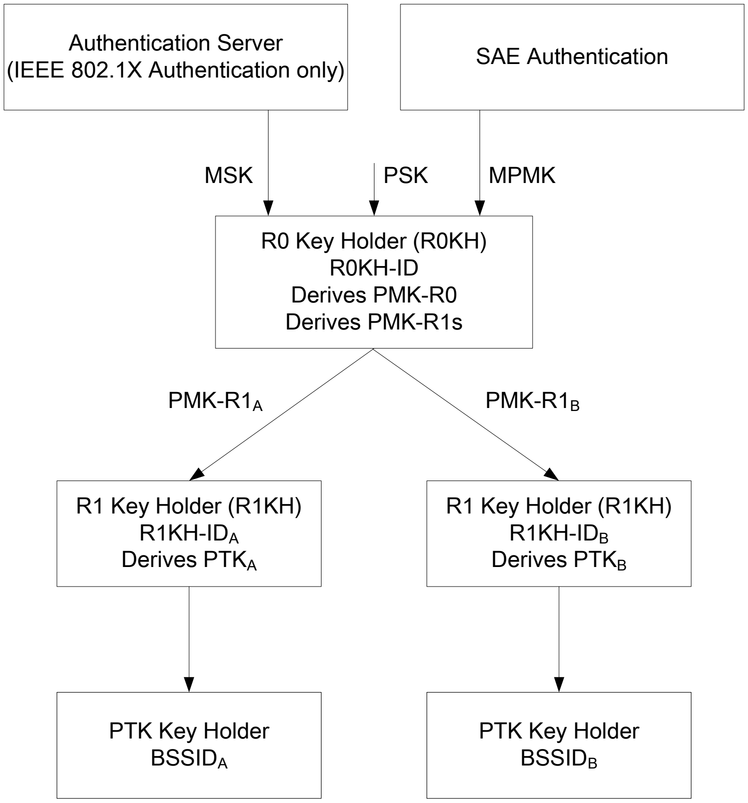


Figure 11-27—FT key hierarchy at an Authenticator

It is assumed by this standard that the PSK is specific to a single S0KH and a single R0KH.

The lifetime of the PMK-R0, PMK-R1, and PTK are bound to the lifetime of the MPMK, PSK, or MSK from which it was derived. For example, the AS may communicate the MSK lifetime with the MSK. If such an attribute is provided, the lifetime of the PMK-R0 shall be not more than the lifetime of the MSK. The lifetime of the PTK and PMK-R1 is the same as that of the PMK-R0. When the key lifetime expires, each key holder shall delete its respective PMK-R0, PMK-R1 or PTK security association.

The FT key hierarchy derives its keys using the KDF defined in 11.6.1.7.2 with separate labels to further distinguish derivations.

During a fast BSS transition, a STA shall negotiate the same pairwise cipher suite with target APs as was negotiated in the FT initial mobility domain association. Using the pairwise cipher suite selector value in the PMK-R1 security association received from the R0KH, the target AP shall verify that the same pairwise cipher suite selector is being used.

The distribution of keys from the R0KH to the R1KHs is outside the scope of this standard. It is assumed that the PMK-R1s are distributed from the R0KH to the R1KHs following the requirements specified in 12.2.2.

The PMK-R0 may be deleted by the R0KH after PMK-R1s have been derived. When the PMK-R0 is deleted, the R0KH needs only to maintain the PMK-R1 security associations.

11.6.1.7.2 Key derivation function (KDF)

The KDF for the FT key hierarchy is a variant of the pseudorandom function (PRF) defined in 11.6.1.2 and is defined as follows:

```
Output ← KDF-Length (K, label, Context) where
Input:  K, a 256-bit key derivation key
        label, a string identifying the purpose of the keys derived using this KDF
        Context, a bit string that provides context to identify the derived key
        Length, the length of the derived key in bits
Output: a Length-bit derived key

result ← ""
iterations ← (Length+255)/256
do i = 1 to iterations
        result ← result || HMAC-SHA256(K, i || label || Context || Length)
od
return first Length bits of result, and securely delete all unused bits
```

In this algorithm, *i* and *Length* are encoded as 16-bit unsigned integers, represented using the bit ordering conventions of 8.2.2. *K*, *label*, and *Context* are bit strings and are represented using the ordering conventions of 8.2.2.

11.6.1.7.3 PMK-R0

The first-level key in the FT key hierarchy, PMK-R0, is derived using the KDF defined in 11.6.1.7.2. The PMK-R0 is the first level 256-bit keying material used to derive the next level keys (PMK-R1s):

R0-Key-Data = KDF-384(XXKey, "FT-R0", SSIDlength || SSID || MDID || R0KHlength || R0KH-ID || S0KH-ID)

PMK-R0 = L(R0-Key-Data, 0, 256)

PMK-R0Name-Salt = L(R0-Key-Data, 256, 128)

where

- KDF-384 is the KDF as defined in 11.6.1.7.2 used to generate a key of length 384 bits.
- L(-) is defined in 11.6.1.
- If the AKM negotiated is 00-0F-AC:3, then XXKey shall be the second 256 bits of the MSK (which is derived from the IEEE 802.1X authentication), i.e., XXKey = L(MSK, 256, 256). If the AKM negotiated is 00-0F-AC:4, then XXKey shall be the PSK. If the AKM negotiated is 00-0F-AC:9, then XXKey shall be the MPMK generated as the result of SAE authentication.
- "FT-R0" is 0x46 0x54 0x2D 0x52 0x30.
- SSIDlength is a single octet whose value is the number of octets in the SSID.
- SSID is the service set identifier, a variable-length sequence of octets, as it appears in the Beacon and Probe Response frames.
- MDID is the Mobility Domain Identifier field from the MDE that was used during FT initial mobility domain association.
- R0KHlength is a single octet whose value is the number of octets in the R0KH-ID.

- R0KH-ID is the identifier of the holder of PMK-R0 in the Authenticator.
- S0KH-ID is the Supplicant's MAC address (SPA).

PMK-R0 shall be computed as the first 256 bits (bits 0–255) of the R0-Key-Data. The latter 128 bits of R0-Key-Data shall be used as the PMK-R0Name-Salt to generate the PMKR0Name.

The PMK-R0 is referenced and named as follows:
PMKR0Name = Truncate-128(SHA-256("FT-R0N" || PMK-R0Name-Salt))

where

- "FT-R0N" is 0x46 0x54 0x2D 0x52 0x30 0x4E.
- Truncate-128(-) returns the first 128 bits of its argument and securely destroys the remainder.
- SHA-256 is as defined in FIPS PUB 180-3-2008.

The PMKR0Name is used to identify the PMK-R0.

11.6.1.7.4 PMK-R1

The second-level key in the FT key hierarchy, PMK-R1, is a 256-bit key used to derive the PTK. The PMK-R1 is derived using the KDF defined in 11.6.1.7.2:
PMK-R1 = KDF-256(PMK-R0, "FT-R1", R1KH-ID || S1KH-ID)

where

- KDF-256 is the KDF as defined in 11.6.1.7.2 used to generate a key of length 256 bits.
- PMK-R0 is the first level key in the FT key hierarchy.
- "FT-R1" is 0x46 0x54 0x2D 0x52 0x31.
- R1KH-ID is a MAC address of the holder of the PMK-R1 in the Authenticator of the AP.
- S1KH-ID is the SPA.

The PMK-R1 is referenced and named as follows:
PMKR1Name = Truncate-128(SHA-256("FT-R1N" || PMKR0Name || R1KH-ID || S1KH-ID))

where

- "FT-R1N" is 0x46 0x54 0x2D 0x52 0x31 0x4E.

PMKR1Name is used to identify the PMK-R1.

11.6.1.7.5 PTK

The third-level key in the FT key hierarchy is the PTK. This key is mutually derived by the S1KH and the R1KH used by the target AP, with the key length being a function of the negotiated cipher suite as defined by Table 11-4 in 11.6.2.

Using the KDF defined in 11.6.1.7.2, the PTK derivation is as follows:
PTK = KDF-PTKLen(PMK-R1, "FT-PTK", SNonce || ANonce || BSSID || STA-ADDR)

where

- KDF-PTKLen is the KDF as defined in 11.6.1.7.2 used to generate a PTK of length PTKLen.
- PMK-R1 is the key that is shared between the S1KH and the R1KH.
- "FT-PTK" is 0x46 0x54 0x2D 0x50 0x54 0x4B.

- SNonce is a 256-bit random bit string contributed by the S1KH.
- ANonce is a 256-bit random bit string contributed by the R1KH.
- STA-ADDR is the non-AP STA's MAC address.
- BSSID is the BSSID of the target AP.
- PTKlen is the total number of bits to derive, i.e., number of bits of the PTK. The length is dependent on the negotiated cipher suites as defined by Table 11-4 in 11.6.2.

Each PTK has three component keys, KCK, KEK, and a temporal key, derived as follows:

The KCK shall be computed as the first 128 bits (bits 0–127) of the PTK:
KCK = L(PTK, 0, 128)

where L(-) is defined in 11.6.1.

The KCK is used to provide data origin authenticity in EAPOL-Key messages, as defined in 11.6.2, and in the FT authentication sequence, as defined in 12.8.

The KEK shall be computed as bits 128–255 of the PTK:
KEK = L(PTK, 128, 128)

The KEK is used to provide data confidentiality for certain fields (KeyData) in EAPOL-Key messages, as defined in 11.6.2, and in the FT authentication sequence, as defined in 12.8.

The temporal key (TK) shall be computed as bits 256–383 (for CCMP) of the PTK:
TK = L(PTK, 256, 128)

For vendor-specific cipher suites, the length of the temporal key (and the value of PTKLen) depend on the vendor-specific algorithm.

The temporal key is configured into the STA by the SME through the use of the MLME-SETKEYS.request primitive. The STA uses the temporal key with the pairwise cipher suite; interpretation of this value is specific to the cipher suite.

The PTK is referenced and named as follows:
PTKName = Truncate-128(SHA-256(PMKR1Name || "FT-PTKN" || SNonce || ANonce || BSSID || STA-ADDR))

where

- "FT-PTKN" is 0x46 0x54 0x2D 0x50 0x54 0x4B 0x4E.

The PTKName is used to identify the PTK key.

11.6.2 EAPOL-Key frames

IEEE Std 802.11 uses EAPOL-Key frames to exchange information between STAs' Supplicants and Authenticators. These exchanges result in cryptographic keys and synchronization of security association state. EAPOL-Key frames are used to implement three different exchanges:

- 4-Way Handshake, to confirm that the PMK between associated STAs is the same and live and to transfer the GTK to the STA.
- Group Key Handshake, to update the GTK at the STA.
- PeerKey initial SMK Handshake to deliver the SMK and final 4-Way STK Handshake to deliver the STK to the initiating and peer STAs.

When priority processing of data frames is supported, an SME should send EAPOL-Key frames at the highest priority.

The RSNA key descriptor used by IEEE Std 802.11 does not use the IEEE 802.1X key descriptor. Instead, it uses the key descriptor described in this subclause.

The bit and octet convention for fields in the EAPOL-Key frame are defined in 7.1 of IEEE Std 802.1X-2004. EAPOL-Key frames containing invalid field values shall be silently discarded. Figure 11-28 depicts the format of an EAPOL-Key frame.

<table>
<tr><td>Protocol Version – 1 octet</td><td>Packet Type – 1 octet</td><td>Packet Body Length – 2 octets</td></tr>
<tr><td colspan="3">Descriptor Type – 1 octet</td></tr>
<tr><td colspan="2">Key Information – 2 octets</td><td>Key Length – 2 octets</td></tr>
<tr><td colspan="3">Key Replay Counter – 8 octets</td></tr>
<tr><td colspan="3">Key Nonce – 32 octets</td></tr>
<tr><td colspan="3">EAPOL - Key IV – 16 octets</td></tr>
<tr><td colspan="3">Key RSC – 8 octets</td></tr>
<tr><td colspan="3">Reserved - 8 octets</td></tr>
<tr><td colspan="3">Key MIC – variable</td></tr>
<tr><td colspan="2">Key Data Length – 2 octets</td><td>Key Data – n octets</td></tr>
</table>

Figure 11-28—EAPOL-Key frame

The fields of a EAPOL-Key frame body are as follows:

a) **Descriptor Type.** This field is 1 octet and has a value defined by IEEE Std 802.1X-2004, identifying the IEEE 802.11 key descriptor.

b) **Key Information**. This field is 2 octets and specifies characteristics of the key. See Figure 11-29.

B0 B2	B3	B4 B5	B6	B7	B8	B9	B10	B11	B12	B13	B14 B15
Key Descriptor Version	Key Type	Reserved	Install	Key Ack	Key MIC	Se-cure	Error	Request	Encrypted Key Data	SMK Message	Reserved

Figure 11-29—Key Information bit layout

The bit convention used is as in 7.1 of IEEE Std 802.1X-2004. The subfields of the Key Information field are as follows:

1) Key Descriptor Version (bits 0–2) shall be set to 0 on all transmitted EAPOL-Key frames except under the following circumstances:

   i) The value 1 shall be used for all EAPOL-Key frames to a STA when the negotiated AKM is 00-0F-AC:1 or 00-0F-AC:2 and the pairwise cipher is TKIP or "Use Group Cipher" for Key Descriptor 1. This value indicates the following:

      — HMAC-MD5 is the EAPOL-Key MIC.

      — ARC4 is the EAPOL-Key encryption algorithm used to protect the Key Data field.

      — The MIC is 16 octets.

ii) The value 2 shall be used for all EAPOL-Key frames to a STA when the negotiated AKM is 00-0F-AC:1 or 00-0F-AC:2 and either the pairwise or the group cipher is an enhanced data cryptographic encapsulation mechanism other than TKIP for Key Descriptor 2. This value indicates the following:

— HMAC-SHA1-128 is the EAPOL-Key MIC. HMAC is defined in IETF RFC 2104; and SHA1, by FIPS PUB 180-3-2008. The output of the HMAC-SHA1 shall be truncated to its 128 MSBs (octets 0–15 of the digest output by HMAC-SHA1), i.e., the last four octets generated shall be discarded.

— The NIST AES key wrap is the EAPOL-Key encryption algorithm used to protect the Key Data field. IETF RFC 3394 defines the NIST AES key wrap algorithm.

— The MIC is 16 octets.

iii) The value 3 shall be used for all EAPOL-Key frames to a STA when the negotiated AKM is 00-0F-AC:3, 00-0F-AC:4, 00-0F-AC:5, or 00-0F-AC:6. This value indicates the following:

— AES-128-CMAC is the EAPOL-Key MIC. AES-128-CMAC is defined by FIPS SP800-38B and also found in IETF RFC 4493. The output of the AES-128-CMAC shall be 128 bits.

— The NIST AES key wrap is the EAPOL-Key encryption algorithm used to protect the Key Data field. IETF RFC 3394 defines the NIST AES key wrap algorithm.

— The MIC is 16 octets.

2) Key Type (bit 3) specifies whether this EAPOL-Key frame is part of a 4-Way Handshake deriving a PTK.

i) The value 0 (Group/SMK) indicates the message is not part of a PTK derivation.

ii) The value 1 (Pairwise) indicates the message is part of a PTK derivation.

3) Reserved (bits 4–5). The sender shall set them to 0, and the receiver shall ignore the value of these bits.

4) Install (bit 6).

i) If the value of Key Type (bit 3) is 1, then for the Install bit,

— The value 1 means the IEEE 802.1X component shall configure the temporal key derived from this message into its IEEE 802.11 STA.

— The value 0 means the IEEE 802.1X component shall not configure the temporal key into the IEEE 802.11 STA.

ii) If the value of Key Type (bit 3) is 0, then this bit shall be 0 on transmit and ignored on receive.

5) Key Ack (bit 7) is set to 1 in messages from the Authenticator if an EAPOL-Key frame is required in response to this message and is 0 otherwise. The Supplicant's response to this message shall use the same replay counter as this message.

6) Key MIC (bit 8) is set to 1 if a MIC is in this EAPOL-Key frame and is set to 0 if this message contains no MIC.

7) Secure (bit 9) is set to 1 once the initial key exchange is complete.

The Authenticator shall set the Secure bit to 0 in all EAPOL-Key frames sent before the Supplicant has the PTK and the GTK. The Authenticator shall set the Secure bit to 1 in all EAPOL-Key frames it sends to the Supplicant containing the last key needed to complete the Supplicant's initialization.

The Supplicant shall set the Secure bit to 0 in all EAPOL-Key frames it sends before it has the PTK and the GTK and before it has received an EAPOL-Key frame from the Authenticator with the Secure bit equal to 1 (this should be before receiving Message 3 of the 4-Way

Handshake). The Supplicant shall set the Secure bit to 1 in all EAPOL-Key Frames sent after this until it loses the security association it shares with the Authenticator.

8) Error (bit 10) is set by a Supplicant to report that a MIC failure occurred in a TKIP MSDU or SMK Handshake failure. In case of a MIC failure, a Supplicant shall set this bit to 1 only when the Request (bit 11) is 1. When the SMK Message bit is 1, Error shall be set to 1 to indicate the key data field contains an Error KDE.

9) Request (bit 11) is set to 1 by a Supplicant to request that the Authenticator initiate either a 4-Way Handshake or Group Key Handshake, is set to 1 by a Supplicant in a Michael MIC Failure Report, and is set to 1 by the STSL peer STA to request initiator STA rekeying of the STK. The Supplicant shall not set this bit to 1 in on-going 4-Way Handshakes, i.e., the Key Ack bit (bit 7) shall not be set to 1 in any message in which the Request bit is 1. The Authenticator shall never set this bit to 1.

   In a Michael MIC Failure Report, setting the bit is not a request to initiate a new handshake. However, the recipient may initiate a new handshake on receiving such a message.

   If an EAPOL-Key frame in which the Request bit is 1 has a key type of Pairwise, the Authenticator shall initiate a 4-Way Handshake. If the EAPOL-Key frame in which the Request bit is 1 has a key type of Group/SMK, the Authenticator shall change the GTK, initiate a 4-Way Handshake with the Supplicant, and then execute the Group Key Handshake to all Supplicants.

   If an EAPOL-Key frame in which the Request bit is 1 has the SMK Message bit equal to 1, the initiator STA shall take appropriate action to create a new STK (based on 11.6.8).

10) Encrypted Key Data (bit 12) is set to 1 if the Key Data field is encrypted and is set to 0 if the Key Data field is not encrypted. This subfield shall be set to 1, and the Key Data field shall be encrypted, if any key material (e.g., GTK or SMK) is included in the frame.

11) SMK Message (bit 13) specifies whether this EAPOL-Key frame is part of an SMK Handshake. If the SMK Handshake is not supported, the STA shall set the SMK message bit to 0 and shall ignore the value of this bit upon receipt.

12) Reserved (bits 14–15). The sender shall set them to 0, and the receiver shall ignore the value of these bits.

c) **Key Length**. This field is 2 octets in length, represented as an unsigned binary number. The value defines the length in octets of the pairwise temporal key to configure into IEEE Std 802.11. See Table 11-4.

Table 11-4—Cipher suite key lengths

Cipher suite	Key length (octets)	TK_bits (bits)
WEP-40	5	40
WEP-104	13	104
TKIP	32	256
CCMP	16	128
BIP	16	128

d) **Key Replay Counter**. This field is 8 octets, represented as an unsigned binary number, and is initialized to 0 when the PMK is established. The Supplicant shall use the key replay counter in the received EAPOL-Key frame when responding to an EAPOL-Key frame. It carries a sequence number that the protocol uses to detect replayed EAPOL-Key frames.

The Supplicant and Authenticator shall track the key replay counter per security association. The key replay counter shall be initialized to 0 on (re)association. The Authenticator shall increment the key replay counter on each successive EAPOL-Key frame.

When replying to a message from the Authenticator, the Supplicant shall use the Key Replay Counter field value from the last valid EAPOL-Key frames received from the Authenticator. The Authenticator should use the key replay counter to identify invalid messages to silently discard. The Supplicant should also use the key replay counter and ignore EAPOL-Key frames with a Key Replay Counter field value smaller than or equal to any received in a valid message. The local Key Replay Counter field should not be updated until after the EAPOL-Key MIC is checked and is found to be valid. In other words, the Supplicant never updates the Key Replay Counter field for Message 1 in the 4-Way Handshake, as it includes no MIC. This implies the Supplicant needs to allow for retransmission of Message 1 when checking for the key replay counter of Message 3.

The Supplicant shall maintain a separate key replay counter for sending EAPOL-Key request frames to the Authenticator; the Authenticator also shall enforce monotonicity of a separate replay counter to filter received EAPOL-Key Request frames.

NOTE—The key replay counter does not play any role beyond a performance optimization in the 4-Way Handshake. In particular, replay protection is provided by selecting a never-before-used nonce value to incorporate into the PTK. It does, however, play a useful role in the Group Key Handshake.

e) **Key Nonce**. This field is 32 octets. It conveys the ANonce from the Authenticator and the SNonce from the Supplicant. It may contain 0 if a nonce is not required to be sent.

f) **EAPOL-Key IV**. This field is 16 octets. It contains the IV used with the KEK. It shall contain 0 when an IV is not required. It should be initialized by taking the current value of the global key counter (see 11.6.11) and then incrementing the counter. Note that only the lower 16 octets of the counter value are used.

g) **Key RSC**. This field is 8 octets in length. It contains the receive sequence counter (RSC) for the GTK being installed in IEEE Std 802.11. It is used in Message 3 of the 4-Way Handshake and Message 1 of the Group Key Handshake, where it is used to synchronize the IEEE 802.11 replay state. It may also be used in the Michael MIC Failure Report frame, to report the TSC field value of the frame experiencing a MIC failure. It shall contain 0 in other messages. The Key RSC field gives the current message number for the GTK, to allow a STA to identify replayed MPDUs. If the Key RSC field value is less than 8 octets in length, the remaining octets shall be set to 0. The least significant octet of the TSC or PN should be in the first octet of the Key RSC field. The encoding of the Key RSC field is defined in Table 11-5.

Table 11-5—Key RSC field

KeyRSC 0	KeyRSC 1	KeyRSC 2	KeyRSC 3	KeyRSC 4	KeyRSC 5	KeyRSC 6	KeyRSC 7
TSC0	TSC1	TSC2	TSC3	TSC4	TSC5	0	0
PN0	PN1	PN2	PN3	PN4	PN5	0	0

For WEP, the Key RSC value shall be set to 0 on transmit and shall not be used at the receiver.

h) **Key MIC**. The EAPOL Key MIC is a MIC of the EAPOL-Key frames, from and including the EAPOL protocol version field to and including the Key Data field, calculated with the Key MIC field set to 0. If the Encrypted Key Data subfield (of the Key Information field) is 1, the Key Data field is encrypted prior to computing the MIC. The length of this field depends on the negotiated AKM as defined in 11.6.3.

i) **Key Data Length**. This field is 2 octets in length, taken to represent an unsigned binary number. This represents the length of the Key Data field in octets. If the Encrypted Key Data subfield (of the Key Information field) is 1, the length is the length of the Key Data field after encryption, including any padding.

j) **Key Data**. This field is a variable-length field that is used to include any additional data required for the key exchange that is not included in the fields of the EAPOL-Key frame. The additional data may be zero or more element(s) (such as the RSNE) and zero or more key data cryptographic encapsulation(s) (KDEs) (such as GTK(s) or PMKID(s)). Elements sent in the Key Data field include the Element ID and Length subfields. KDEs shall be encapsulated using the format in Figure 11-30.

	Type (0xdd)	Length	OUI	Data Type	Data
Octets:	1	1	3	1	(Length – 4)

Figure 11-30—KDE format

The Type field shall be set to 0xdd. The Length field specifies the number of octets in the OUI, Data Type, and Data fields. The order of the OUI field is described in 8.2.2.

Table 11-6 lists the KDE selectors defined by this standard.

Table 11-6—KDE

OUI	Data type	Meaning
00-0F-AC	0	Reserved
00-0F-AC	1	GTK KDE
00-0F-AC	2	Reserved
00-0F-AC	3	MAC address KDE
00-0F-AC	4	PMKID KDE
00-0F-AC	5	SMK KDE
00-0F-AC	6	Nonce KDE
00-0F-AC	7	Lifetime KDE
00-0F-AC	8	Error KDE
00-0F-AC	9	IGTK KDE
00-AF-AC	10	Key ID KDE
00-0F-AC	11–255	Reserved
Vendor OUI	Any	Vendor specific
Other	Any	Reserved

STAs shall ignore any elements and KDEs they do not understand.

If the Encrypted Key Data subfield (of the Key Information field) is 1, the entire Key Data field shall be encrypted. If the Key Data field uses the NIST AES key wrap, then the Key Data field shall be padded before encrypting if the key data length is less than 16 octets or if it is not a multiple of 8. The padding consists of appending a single octet 0xdd followed by zero or more 0x00 octets. When processing a received EAPOL-Key message, the receiver shall ignore this trailing padding. Key Data fields that are encrypted, but do not contain the GroupKey or SMK KDE, shall be accepted.

If the GroupKey or SMK KDE is included in the Key Data field, but the Key Data field is not encrypted, the EAPOL-Key frames shall be ignored.

The format of the GTK KDE is shown in Figure 11-31.

KeyID (0,1,2, or 3)	Tx	Reserved (0)	Reserved (0)	GTK
bits 0–1	bit 2	bit 3–7	1 octet	(Length – 6) octets

Figure 11-31—GTK KDE format

If the value of the Tx field is 1, then the IEEE 802.1X component shall configure the temporal key derived from this KDE into its IEEE 802.11 STA for both transmission and reception.

If the value of the Tx field is 0, then the IEEE 802.1X component shall configure the temporal key derived from this KDE into its IEEE 802.11 STA for reception only.

The format of the MAC address KDE is shown in Figure 11-32.

	MAC Address
Octets:	6

Figure 11-32—MAC address KDE format

The format of the PMKID KDE is shown in Figure 11-33.

	PMKID
Octets:	16

Figure 11-33—PMKID KDE format

The format of the SMK KDE is shown in Figure 11-34.

	SMK	Key Nonce
Octets:	32	32

Figure 11-34—SMK KDE format

The format of the Nonce KDE is shown in Figure 11-35.

	Key Nonce
Octets:	32

Figure 11-35—Nonce KDE format

The format of the Lifetime KDE is shown in Figure 11-36. The Key Lifetime value is expressed in seconds and uses big endian byte order.

	Key Lifetime (in seconds)
Octets:	4

Figure 11-36—Lifetime KDE format

The format of the Error KDE is shown in Figure 11-37. Both MUI and Error Type fields are in big endian byte order. Table 11-7 shows different values of MUI, and Table 11-8 shows different values of SMK error types.

	MUI	Error Type
Octets:	2	2

Figure 11-37—Error KDE format

Table 11-7—MUI values

Handshake type	MUI value
4-Way PTK Handshake	00–01
4-Way STK Handshake	00–02
GTK Handshake	00–03
SMK Handshake	00–04

Table 11-8—SMK error types

Error name	Error type	Error meaning
ERR_STA_NR	1	STA is not reachable from AP. See 11.6.8.5.2
ERR_STA_NRSN	2	STA to AP secure network not present. See 11.6.8.5.3
ERR_CPHR_NS	3	Cipher suites not supported. See 11.6.8.5.4.
ERR_NO_STSL	4	No STSL session present. See 11.6.8.5.5.

The format of the IGTK KDE is shown in Figure 11-38. The IPN corresponds to the last packet number used by the broadcast/multicast transmitter, and is used by the receiver as the initial value for the BIP replay counter.

	KeyID	IPN	IGTK
Octets:	2	6	(Length – 12)

Figure 11-38—IGTK KDE format

The following EAPOL-Key frames are used to implement the three different exchanges:

— **4-Way Handshake Message 1** is an EAPOL-Key frame with the Key Type subfield equal to 1. The Key Data field shall contain an encapsulated PMKID for the PMK that is being used in this key derivation and need not be encrypted.

— **4-Way Handshake Message 2** is an EAPOL-Key frame with the Key Type subfield equal to 1. The Key Data field shall contain an RSNE and need not be encrypted.

An ESS Supplicant’s SME shall insert the RSNE it sent in its (Re)Association Request frame. The RSNE is included as transmitted in the management frame. On receipt of Message 2, the

Authenticator's SME shall validate the selected security configuration against the RSNE received in the IEEE 802.11 (Re)Association Request.

An IBSS Supplicant's SME shall insert an RSNE containing a selected pairwise cipher suite. The Authenticator's SME shall validate that the pairwise cipher suite selected is one of its configured cipher suites and that the group cipher suite and AKM are consistent.

— **4-Way Handshake Message 3** is an EAPOL-Key frame with the Key Type subfield equal to 1. The Key Data field shall contain one or two RSNEs. If a group cipher has been negotiated, this field shall also include an encapsulated GTK. This field shall be encrypted if a GTK is included.

  An Authenticator's SME shall insert the RSNE it sent in its Beacon or Probe Response frame. The Supplicant's SME shall validate the selected security configuration against the RSNE received in Message 3. If the second optional RSNE is present, the STA shall either use that cipher suite with its pairwise key or deauthenticate. In either case, if the values do not match, then the receiver shall consider the RSNE modified and shall use the MLME-DEAUTHENTICATE.request primitive to break the association. A security error should be logged at this time.

  It may happen, for example, that a Supplicant selects a pairwise cipher suite which is advertised by an AP, but which policy disallows for this particular STA. An Authenticator may, therefore, insert a second RSNE to overrule the STA's selection. An Authenticator's SME shall insert the second RSNE, after the first RSNE, only for this purpose. The pairwise cipher suite in the second RSNE included shall be one of the ciphers advertised by the Authenticator. All other fields in the second RSNE shall be identical to the first RSNE.

  An encapsulated GTK shall be included and the unencrypted length of the GTK is six less than the length of the GTK KDE in octets. The entire Key Data field shall be encrypted as specified by the key descriptor version.

— **4-Way Handshake Message 4** is an EAPOL-Key frame with the Key Type subfield equal to 1. The Key Data field can be empty.

— **Group Key Handshake Message 1** is an EAPOL-Key frame with the Key Type subfield equal to 0. The Key Data field shall contain a GTK KDE and shall be encrypted.

— **Group Key Handshake Message 2** is an EAPOL-Key frame with the Key Type subfield equal to 0. The Key Data field can be empty.

PeerKey Handshake Messages use EAPOL-Key frames as defined in 11.6.8.

The key wrap algorithm selected depends on the negotiated AKM as defined in 11.6.3.

The format of the Key ID KDE is shown in Figure 11-39.

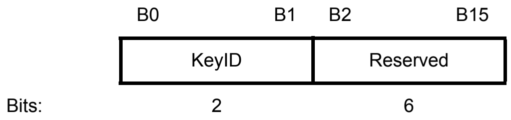


Figure 11-39—Key ID KDE

The KeyID field contains the Authenticator selected Key ID.

11.6.3 EAPOL-Key frame construction and processing

EAPOL-Key frames are constructed and processed according to the AKM negotiated at association time. The negotiated AKM determines what algorithm is used to construct and verify a MIC, the size of the MIC, and the algorithm used to wrap and unwrap the Key Data field.

Table 11-9 indicates the particular algorithms to use when constructing and processing EAPOL-Key frames. The AKM of "Deprecated" indicates an AKM of 00-0F-AC:1 or 00-0F-AC:2 when either TKIP or "Use Group Cipher" is the negotiated pairwise cipher. For all other AKMs the negotiated pairwise cipher suite does not influence the algorithms used to process EAPOL-Key frames.

Table 11-9—Integrity and key-wrap algorithms

AKM	Integrity algorithm	Size of MIC	Key-wrap algorithm
Deprecated	HMAC-MD5	16	ARC4
00-0F-AC:1	HMAC-SHA1-128	16	NIST AES Key Wrap
00-0F-AC:2	HMAC-SHA1-128	16	NIST AES Key Wrap
00-0F-AC:3	AES-128-CMAC	16	NIST AES Key Wrap
00-0F-AC:4	AES-128-CMAC	16	NIST AES Key Wrap
00-0F-AC:5	AES-128-CMAC	16	NIST AES Key Wrap
00-0F-AC:6	AES-128-CMAC	16	NIST AES Key Wrap

11.6.4 EAPOL-Key frame notation

The following notation is used throughout the remainder of 11.6 and 12.4 to represent EAPOL-Key frames:

EAPOL-Key(S, M, A, I, K, SM, KeyRSC, ANonce/SNonce, MIC, DataKDs)

where

S means the initial key exchange is complete. This is the Secure bit of the Key Information field.

M means the MIC is available in message. This should be set in all messages except Message 1 of a 4-Way Handshake. This is the Key MIC bit of the Key Information field.

A means a response is required to this message. This is used when the receiver should respond to this message. This is the Key Ack bit of the Key Information field.

I is the Install bit: Install/Not install for the pairwise key. This is the Install bit of the Key Information field.

K is the key type: P (Pairwise), G (Group/SMK). This is the Key Type bit of the Key Information field.

SM is the SMK Message bit: indicates that this message is part of SMK Handshake.

KeyRSC is the key RSC. This is the Key RSC field.

ANonce/SNonce is the Authenticator/Supplicant nonce. This is the Key Nonce field.

MIC is the integrity check, which is generated using the KCK. This is the Key MIC field.

DataKDs is a sequence of zero or more elements and KDEs, contained in the Key Data field, which may contain the following:

- RSNE is the RSN element, described in 8.4.2.27.
- RSNE[KeyName] is the RSN element, with the PMKID field set to KeyName.
- GTK[N] is the GTK, with the key identifier field set to N. The key identifier specifies which index is used for this GTK. Index 0 shall not be used for GTKs, except in mixed environments, as described in 11.6.1.
- FTE is the Fast BSS Transition element, described in 8.4.2.50
- MDE is the Mobility Domain element, described in 8.4.2.49

TIE[IntervalType] is a Timeout Interval element of type IntervalType, as described in 8.4.2.51, containing e.g., for type KeyLifetime, the lifetime of the FT key hierarchy.
IGTK[M] is the IGTK, with key identifier field set to M.
IPN is the current IGTK replay counter value provided by the IGTK KDE
PMKID is of type PMKID KDE and is the key identifier used during 4-Way PTK Handshake for PMK key identification and during 4-Way STK Handshake for SMK key identification.
Lifetime is the key lifetime KDE used for sending the expiry timeout value for SMK used during PeerKey Handshake for SMK identification.
Initiator MAC is the Initiator MAC KDE used during PeerKey Handshake
Peer MAC is the Peer MAC KDE used during PeerKey Handshake
Initiator Nonce is the Initiator Nonce KDE used during PeerKey Handshake. This is used when multiple nonces need to be sent.
Peer Nonce is the Peer Nonce KDE used during PeerKey Handshake. This is used when multiple nonces need to be sent.
SMK KDE is the encapsulated SMK during SMK Handshake.
Error KDE is an error KDE used when error bit E is equal to 1 during PeerKey Handshake.

11.6.5 Nonce generation

The following is an informative description of Nonce generation.

All STAs contain a global key counter, which is 256 bits in size. It should be initialized at system boot-up time to a fresh cryptographic-quality random number. Refer to M.5 on random number generation. It is recommended that the counter value is initialized to the following:

PRF-256(Random number, “Init Counter”, Local MAC Address || Time)

The local MAC address should be AA on the Authenticator and SPA on the Supplicant.

The random number is 256 bits in size. Time should be the current time [from Network Time Protocol (NTP) or another time in NTP format] whenever possible. This initialization is to ensure that different initial key counter values occur across system restarts regardless of whether a real-time clock is available. The key counter is incremented (all 256 bits) each time a value is used as an IV. The key counter is not allowed to wrap to the initialization value.

11.6.6 4-Way Handshake

11.6.6.1 General

RSNA defines a protocol using IEEE 802.1X EAPOL-Key frames called the 4-Way Handshake. The handshake completes the IEEE 802.1X authentication process. The information flow of the 4-Way Handshake is as follows:

Message 1:Authenticator → Supplicant: EAPOL-Key(0,0,1,0,P,0,0,ANonce,0,DataKD_M1)
where DataKD_M1 = 0 or PMKID for PTK generation, or PMKID KDE (for sending SMKID) for STK generation

Message 2:Supplicant → Authenticator: EAPOL-Key(0,1,0,0,P,0,0,SNonce,MIC,DataKD_M2)
where DataKD_M2 = RSNE for creating PTK generation or peer RSNE, Lifetime KDE, SMKID KDE (for sending SMKID) for STK generation

Message 3:Authenticator → Supplicant:
EAPOL-Key(1,1,1,1,P,0,KeyRSC,ANonce,MIC,DataKD_M3)

where DataKD_M3 = RSNE,GTK[N] for creating PTK generation or initiator RSNE, Lifetime KDE for STK generation

Message 4:Supplicant → Authenticator: EAPOL-Key(1,1,0,0,P,0,0,0,MIC,DataKD_M4)
where DataKD_M4 = 0.

The FT Initial Mobility Domain Association uses the FT 4-way handshake to establish an initial FT Security Association, that is based on this protocol. The FT 4-way handshake protocol is described in 12.4.

Here, the following assumptions apply:

- EAPOL-Key(·) denotes an EAPOL-Key frame conveying the specified argument list, using the notation introduced in 11.6.4.
- ANonce is a nonce that the Authenticator contributes for PTK generation or that the initiator STA contributes for STK generation. ANonce has the same value in Message 1 and Message 3.
- SNonce is a nonce from the Supplicant for PTK generation or from the peer STA for STK generation.
- P means the pairwise bit is set.
- The MIC is computed over the body of the EAPOL-Key frame (with the Key MIC field first zeroed before the computation) using the KCK defined in 11.6.1.3 for PTK generation or SKCK defined in 11.6.1.6.
- RSNE represents the appropriate RSNEs.
- GTK[N] represents the encapsulated GTK with its key identifier.
- SMKID represents the SMKID key identifier used during STK generation.
- Lifetime represents the expiry timeout used for exchanging SMK expiry value.

NOTE—While the MIC calculation is the same in each direction, the Key Ack bit is different in each direction. It is set in EAPOL-Key frames from the Authenticator and 0 in EAPOL-Key frames from the Supplicant. 4-Way Handshake requests from the Supplicant have the Request bit equal to 1. The Authenticator and Supplicant need to check these bits to stop reflection attacks. It is important that Message 1 contents not be used to update state, in particular the keys in use, until the data are validated with Message 3.

11.6.6.2 4-Way Handshake Message 1

Message 1 uses the following values for each of the EAPOL-Key frame fields:

Descriptor Type = N – see 11.6.2

Key Information:

Key Descriptor Version = 1 (ARC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA1-128) or 3 (NIST AES key wrap with AES-128-CMAC), in all other cases 0

Key Type = 1 (Pairwise)

SMK Message = 0

Install = 0

Key Ack = 1

Key MIC = 0

Secure = 0

Error = 0

Request = 0

Encrypted Key Data = 0

Reserved = 0 – unused by this protocol version

Key Length = Cipher-suite-specific; see Table 11-4

Key Replay Counter = *n* – to allow Authenticator or initiator STA to match the right Message 2 from Supplicant or peer STA

Key Nonce = ANonce

EAPOL-Key IV = 0

Key RSC = 0

Key MIC = 0

Key Data Length = length of Key Data field in octets

Key Data = PMKID for the PMK being used during PTK generation or SMKID for SMK being used during STK generation

Processing for PTK generation is as follows:

The Authenticator sends Message 1 to the Supplicant at the end of a successful IEEE 802.1X authentication, after (re)association completes for a STA that has authenticated with SAE or PSK authentication is negotiated, when a cached PMKSA is used, or after a STA requests a new key. On reception of Message 1, the Supplicant determines whether the Key Replay Counter field value has been used before with the current PMKSA. If the Key Replay Counter field value is less than or equal to the current local value, the Supplicant discards the message. Otherwise, the Supplicant:

a) Generates a new nonce SNonce.
b) Derives PTK.
c) Constructs Message 2.

Processing for STK generation is as follows:

The initiator STA (STA_I) sends Message 1 to the peer STA (STA_P) at the end of a successful SMK Handshake, when SMKSA is created. On reception of Message 1, the STA_P determines whether the Key Replay Counter field value has been used before with the current SMKSA. If the Key Replay Counter field value is less than or equal to the current local value, the STA_P discards the message. Otherwise, the STA_P

a) Generates 256-bit random number that is sent as a peer nonce as part of the Key Nonce field. This Nonce is different from the peer nonce generated as part of the SMK Handshake Message 3.
b) Derives STK.
c) Constructs Message 2.

11.6.6.3 4-Way Handshake Message 2

Message 2 uses the following values for each of the EAPOL-Key frame fields:

Descriptor Type = N – see 11.6.2

Key Information:

Key Descriptor Version = 1 (ARC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA1-128) or 3 (NIST AES key wrap with AES-128-CMAC), in all other cases 0 – same as Message 1

Key Type = 1 (Pairwise) – same as Message 1

SMK Message = 0 – same as Message 1

Install = 0

Key Ack = 0

Key MIC = 1

Secure = 0 – same as Message 1

Error = 0 – same as Message 1

Request = 0 – same as Message 1

Encrypted Key Data = 0

Reserved = 0 – unused by this protocol version

Key Length = 0

Key Replay Counter = *n* – to let the Authenticator or initiator STA know to which Message 1 this corresponds

Key Nonce = SNonce

EAPOL-Key IV = 0

Key RSC = 0

Key MIC = MIC(KCK, EAPOL) – MIC computed over the body of this EAPOL-Key frame with the Key MIC field first initialized to 0

Key Data Length = length of Key Data field in octets

Key Data = included RSNE – the sending STA's RSNE for PTK generation or peer RSNE, Lifetime of SMK and SMKID for STK generation

Processing for PTK generation is as follows:

The Supplicant sends Message 2 to the Authenticator.

On reception of Message 2, the Authenticator checks that the key replay counter corresponds to the outstanding Message 1. If not, it silently discards the message. Otherwise, the Authenticator:

a) Derives PTK.
b) Verifies the Message 2 MIC.
   1) If the calculated MIC does not match the MIC that the Supplicant included in the EAPOL-Key frame, the Authenticator silently discards Message 2.
   2) If the MIC is valid and it is part of a Fast BSS Transition Initial Mobility Domain Association, see 12.4.2. If the MIC is valid and it is not part of a Fast BSS Transition Initial Mobility Domain Association, the Authenticator checks that the RSNE bit-wise matches that from the (Re)Association Request message.
      i) If these are not exactly the same, the Authenticator uses MLME-DEAUTHENTICATE.request primitive to terminate the association.
      ii) If they do match bit-wise, the Authenticator constructs Message 3.
c) If management frame protection is being negotiated, the AP initializes the SA Query TransactionIdentifier to an implementation-specific non-negative integer value, valid for the current pairwise security association.

Processing for STK generation is as follows:

The STA_P sends Message 2 to the STA_I. On reception of Message 2, the STA_I checks that the key replay counter corresponds to Message 1. If not, it silently discards the message. Otherwise, the STA_I

a) Derives the STK.
b) Verifies the Message 2 MIC using SKCK key. If the calculated MIC does not match the MIC that the STA_P included in the EAPOL-Key frame, the STA_I silently discards Message 2.
c) If the MIC is valid, the STA_I checks that the RSNE bit-wise matches that from the SMK Handshake Message 5. If these are not exactly the same, STA_I silently discards the message and restarts the 4-Way Handshake after deleting the existing 4-Way Handshake states.

d) If they do match bit-wise, the STA_I checks SMKID with the value of SMKID in SMKSA. If these are not exactly the same, STA_I silently discards the message and restarts the 4-Way Handshake after deleting the existing 4-Way Handshake states.

e) If they do match, the STA_I constructs Message 3. It also compares the Key Lifetime value from the KDE with value in its SMKSA. If value in its SMKSA is less, it discards the value received in Message 2. Otherwise, it updates the value in SMKSA with value in Message 2.

11.6.6.4 4-Way Handshake Message 3

Message 3 uses the following values for each of the EAPOL-Key frame fields:

Descriptor Type = N – see 11.6.2

Key Information:

Key Descriptor Version = 1 (ARC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA1-128) or 3 (NIST AES key wrap with AES-128-CMAC), in all other cases 0 – same as Message 1

Key Type = 1 (Pairwise) – same as Message 1

SMK Message = 0 - same as Message 1

Install = 0/1 – For PTK generation, 0 only if the AP does not support key mapping keys, or if the STA has the No Pairwise bit (in the RSN Capabilities field) equal to 1and only the group key is used. For STK generation, this bit is set to 1.

Key Ack = 1

Key MIC = 1

Secure = 1 (keys installed)

Error = 0 – same as Message 1

Request = 0 – same as Message 1

Encrypted Key Data = 1

Reserved = 0 – unused by this protocol version

Key Length = Cipher-suite-specific; see Table 11-4

Key Replay Counter = $n+1$

Key Nonce = ANonce – same as Message 1

EAPOL-Key IV = 0 (Version 2) or random (Version 1)

Key RSC = For PTK generation, starting sequence number that the Authenticator’s STA uses in MPDUs protected by GTK. For STK generation, this is set to 0.

Key MIC = MIC(KCK, EAPOL) or MIC(SKCK, EAPOL) – MIC computed over the body of this EAPOL-Key frame with the Key MIC field first initialized to 0

Key Data Length = length of Key Data field in octets of included RSNEs and GTK

Key Data = For PTK generation, the AP’s Beacon/Probe Response frame’s RSNE, and, optionally, a second RSNE that is the Authenticator’s pairwise cipher suite assignment, and, if a group cipher has been negotiated, the encapsulated GTK and the GTK’s key identifier (see 11.6.2), and if management frame protection is negotiated, the IGTK KDE. For STK generation Initiator RSNE, Lifetime of SMK is used. If the Extended Key ID for Individually Addressed Frames subfield of the RSN Capabilities field is 1 for both the Authenticator/STA_I and Supplicant/STA_P, then the Authenticator/STA_I includes the Key ID KDE with the assigned key identifier.

Processing for PTK generation is as follows:

If the Extended Key ID for Individually Addressed Frames subfield of the RSN Capabilities field is 1 for both the Authenticator and the Supplicant, then the Authenticator assigns a new Key ID for the PTKSA in the range 0 to 1 that is different from the Key ID assigned in the previous handshake and uses the MLME-SETKEYS.request primitive to install the new key in the IEEE 802.11 MAC to receive individually addressed MPDUs protected by the PTK with the assigned Key ID. Otherwise the Key ID 0 is used and installation of the key is deferred until after Message 4 has been received. The Authenticator sends Message 3 to the Supplicant.

NOTE—If an existing PTK is still in effect, the Authenticator IEEE 802.11 MAC continues to transmit protected, individually addressed MPDUs (if any) using the existing key. With the installation of the new key for receive, the Authenticator is able to receive protected, individually addressed MPDUs using either the old key (if present) or the new key.

On reception of Message 3, the Supplicant silently discards the message if the Key Replay Counter field value has already been used or if the ANonce value in Message 3 differs from the ANonce value in Message 1. The Supplicant also:

a) Verifies the RSNE. If it is part of a Fast BSS Transition Initial Mobility Domain Association, see 12.4.2. Otherwise, if it is not identical to that the STA received in the Beacon or Probe Response frame, the STA shall disassociate. If a second RSNE is provided in the message, the Supplicant uses the pairwise cipher suite specified in the second RSNE or deauthenticates.
b) Verifies the Message 3 MIC. If the calculated MIC does not match the MIC that the Authenticator included in the EAPOL-Key frame, the Supplicant silently discards Message 3.
c) Updates the last-seen value of the Key Replay Counter field.
d) If the Extended Key ID for Individually Addressed Frames subfield of the RSN Capabilities field is 1 for both the Authenticator and Supplicant: Uses the MLME-SETKEYS.request primitive to configure the IEEE 802.11 MAC to receive individually addressed MPDUs protected by the PTK with the assigned Key ID.
e) Constructs Message 4.
f) Sends Message 4 to the Authenticator.
g) Uses the MLME-SETKEYS.request primitive to configure the IEEE 802.11 MAC to send and, if the receive key has not yet been installed, to receive individually addressed MPDUs protected by the PTK. The GTK is also configured by MLME-SETKEYS primitive.

Processing for STK generation is as follows:

If the Extended Key ID for Individually Addressed Frames subfield of the RSN Capabilities field is set to 1 for both the STA_I and the STA_P, then the Authenticator assigns a new Key ID for the STKSA in the range 0 to 1 that is different from the Key ID assigned in the previous handshake and uses the MLME-SETKEYS.request primitive to install the new key in the IEEE 802.11 MAC to receive individually addressed MPDUs protected by the STK with the assigned Key ID. Otherwise the Key ID 0 is used and installation of the key is deferred until after Message 4 has been received. The STA_I sends Message 3 to the STA_P.

NOTE—If an existing STK is still in effect, the STA_I IEEE 802.11 MAC continues to transmit protected, individually addressed MPDUs (if any) using the existing key. With the installation of the new key for receive, the STA_I is able to receive protected, individually addressed MPDUs using both the old key (if present) or the new key.

On reception of Message 3, the STA_P silently discards the message if the Key Replay Counter field value has already been used or if the INonce value in Message 3 differs form the INonce value in Message 1. Otherwise,

a) The STA_P verifies the Message 3 MIC using SKCK key in SMKSA. If the calculated MIC does not match the MIC that the STA_P included in the EAPOL-Key frame, the STA_I silently discards Message 3.

b) If the MIC is valid, the STA_P checks that the RSNE bit-wise matches that from the 4-Way Handshake Message 2. If these are not exactly the same, STA_P silently discards the message and deletes existing 4-Way Handshake states.

c) If they do match, the STA_P constructs Message 4. It also compares the Key Lifetime value from KDE with value in its SMKSA. If value in SMKSA is less, it discards the value received in Message 3. Otherwise, it updates the value in SMKSA with value in Message 3.

d) If the Extended Key ID for Individually Addressed Frames subfield of the RSN Capabilities field is 1 for both the Authenticator and Supplicant then prior to sending Message 4, STA_P uses the MLME-SETKEYS.request primitive to configure the IEEE 802.11 MAC to receive individually addressed MPDUs protected by the STK with the assigned Key ID.

e) After sending Message 4, STA_P uses the MLME-SETKEYS.request primitive to configure the IEEE 802.11 MAC to send and, if the receive key has not yet been installed, to receive individually addressed MPDUs protected by the STK with the assigned Key ID.

11.6.6.5 4-Way Handshake Message 4

Message 4 uses the following values for each of the EAPOL-Key frame fields:

Descriptor Type = N – see 11.6.2

Key Information:

Key Descriptor Version = 1 (ARC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA1-128) or 3 (NIST AES key wrap with AES-128-CMAC), in all other cases 0 – same as Message 1

Key Type = 1 (Pairwise) – same as Message 1

SMK Message = 0 - same as Message 1

Install = 0

Key Ack = 0 – this is the last message

Key MIC = 1

Secure = 1

Error = 0

Request = 0

Encrypted Key Data = 0

Reserved = 0 – unused by this protocol version

Key Length = 0

Key Replay Counter = *n+1*

Key Nonce = 0

EAPOL-Key IV = 0

Key RSC = 0

Key MIC = MIC(KCK, EAPOL) or MIC(SKCK, EAPOL) – MIC computed over the body of this EAPOL-Key frame with the Key MIC field first initialized to 0

Key Data Length = length of Key Data field in octets

Key Data = none required

Processing for PTK generation is as follows:

The Supplicant sends Message 4 to the Authenticator. Note that when the 4-Way Handshake is first used, Message 4 is sent in the clear.

On reception of Message 4, the Authenticator verifies that the Key Replay Counter field value is one that it used on this 4-Way Handshake; if it is not, it silently discards the message. Otherwise:

a) The Authenticator checks the MIC. If the calculated MIC does not match the MIC that the Supplicant included in the EAPOL-Key frame, the Authenticator silently discards Message 4.
b) If the MIC is valid, the Authenticator uses the MLME-SETKEYS.request primitive to configure the IEEE 802.11 MAC to send and, if the receive key has not yet been installed, to receive protected, individually addressed MPDUs using for the new PTK.
c) The Authenticator updates the Key Replay Counter field so that it uses a fresh value if a rekey becomes necessary.

Processing for STK generation is as follows:

The STA_P sends Message 4 to the STA_I. On reception of Message 4, the STA_I verifies that the Key Replay Counter field value is one that it used on this 4-Way Handshake; if it is not, it silently discards the message. Otherwise,

a) The STA_I checks the MIC. If the calculated MIC does not match the MIC that the STA_P included in the EAPOL-Key frame, the STA_I silently discards Message 4.
b) If the MIC is valid, the STA_I uses the MLME-SETKEYS.request primitive to configure the 802.11 MAC to send and, if the receive key has not yet been installed, to receive protected, individually addressed MPDUs using for the new STK.
c) The STA_I updates the Key Replay Counter field so that it uses a fresh value if a rekey becomes necessary.

11.6.6.6 4-Way Handshake implementation considerations

When the 4-Way Handshake is used as part of the STK Handshake, the initiator STA acts as Authenticator and peer STA acts as Supplicant.

If the Authenticator does not receive a reply to its messages, it shall attempt dot11RSNAConfigPairwiseUpdateCount transmits of the message, plus a final timeout. The retransmit timeout value shall be 100 ms for the first timeout, half the listen interval for the second timeout, and the listen interval for subsequent timeouts. If there is no listen interval or the listen interval is zero, then 100 ms shall be used for all timeout values. If it still has not received a response after these retries, then for PTK generation the Authenticator should deauthenticate the STA, and for STK generation the STAs should delete the SMKSA and initiate an STSL application teardown procedure.

For PTK generation, if the STA does not receive Message 1 within the expected time interval (prior to IEEE 802.1X timeout), it should disassociate, deauthenticate, and try another AP/STA. For STK generation, if the peer STA does not receive Message 1 or Message 3 within the expected time interval (prior to dot11RSNAConfigSATimeout as specified in 11.6.8), it deletes the SMKSA and invokes an STSL application teardown procedure.

The Authenticator should ignore EAPOL-Key frames it is not expecting in reply to messages it has sent or EAPOL-Key frames in which the Ack bit is 1. This stops an attacker from sending the first message to the Supplicant who responds to the Authenticator.

An implementation should save the KCK and KEK beyond the 4-Way Handshake, as they are needed for Group Key Handshakes, STK Rekeying, and recovery from TKIP MIC failures.

The Supplicant uses the MLME-SETKEYS.request primitive to configure the temporal key from 11.6.1 into its STA after sending Message 4 to the Authenticator.

If the RSNE check for Message 2 or Message 3 fails, the SME should log an error and deauthenticate the peer.

11.6.6.7 Sample 4-Way Handshake

The following is an informative sample of a 4-Way Handshake for illustration.

After IEEE 802.1X authentication completes by the AP sending an EAP-Success, the AP initiates the 4-Way Handshake. See Figure 11-40.

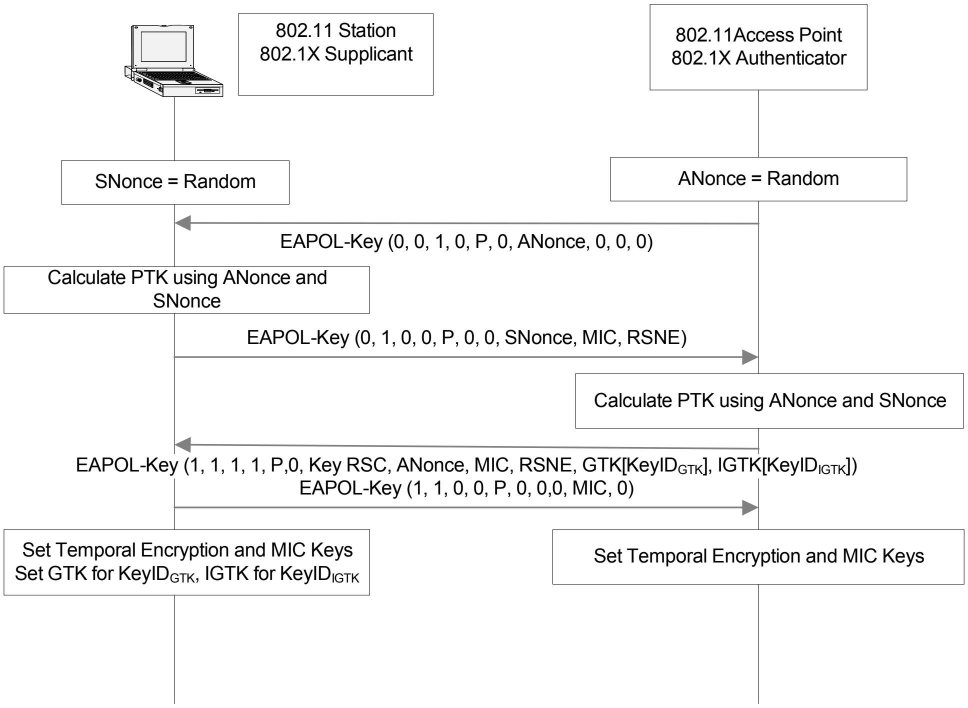

802.11 Station
802.1X Supplicant
802.11Access Point
802.1X Authenticator
SNonce = Random
ANonce = Random
EAPOL-Key (0, 0, 1, 0, P, 0, ANonce, 0, 0, 0)
Calculate PTK using ANonce and SNonce
EAPOL-Key (0, 1, 0, 0, P, 0, 0, SNonce, MIC, RSNE)
Calculate PTK using ANonce and SNonce
EAPOL-Key (1, 1, 1, 1, P,0, Key RSC, ANonce, MIC, RSNE, GTK[KeyID$_{GTK}$], IGTK[KeyID$_{IGTK}$])
EAPOL-Key (1, 1, 0, 0, P, 0, 0,0, MIC, 0)
Set Temporal Encryption and MIC Keys
Set GTK for KeyID$_{GTK}$, IGTK for KeyID$_{IGTK}$
Set Temporal Encryption and MIC Keys

Figure 11-40—Sample 4-Way Handshake

The 4-Way Handshake consists of the following steps:

a) The Authenticator sends an EAPOL-Key frame containing an ANonce.

b) The Supplicant derives a PTK from ANonce and SNonce.

c) The Supplicant sends an EAPOL-Key frame containing SNonce, the RSNE from the (Re)Association Request frame, and a MIC.

d) The Authenticator derives PTK from ANonce and SNonce and validates the MIC in the EAPOL-Key frame.

e) The Authenticator sends an EAPOL-Key frame containing ANonce, the RSNE from its Beacon or Probe Response messages, MIC, whether to install the temporal keys, the encapsulated GTK, and if management frame protection is negotiated, the IGTK.

f) The Supplicant sends an EAPOL-Key frame to confirm that the temporal keys are installed.

11.6.6.8 4-Way Handshake analysis

The following is an informative analysis of the 4-Way Handshake.

This subclause makes the trust assumptions used in this protocol explicit. The protocol assumes the following:

- The PMK is known only by the Supplicant's STA and the Authenticator's STA.
- The Supplicant's STA uses IEEE 802 address SPA.
- The Authenticator's STA uses IEEE 802 address AA.

In many instantiations the RSNA architecture immediately breaks the first assumption because the IEEE 802.1X AS also knows the PMK. Therefore, additional assumptions are required:

- The AS does not expose the PMK to other parties.
- The AS does not masquerade as the Supplicant to the Authenticator.
- The AS does not masquerade as the Authenticator to the Supplicant.
- The AS does not masquerade as the Supplicant's STA.
- The AS does not masquerade as the Authenticator's STA.

The protocol also assumes this particular Supplicant-Authenticator pair is authorized to know this PMK and to use it in the 4-Way Handshake. If any of these assumptions are broken, then the protocol fails to provide any security guarantees.

The protocol also assumes that the AS delivers the correct PMK to the AP with IEEE 802 address AA and that the STA with IEEE 802 address SPA hosts the Supplicant that negotiated the PMK with the AS. None of the protocols defined by this standard and IEEE Std 802.1X-2004 permit the AS, the Authenticator, the Supplicant, or either STA to verify these assumptions.

The PTK derivation step

$$\text{PTK} \leftarrow \text{PRF-X(PMK, “Pairwise key expansion”} \,||\, \text{Min(AA,SPA)} \,||\, \text{Max(AA,SPA)} \,||\, \text{Min(ANonce,SNonce)} \,||\, \text{Max(ANonce,SNonce))}$$

performs a number of functions:

- Including the AA and SPA in the computation
  - Binds the PTK to the communicating STAs and
  - Prevents undetected man-in-the-middle attacks against 4-Way Handshake messages between the STAs with these two IEEE 802 addresses.
- If ANonce is randomly selected, including ANonce
  - Guarantees the STA at IEEE 802 address AA that PTK is fresh,
  - Guarantees that Message 2 and Message 4 are live, and
  - Uniquely identifies PTK as <AA, ANonce>.
- If SNonce is randomly selected, including SNonce
  - Guarantees the STA at IEEE 802 address SPA that PTK is fresh,
  - Guarantees that Message 3 is live, and
  - Uniquely identifies PTK as <SPA, SNonce>.

Choosing the nonces randomly helps prevent precomputation attacks. With unpredictable nonces, a man-in-the-middle attack that uses the Supplicant to precompute messages to attack the Authenticator cannot progress beyond Message 2, and a similar attack against the Supplicant cannot progress beyond Message 3. The protocol might execute further before an error if predictable nonces are used.

Message 1 delivers ANonce to the Supplicant and initiates negotiation for a new PTK. It identifies AA as the peer STA to the Supplicant's STA. If an adversary modifies either of the addresses or ANonce, the Authenticator detects the result when validating the MIC in Message 2. Message 1 does not carry a MIC, as it is impossible for the Supplicant to distinguish this message from a replay without maintaining state of all security associations through all time (PMK might be a static key).

Message 2 delivers SNonce to the Authenticator so it can derive the PTK. If the Authenticator selected ANonce randomly, Message 2 also demonstrates to the Authenticator that the Supplicant is live, that the PTK is fresh, and that there is no man-in-the-middle attack, as the PTK includes the IEEE 802 MAC addresses of both. Inclusion of ANonce in the PTK derivation also protects against replay. The MIC prevents undetected modification of Message 2 contents.

Message 3 confirms to the Supplicant that there is no man-in-the-middle attack. If the Supplicant selected SNonce randomly, it also demonstrates that the PTK is fresh and that the Authenticator is live. The MIC again prevents undetected modification of Message 3.

While Message 4 serves no cryptographic purpose, it serves as an acknowledgment to Message 3. It is required to ensure reliability and to inform the Authenticator that the Supplicant has installed the PTK and GTK and hence can receive encrypted frames.

The PTK and GTK are installed by using MLME-SETKEYS.request primitive after Message 4 is sent. The PTK is installed before the GTK.

Then the 4-Way Handshake uses a correct, but unusual, mechanism to guard against replay. As noted earlier in this subclause, ANonce provides replay protection to the Authenticator, and SNonce to the Supplicant. In most session initiation protocols, replay protection is accomplished explicitly by selecting a nonce randomly and requiring the peer to reflect the received nonce in a response message. The 4-Way Handshake instead mixes ANonce and SNonce into the PTK, and replays are detected implicitly by MIC failures. In particular, the Key Replay Counter field serves no cryptographic purpose in the 4-Way Handshake. Its presence is not detrimental, however, and it plays a useful role as a minor performance optimization for processing stale instances of Message 2. This replay mechanism is correct, but its implicit nature makes the protocol harder to understand than an explicit approach.

It is critical to the correctness of the 4-Way Handshake that at least one bit differs in each message. Within the 4-Way Handshake, Message 1 can be recognized as the only one in which the Key MIC bit is 0, meaning Message 1 does not include the MIC, while Message 2 to Message 4 do. Message 3 differs from Message 2 by not asserting the Ack bit and from Message 4 by asserting the Ack Bit. Message 2 differs from Message 4 by including the RSNE.

Request messages are distinct from 4-Way Handshake messages because the former asserts the Request bit and 4-Way Handshake messages do not. Group Key Handshake messages are distinct from 4-Way Handshake messages because they assert a different key type.

11.6.7 Group Key Handshake

11.6.7.1 General

The Authenticator uses the Group Key Handshake to send a new GTK and, if management frame protection is negotiated, a new IGTK to the Supplicant.

The Authenticator may initiate the exchange when a Supplicant is disassociated or deauthenticated.

> Message 1: Authenticator → Supplicant:
> EAPOL-Key(1,1,1,0,G,0,Key RSC,0, MIC,GTK[N],IGTK[M])
>
> Message 2: Supplicant → Authenticator: EAPOL-Key(1,1,0,0,G,0,0,0,MIC,0)

Here, the following assumptions apply:

- Key RSC denotes the last frame sequence number sent using the GTK.
- GTK[N] denotes the GTK encapsulated with its key identifier as defined in 11.6.2 using the KEK defined in 11.6.1.3 and associated IV.
- IGTK[M], when present, denotes the IGTK encapsulated with its key identifier as defined in 11.6.2 using the KEK defined in 11.6.1.3 and associated IV.
- The MIC is computed over the body of the EAPOL-Key frame (with the MIC field zeroed for the computation) using the KCK defined in 11.6.1.3.

The Supplicant may trigger a Group Key Handshake by sending an EAPOL-Key frame with the Request bit set to 1 and the type of the Group Key bit.

An Authenticator shall do a 4-Way Handshake before a Group Key Handshake if both are required to be done.

NOTE—The Authenticator cannot initiate the Group Key Handshake until the 4-Way Handshake completes successfully.

11.6.7.2 Group Key Handshake Message 1

Message 1 uses the following values for each of the EAPOL-Key frame fields:

Descriptor Type = N – see 11.6.2

Key Information:

Key Descriptor Version = 1 (ARC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA1-128) or 3 (NIST AES key wrap with AES-128-CMAC), in all other cases 0

Key Type = 0 (Group/SMK)

SMK Message = 0

Install = 0

Key Ack = 1

Key MIC = 1

Secure = 1

Error = 0

Request = 0

Encrypted Key Data = 1

Reserved = 0

Key Length = 0

Key Replay Counter = n+2

Key Nonce = 0

EAPOL-Key IV = 0 (Version 2) or random (Version 1)

Key RSC = last transmit sequence number for the GTK

Key MIC = MIC(KCK, EAPOL)

Key Data Length = Cipher-suite-specific; see Table 11-4

Key Data = encrypted, encapsulated

- GTK and the GTK's key identifier (see 11.6.2)
- When present, IGTK, IGTK's key identifier, and IPN (see 11.6.2)

The Authenticator sends Message 1 to the Supplicant.

On reception of Message 1, the Supplicant:

a) Verifies that the Key Replay Counter field value has not yet been seen before, i.e., its value is strictly larger than that in any other EAPOL-Key frame received thus far during this session.

b) Verifies that the MIC is valid, i.e., it uses the KCK that is part of the PTK to verify that there is no data integrity error.

c) Uses the MLME-SETKEYS.request primitive to configure the temporal GTK and, when present, IGTK into its IEEE 802.11 MAC.

d) Responds by creating and sending Message 2 of the Group Key Handshake to the Authenticator and incrementing the replay counter.

NOTE—The Authenticator increments and uses a new Key Replay Counter field value on every Message 1 instance, even retries, because the Message 2 responding to an earlier Message 1 might have been lost. If the Authenticator did not increment the replay counter, the Supplicant discards the retry, and no responding Message 2 ever arrives.

11.6.7.3 Group Key Handshake Message 2

Message 2 uses the following values for each of the EAPOL-Key frame fields:

Descriptor Type = N – see 11.6.2

Key Information:

Key Descriptor Version = 1 (ARC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA1-128) or 3 (NIST AES key wrap with AES-128-CMAC), in all other cases 0 – same as Message 1

Key Type = 0 (Group/SMK) – same as Message 1

Install = 0

Key Ack = 0

Key MIC = 1

Secure = 1

Error = 0

Request = 0

Encrypted Key Data = 0

Reserved = 0

Key Length = 0

Key Replay Counter = *n*+2 – same as Message 1

Key Nonce = 0

EAPOL-Key IV = 0

Key RSC = 0

Key MIC = MIC(KCK, EAPOL)

Key Data Length = 0

Key Data = none required

On reception of Message 2, the Authenticator:

a) Verifies that the Key Replay Counter field value matches one it has used in the Group Key Handshake.

b) Verifies that the MIC is valid, i.e., it uses the KCK that is part of the PTK to verify that there is no data integrity error.

11.6.7.4 Group Key Handshake implementation considerations

If the Authenticator does not receive a reply to its messages, its shall attempt dot11RSNAConfigGroupUpdateCount transmits of the message, plus a final timeout. The retransmit timeout value shall be 100 ms for the first timeout, half the listen interval for the second timeout, and the listen interval for subsequent timeouts. If there is no listen interval or the listen interval is zero, then 100 ms shall be used for all timeout values. If it still has not received a response after this, then the Authenticator's STA should use the MLME-DEAUTHENTICATE.request primitive to deauthenticate the STA.

11.6.7.5 Sample Group Key Handshake

The following is an informative sample of a Group Key Handshake for illustration.

The state machines in 11.6.10 and 11.6.11 change the GTK and, when present, IGTK in use by the network. See Figure 11-41.

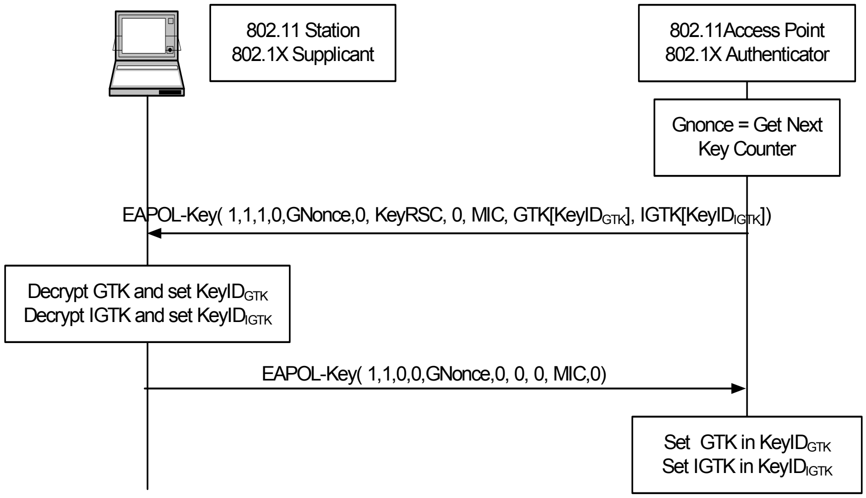

802.11 Station
802.1X Supplicant
802.11Access Point
802.1X Authenticator
Gnonce = Get Next Key Counter
EAPOL-Key(1,1,1,0,GNonce,0, KeyRSC, 0, MIC, GTK[KeyID$_{GTK}$], IGTK[KeyID$_{IGTK}$])
Decrypt GTK and set KeyID$_{GTK}$
Decrypt IGTK and set KeyID$_{IGTK}$
EAPOL-Key(1,1,0,0,GNonce,0, 0, 0, MIC,0)
Set GTK in KeyID$_{GTK}$
Set IGTK in KeyID$_{IGTK}$

Figure 11-41—Sample Group Key Handshake

The following steps occur:

a) The Authenticator generates a new GTK and when management frame protection has been negotiated, a new IGTK. It encapsulates the GTK and, as necessary, the IGTK, and sends an EAPOL-Key frame containing the GTK and IGTK (Message 1), along with the last sequence number used with the GTK (RSC) and the last IPN used with the IGTK.

b) On receiving the EAPOL-Key frame, the Supplicant validates the MIC, decapsulates the GTK and, when present, the IGTK, and uses the MLME-SETKEYS.request primitive to configure the GTK, PN, IGTK, RSC, and IPN in its STA.

c) The Supplicant then constructs and sends an EAPOL-Key frame in acknowledgment to the Authenticator.

d) On receiving the EAPOL-Key frame, the Authenticator validates the MIC. If the GTK and, if present, the IGTK are is not already configured into IEEE 802.11 MAC, after the Authenticator has delivered the GTK and IGTK to all associated STAs, it uses the MLME-SETKEYS.request primitive to configure the GTK and IGTK into the IEEE 802.11 STA.

11.6.8 PeerKey Handshake

11.6.8.1 General

The PeerKey Handshake occurs after any other STSL setup procedures and is used to create an STKSA providing data confidentiality between the two STAs. The AP establishes an RSNA with each STA prior to PeerKey setup. The initiator STA starts the PeerKey Handshake, at the conclusion of which a key is established to secure the connection.

STSL security PeerKey Handshake is used to establish security for data frames passed directly between two STAs associated with the same AP. The AP establishes an RSNA with each STA prior to the PeerKey Handshake. After the STAs establish the STSL, the initiator STA starts the PeerKey Handshake, at the conclusion of which a key is established to secure the connection. The PeerKey Handshake is used to create an STKSA between the two STAs.

The PeerKey EAPOL-Key exchange provides a mechanism for obtaining the keys to be used for direct station-to-station communication. The initiator STA shall start a timer when it sends the first EAPOL-Key message, and the peer STA shall do the same on receipt of the first EAPOL-Key message. On expiration of this timer, the STA shall transition to the STKINIT state.

A STA should use the PeerKey Handshake prior to transferring any direct station-to-station data frames. The STKSA should be deleted when the station-to-station connection is terminated.

Here, the following assumptions apply:

- EAPOL-Key() denotes an EAPOL-Key frame conveying the specified argument list, using the notation introduced in 11.6.4.
- STA_I is the initiator STA.
- STA_P is the peer STA.
- AP is the access point with which both the STA_I and the STA_P are associated.
- MAC_I is the MAC address of the STA_I.
- MAC_P is the MAC address of the STA_I.
- INonce is the nonce generated by the STA_I.
- PNonce is the nonce generated by the STA_P.

The PeerKey Handshake has two components:

a) **SMK Handshake:** This handshake is initiated by the initiator STA, and as a result of this handshake, the SMKSA is installed in both the STAs. This message exchange goes through the AP and is protected using the PTK.
b) **4-Way STK Handshake:** Using the installed SMKSA, the initiator STA initiates the 4-Way Handshake (per 11.6.6.5), and as a result of this, the STKSA gets installed in both the STAs. The STKSA is used for securing data exchange between the initiator STA and the peer STA. The 4-Way Handshake analysis described in 11.6.6.8 applies to the 4-Way STK Handshake.

11.6.8.2 SMK Handshake

11.6.8.2.1 General

The initiator STA initiates the SMK Handshake by sending the first message to the AP to establish an SMKSA between itself and another STA associated with the same AP. Unlike the 4-Way Handshake and the Group Key Handshake, the SMK Handshake is initiated by the initiator STA.

Message 1: Initiator STA → AP: EAPOL-Key(1,1,0,0,0,1,0, INonce, MIC, RSNE_I, MAC_P KDE)

Message 2: AP → Peer STA: EAPOL-Key(1,1,1,0,0,1,0, INonce, MIC, RSNE_I, MAC_I KDE)

Message 3: Peer STA → AP: EAPOL-Key(1,1,0,0,0,1,0, PNonce, MIC, RSNE_P, MAC_I KDE, Initiator Nonce KDE)

Message 4: AP → Peer STA: EAPOL-Key(1,1,0,1,0,1,0, PNonce, MIC, MAC_I KDE, Initiator Nonce KDE, SMK KDE, Lifetime KDE)

Message 5: AP → Initiator STA: EAPOL-Key(1,1,0,0,0,1,0, INonce, MIC, RSNE_P, MAC_P KDE, Peer Nonce KDE, SMK KDE, Lifetime KDE)

11.6.8.2.2 SMK Handshake Message 1

The initiator STA creates the RSNE (see 8.4.2.27) by including the element ID, length, version, and pairwise cipher suite list fields. Since the group cipher suit field is before the pairwise cipher suite list field (so the STA needs to include it), the STA includes any value in this field, and the receiving STA ignores it. The initiator STA also generates a 256-bit random number that is sent in the Key Nonce field.

Message 1 uses the following values for each of the EAPOL-Key frame fields:

Descriptor Type = N – see 11.6.2

Key Information:

Key Descriptor Version = 1 (ARC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA1-128) or 3 (NIST AES key wrap with AES-128-CMAC), in all other cases 0

Key Type = 0 (Group/SMK)

SMK Message = 1 (SMK)

Install = 0

Key Ack = 0

Key MIC = 1

Secure = 1

Error = 0

Request = 1

Encrypted Key Data = 0

Reserved = 0

Key Length = 0

Key Replay Counter = request EAPOL replay counter of initiating STA

Key Nonce = INonce

EAPOL-Key IV = 0

Key RSC = 0

Key MIC = MIC (initiating STA's KCK, EAPOL)

Key Data Length = Length of Key Data field in octets

Key Data = Initiator RSNE, peer MAC address KDE

The STA_I sends Message 1 to the AP.

On receipt of Message 1, the AP checks that the key replay counter corresponds to Message 1. If not, it silently discards the message. Otherwise:

a) The AP verifies the Message 1 MIC using the STA_I PTKSA. If the calculated MIC does not match the MIC that the STA_I included in the EAPOL-Key frame, the AP silently discards Message 1.

b) If the MIC is correct, the AP checks if the STA_P is reachable. If it is not reachable, the AP shall send an error EAPOL-Key message to STA_I per 11.6.8.5.2. After sending the message, AP silently discards Message 1.

c) The AP checks if the AP has a secure connection with STA_P. If not, the AP shall send an error EAPOL-Key message to STA_I per 11.6.8.5.3. After sending the message, AP silently discards Message 1.

d) If all checks succeed, the AP creates Message 2 using the STA_P PTKSA. The AP copies the contents of Message 1 to create Message 2.

11.6.8.2.3 SMK Handshake Message 2

Message 2 uses the following values for each of the EAPOL-Key frame fields:

Descriptor Type = N – see 11.6.2

Key Information:

Key Descriptor Version = 1 (ARC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA1-128) or 3 (NIST AES key wrap with AES-128-CMAC), in all other cases 0

Key Type = 0 (Group/SMK)

SMK Message = 1 (SMK)

Install = 0

Key Ack = 1

Key MIC = 1

Secure = 1

Error = 0

Request = 0

Encrypted Key Data = 0

Reserved = 0

Key Length = 0

Key Replay Counter = request EAPOL replay counter of AP and STA_P

Key Nonce = INonce

EAPOL-Key IV = 0

Key RSC = 0

Key MIC = MIC (KCK of the STA_P, EAPOL)

Key Data Length = Length of Key Data field in octets

Key Data = Initiator RSNE, initiator MAC address KDE

The AP sends Message 2 to the STA_P. On receipt of Message 2, the STA_P checks that the key replay counter corresponds to Message 2. If not, it silently discards the message. Otherwise,

a) The STA_P verifies the Message 2 MIC using the STA_P PTKSA. If the calculated MIC does not match the MIC that the AP included in the EAPOL-Key frame, the STA_P silently discards Message 2.

b) If the MIC is correct, the STA_P checks if it supports at least one cipher suites proposed by the STA_I. If it does not, the STA_P shall send an error EAPOL-Key message to STA_I through the AP per 11.6.8.5.4. After sending the error message, the STA_P silently discards Message 2.

c) STA_O checks if it has already created an STSL with STA_I. If it has not, STA_P shall send an error EAPOL-Key message to STA_I through the AP per 11.6.8.5.5. After sending the error message, the STA_P silently discards Message 2.

d) If all checks succeed, the STA_P creates the state of PeerKey Handshake and stores the INonce and the RSNE received in Message 2.

e) STA_P selects a support cipher suite from the cipher suite list proposed by the STA_I and creates the peer RSNE, which is sent with Message 3.

f) STA_P generates a 256-bit random number that is sent as the peer Nonce KDE with Message 3.

g) Using all the information, STA_P creates Message 3.

11.6.8.2.4 SMK Handshake Message 3

Message 3 uses the following values for each of the EAPOL-Key frame fields:

Descriptor Type = N – see 11.6.2

Key Information:

Key Descriptor Version = 1 (ARC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA1-128) or 3 (NIST AES key wrap with AES-128-CMAC), in all other cases 0

Key Type = 0 (Group/SMK)

SMK Message = 1 (SMK)

Install = 0

Key Ack = 0

Key MIC = 1

Secure = 1

Error = 0

Request = 0

Encrypted Key Data = 0

Reserved = 0

Key Length = 0

Key Replay Counter = request EAPOL replay counter of peer STA

Key Nonce = PNonce

EAPOL-Key IV = 0

Key RSC = 0

Key MIC = MIC (KCK of STA_I, EAPOL)

Key Data Length = Length of Key Data field in octets

Key Data = Peer RSNE, initiator MAC address KDE, initiator Nonce KDE

The STA_P sends Message 3 to the AP. On receipt of Message 3, the AP checks that the key replay counter corresponds to Message 3. If not, it silently discards the message. Otherwise,

a) The AP verifies the Message 1 MIC using the STA_I PTKSA. If the calculated MIC does not match the MIC that the STA_P included in the EAPOL-Key frame, the AP silently discards Message 1.

b) If MIC is correct, the AP checks if the STA_I is reachable. If it is not reachable, the AP shall send an error EAPOL-Key message to the STA_P per 11.6.8.5.2. After sending the message, the AP silently discards Message 3.

c) The AP checks if the AP has secure connection with STA_I. If it does not, the AP shall send an error EAPOL-Key message to STA_P per 11.6.8.5.3. After sending the message, the AP silently discards Message 3.

d) If all checks succeed, the AP generates a 256-bit random number that is used as the SMK, which is sent with Message 4 and Message 5 as SMK KDEs.

e) Depending on the strength of random number generator, the AP sets the lifetime of the SMK, which is sent with Message 4 and Message 5 as Lifetime KDEs.

f) Using all the information, the AP creates Message 4 and Message 5.

11.6.8.2.5 SMK Handshake Message 4

Message 4 uses the following values for each of the EAPOL-Key frame fields:

Descriptor Type = N – see 11.6.2

Key Information:

Key Descriptor Version = 1 (ARC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA1-128) or 3 (NIST AES key wrap with AES-128-CMAC), in all other cases 0

Key Type = 0 (Group/SMK)

SMK Message = 1 (SMK)

Install = 1

Key Ack = 0

Key MIC = 1

Secure = 1

Error = 0

Request = 0

Encrypted Key Data = 1

Reserved = 0

Key Length = Cipher-suite-specific; see Table 11-4

Key Replay Counter = request EAPOL replay counter of AP

Key Nonce = PNonce

EAPOL-Key IV = 0

Key RSC = 0

Key MIC = MIC (KCK of the STA_I, EAPOL)

Key Data Length = Length of Key Data field in octets

Key Data = Encrypted initiator MAC address KDE, Initiator Nonce KDE, SMK KDE (contains SMK and PNonce), Lifetime KDE

The AP sends Message 4 to the STA_P. On receipt of Message 4, the STA_P checks that the key replay counter corresponds to Message 4. If it does not, STA_P silently discards the message. Otherwise,

a) The STA_P verifies the Message 4 MIC using STA_P PTKSA. If the calculated MIC does not match the MIC that the AP included in the EAPOL-Key frame, the STA_P silently discards Message 4.

b) If the MIC is correct, STA_P identifies the PeerKey session using the PNonce sent as part of the Key Nonce field of Message 4. If STA_P has an existing PeerKey state for this session, i.e., STA_P has received Message 2 and this message is a follow-up to that. If STA_P has an existing PeerKey state for this session, STA_P silently discards Message 4.

c) If all checks succeed, STA_P decrypts the Key Data field of Message 4 and extracts the MAC_I, the INonce, the PNonce, the SMK, and the lifetime from Message 4. The STA_P verifies the extracted INonce against the INonce originally received as part of Message 2.

d) The STA_P calculates the SMKID per 11.6.1.6.

e) The STA_P checks the value of the lifetime with the maximum value it can support. If the lifetime suggested by the AP is too long, the STA_P selects a lower value that it can support.

f) Using all the information, the STA_P creates the SMKSA for this PeerKey session.

11.6.8.2.6 SMK Handshake Message 5

Message 5 uses the following values for each of the EAPOL-Key frame fields:

Descriptor Type = N – see 11.6.2

Key Information:

Key Descriptor Version = 1 (ARC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA1-128) or 3 (NIST AES key wrap with AES-128-CMAC), in all other cases 0

Key Type = 0 (Group/SMK)

SMK Message = 1 (SMK)

Install = 0

Key Ack = 0

Key MIC = 1

Secure = 1

Error = 0

Request = 0

Encrypted Key Data = 1

Reserved = 0

Key Length = Cipher-suite-specific; see Table 11-4

Key Replay Counter = request EAPOL replay counter of AP

Key Nonce = INonce

EAPOL-Key IV = 0

Key RSC = 0

Key MIC = MIC (KCK of the STA_I, EAPOL)

Key Data Length = Length of Key Data field in octets

Key Data = Encrypted peer RSNE, peer MAC address KDE, peer Nonce KDE, SMK KDE (contains SMK and INonce), Lifetime KDE

The AP sends Message 5 to the STA_I. On receipt of Message 5, the STA_I checks that the key replay counter corresponds to Message 5. If it does not, the STA_I silently discards the message. Otherwise,

a) STA_I verifies the Message 4 MIC using STA P PTKSA. If the calculated MIC does not match the MIC that the AP included in the EAPOL-Key frame, the STA_I silently discards Message 5.

b) If the MIC is correct, the STA_I identifies the PeerKey session using the INonce sent as part of the Key Nonce field of Message 5. If STA_I has an existing PeerKey state for this session, i.e., STA_I has initiated this message exchange using Message 1 and this message is a follow-up to that. If STA_I has an existing PeerKey state for this session, STA_I shall silently discard Message 5.

c) If all checks succeed, STA_I decrypts the Key Data field of Message 5 and extracts the peer RSNE, the MAC_P, the INonce, the PNonce, the SMK, and the lifetime from Message 5.

d) The STA_I verifies that the peer RSNE includes a valid cipher (i.e., one that was included in an initiator RSNE). If not, STA_I discards the message and sends an Error KDE ERR_CPHR_NS.

e) The STA_I calculates SMKID per 11.6.1.6.

f) The STA_I checks the value of the lifetime with the maximum value it can support. If the lifetime suggested by the AP is too long, STA_I selects a lower value that can support.

g) Using all the information, the STA_I creates the SMKSA for this PeerKey session.

11.6.8.3 PeerKey setup and handshake error conditions

If the STA_P does not receive a valid SMK Message 2 or a 4-Way STK Message 1 after sending the EAPOL request message to initiate the PeerKey rekey within a 200 ms timeout, the STA_P shall invoke an STSL application teardown procedure.

If the STA_I does not receive an SMK Message 5 from the AP, the STA_I shall attempt dot11RSNAConfigSMKUpdateCount transmits of the SMK Handshake Message 1 plus a final timeout. If the STA_I still has not received a response after these retries, it shall invoke an STSL application teardown procedure. The retransmit timeout value shall be 200 ms for the first timeout, the listen interval for the second timeout, and twice the listen interval for subsequent timeouts. If there is no listen interval or the listen interval is zero, then 200 ms shall be used for all timeout values.

There is no specific recovery mechanism at the AP if the SMK Message 3 is dropped. This results in a timeout by the STA_I after nonreceipt of SMK Message 5, as described in the preceding paragraph.

If the SMK Message 4 is not received by the STA_P, a failure is detected during the 4-Way STK Handshake. In this case, the STA_P discards the EAPOL-Key messages without the proper key. This failure is covered by behavior described in 11.6.6.6 and results in teardown of the STSL.

Upon receipt of the SMK Message 5, the STA_I transmits Message 1 of the 4-Way STK Handshake to the STA_P. If the STA_I does not receive Message 2 of the 4-Way STK Handshake from the STA_P, it shall attempt dot11RSNAConfigSMKUpdateCount transmits of 4-Way STK Handshake Message 1, plus a final timeout. If STA_I still has not received a response after these retries, it shall invoke an STSL application teardown procedure. The retransmit timeout value shall be 100 ms for the first timeout, half the listen interval for the second timeout, and the listen interval for subsequent timeouts. If there is no listen interval or the listen interval is zero, then 100 ms shall be used for all timeout values.

There is no specific recovery mechanism at the STA_P if the SMK Message 3 is lost. This results in a timeout on the STA_I, as described in the preceding paragraph, and a subsequent reinitiation of the SMK Handshake. The STA_P shall allow reinitiation of the SMK Handshake at any point prior to receipt of SMK Message 4.

11.6.8.4 STKSA rekeying

Rekeying is always initiated by the STA_I. When needed, the STA_P sends an EAPOL request message to the STA_I to request rekeying. The STA_P shall wait a minimum of one half the IEEE 802.1X timeout after the STSL setup before initiating a PeerKey rekey procedure. To perform rekeying, there are two cases:

a) If SMK timer has not expired, the STAs initiate a 4-Way Handshake to create a new STK. The 4-Way Handshake is always initiated by the STA_I. In this case, the STA_P should not delete any existing STKSA prior to verifying Message 3 of the 4-Way Handshake with STA_I for this session.

b) If the SMK has expired, the STA_I shall not use an existing STKSA and shall start the SMK Handshake followed by a 4-Way Handshake to create new keys.

The format of the EAPOL-Key request message in case a) from STA_P to STA_I is as follows:

Request Message: STA_P → STA_I: EAPOL-Key(1,1,0,0,1,0,0,0, MIC, PMKID KDE)

The request message uses the following values for each of the EAPOL-Key frame fields:

Descriptor Type = N – see 11.6.2

Key Information:

Key Descriptor Version = 1 (ARC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA1-128) or 3 (NIST AES key wrap with AES-128-CMAC), in all other cases 0

Key Type = 1 (PTK)

SMK Message = 0

Install = 0

Key Ack = 0

Key MIC = 1

Secure = 1

Error = 0

Request = 1

Encrypted Key Data = 0

Reserved = 0

Key Length = 0

Key Replay Counter = request replay counter of peer STA

Key Nonce = 0

EAPOL-Key IV = 0

Key RSC = 0

Key MIC = MIC computed over the body of this EAPOL-Key frame

Key Data Length = Length of Key Data field in octets

Key Data = SMKID in SMKID KDE

11.6.8.5 Error Reporting

11.6.8.5.1 General

Error reporting messages are defined in this subclause and used to report errors whenever STAs or an AP detect an error during the SMK Handshake.

The AP, upon receiving the error messages defined in this subclause or upon generating the error messages defined in this subclause, should log the error. The STA, upon receipt of the error messages defined in this subclause, shall tear down the STSL with the other STA and clear all the PeerKey states.

The format of EAPOL-Key request message for reporting an error message is as follows:

Error Message: EAPOL-Key(1,1,0,0,0,1,0, 0, MIC, Error KDE, MAC Address KDE).

The request message uses the following values for each of the EAPOL-Key frame fields:

Descriptor Type = N – see 11.6.2

Key Information:

Key Descriptor Version = 1 (ARC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA1-128) or 3 (NIST AES key wrap with AES-128-CMAC), in all other cases 0

Key Type = 0 (Group/SMK)

SMK Message = 1 (SMK)

Install = 0

Key Ack = 0

Key MIC = 1

Secure = 1

Error = 1

Request = 1 when the message is going from the STA to an AP or 0 when the message is going from an AP to the STA

Encrypted Key Data = 0

Reserved = 0

Key Length = 0

Key Replay Counter = request EAPOL replay counter

Key Nonce = 0

EAPOL-Key IV = 0

Key RSC = 0

Key MIC = MIC computed over the body of this EAPOL-Key frame

Key Data Length = Length of Key Data field in octets

Key Data = Error KDE (different types defined in Table 11-8), MAC Address KDE

11.6.8.5.2 Error ERR_STA_NR

This error message is sent whenever an AP finds that the STA to which it needs to send a message is not reachable. In response to this error, the AP creates an Error KDE with error type ERR_STA_NR and sends the message back to the other STA involved in the handshake. The MAC address KDE contains the MAC address of the unreachable STA.

11.6.8.5.3 Error ERR_STA_NRSN

This error message is sent whenever the AP finds that the STA to which it needs to send the message does not have a secure RSNA connection. In response of this error, the AP creates an Error KDE with error type ERR_STA_NRSN and sends the message back to the STA from which it received the last message. The MAC address KDE contains the MAC address of the STA with which the AP does not have a secure RSNA connection.

11.6.8.5.4 Error ERR_CPHR_NS

This error message is sent whenever a STA finds that it does not support any of the cipher suites proposed by the other STA. In response to this error, the STA creates an Error KDE with error type ERR_CPHR_NS and sends the message back to the other STA. The MAC address KDE contains the MAC address of the other STA.

11.6.8.5.5 Error ERR_NO_STSL

This error message is sent whenever a STA finds that it does not have an existing STSL with the other STA. In response of this error, the STA creates an Error KDE with error type ERR_NO_STSL and sends the message back to the other STA. The MAC address KDE contains the MAC address of the other STA.

11.6.9 TDLS Peer Key security protocol

11.6.9.1 General

The TDLS Peer Key security protocol is executed between the two non-AP STAs that intend to establish an RSNA for direct-link communication. If any security method (pre-RSNA or RSNA) is enabled on the connection between a STA and the AP, the STA shall require that the TDLS Peer Key security protocol complete successfully before using a direct link. If no security method is enabled on the connection between a STA and the AP, the STA shall not use the TDLS Peer Key security protocol on the direct link. A STA may refuse to setup a TDLS link when the protection on the STA link to the AP is secured with a weak algorithm or when the link between the STA and the AP is not using any security.

11.6.9.2 TDLS Peer Key Handshake

The TDLS Peer Key (TPK) Handshake occurs as part of the TDLS direct-link setup procedure. The TPKSA is the result of the successful completion of the TDLS Peer Key Handshake protocol, which derives keys for providing confidentiality and data origin authentication.

In order to maintain TPK confidentiality, both the TDLS initiator STA and the TDLS responder STAs establish an RSNA with their common AP prior to executing the TDLS Peer Key Handshake. To meet this criterion, a STA may refuse to initiate the TDLS direct link if:

a) The AP does not include an RSNE in its Beacons and Probe Responses to advertise the availability of security;
b) The AP's RSNE indicates that WEP-40 (OUI 00-0F-AC:1) or WEP-104 (OUI 00-0F-AC:5) are enabled as either pairwise or group cipher suites; or
c) The AP's RSNE indicates that Use group cipher suite (00-0F-AC:0) is used as the pairwise cipher suite.

Violation of any of these cases would cause the TDLS Peer Key Handshake to leak the TPK.

The TDLS initiator STA and the TDLS responder STA perform the following exchange to setup a TPK:

TDLS PMK Handshake Message 1: TDLS initiator STA → TDLS responder STA:
Link Identifier element, RSNE, Timeout Interval element, FTE
TDLS PMK Handshake Message 2: TDLS responder STA → TDLS initiator STA:
Link Identifier element, RSNE, Timeout Interval element, FTE
TDLS PMK Handshake Message 3: TDLS initiator STA → TDLS responder STA:
Link Identifier element, RSNE, Timeout Interval element, FTE

where

The TDLS initiator STA Address field of the Link Identifier element is the MAC address of the TDLS initiator STA
The TDLS responder STA Address field of the Link Identifier element is the MAC address of the TDLS responder STA
The PairwiseCipherSuite field of the RSNE identifies the cipher suite used to protect the data frames sent over the direct link
The AKM suite list of the RSNE identifies which Authentication Method was used
The TimeoutIntervalType field of the Timeout Interval element is the key lifetime
The SNonce field of the FTE is a 256 bit value randomly generated by the TDLS initiator STA
The ANonce field of the FTE is a 256 bit value randomly generated by the TDLS responder STA (set to 0 in message 1)
The MIC field of the FTE is 0 for message 1 and computed as described in 11.6.9.4.3 and 11.6.9.4.4 for messages 2 and 3 respectively

The TDLS PMK Handshake Message 1 shall be transmitted in the TDLS Setup Request frame.

TDLS PMK Handshake Message 2 shall be transmitted in the TDLS Setup Response frame.

TDLS PMK Handshake Message 3 shall be transmitted in the TDLS Setup Confirm frame.

The TPK shall be derived as follows:

TPK-Key-Input = SHA-256(min (SNonce, ANonce) || max (SNonce, ANonce))

TPK = KDF-N_KEY(TPK-Key-Input, "TDLS PMK", min (MAC_I, MAC_R)
|| max (MAC_I, MAC_R) || BSSID)

where

N_KEY = TK_bits + 128. TK_bits is cipher-suite specific and specified in Table 11-4
KDF-N_KEY is the key derivation function defined in 11.6.1.7.2
MAC_I and MAC_R are the MAC addresses of the TDLS initiator STA and the TDLS responder STA, respectively
SNonce and ANonce are the nonces generated by the TDLS initiator STA and TDLS responder STA, respectively, for this instance of the TPK handshake. The BSSID is set to the BSSID of the current association of the TDLS initiator STA.

Each TPK has two component keys—TPK-KCK and TPK-TK, defined as follows:

The Key Confirmation Key (KCK) shall be computed as the first 128 bits (bits 0–127) of the TPK

TPK-KCK = L(TPK, 0, 128)

where

L(-) is defined in 11.6.1.

The KCK is used to provide data origin authenticity in TDLS Setup Response and TDLS Setup Confirm messages.

The Temporal keys (TK) shall be computed as the remaining bits (for CCMP, the second 128 bits, i.e., bits 128–255) of the TPK

TPK-TK = L(TPK, 128, N_KEY – 128)

The TPK-TK is used to provide confidentiality for direct-link data.

The temporal key is configured into the STA by the SME through the use of the MLME-SETKEYS.request primitive.

11.6.9.3 TDLS Peer Key Handshake security assumptions

The security of the TDLS PMK Handshake depends on the following:

a) The TDLS initiator STA and the TDLS peer STA each have an RSNA established with the AP that is being used for TDLS Setup.
b) The AP does not expose the nonces exchanged by the TDLS initiator STA and the TDLS responder STA to any external party.
c) The AP does not use these nonces to derive the TPK and attack the TDLS direct-link instance.
d) TDLS message security (encryption and integrity computations) processing at the AP is protected from illegal eavesdropping, alterations, insertions and substitutions.
e) The TDLS initiator STA and TDLS responder STAs do not expose SNonce, ANonce, or the derived key to a third party.
f) The TDLS initiator STA and the TDLS peer STA are associated to the same AP.

11.6.9.4 TDLS Peer Key (TPK) Security Protocol Handshake messages

11.6.9.4.1 Overview

The TDLS Peer Key Handshake consists of three messages. Each message is comprised of a number of elements, and is included in the TDLS Setup Request, TDLS Setup Response, and TDLS Setup Confirm.

In an RSN, these handshake messages serve to provide a session identifier, are identified by the nonces, and are used as association instance identifiers. These nonces are chosen randomly or pseudo randomly, and are used to generate the TPK.

11.6.9.4.2 TPK Handshake Message 1

If the TDLS initiator STA has security enabled on the link with the AP, it shall add an RSNE, FTE, and Timeout Interval element to its TDLS Setup Request frame. The elements shall be formatted as follows:

> The RSNE, if present, shall be set as follows:
>
> > Version shall be set to 1.
> >
> > The pairwise cipher suite list field indicating the pairwise cipher suites the TDLS initiator STA is willing to use with the TPKSA. WEP-40, WEP-104, and TKIP shall not be included in this list.
> >
> > The group cipher suite shall be set to 00-0F-AC:7.
> >
> > The AKM suite count field shall be set to 1.
> >
> > The AKM suite list field shall be set to TPK Handshake (00-0F-AC:7).
> >
> > The Capabilities field shall set the ‘No Pairwise’ subfield to 0 and ‘Peer Key Enabled’ subfield to 1.
> >
> > PMKID-Count subfield, if present, shall be set to 0.
> >
> > PMKID list shall not be present.
> >
> > The Group Management Cipher Suite subfield, if present, shall be set to 00-0F-AC:7.
>
> The Timeout Interval element indicates the lifetime of the TPKSA. The Lifetime Interval Type shall be set to ‘2’ (Key Lifetime Interval). The minimum lifetime shall be 300 seconds.
>
> The FTE shall be set as follows:
>
> > SNonce shall be set to a value chosen randomly by the TDLS initiator STA, following the recommendations of 11.6.5.
> >
> > All other fields shall be set to 0.

The TDLS initiator STA sends Message 1 to the TDLS responder STA.

On reception of Message 1, the TDLS responder STA checks whether the RSNE is present.

> If the TDLS responder STA does not have security enabled on the link with the AP, it shall reject the request with status code 5 (“Security disabled”).
>
> If the TDLS responder STA has security enabled on the link with the AP, it checks whether the request includes an RSNE and FTE. If not, the TDLS responder STA shall reject the request with status code 38 (“The request has not been successful as one or more parameters have invalid values”).
>
> If the version field of the RSNE is 0, then the TDLS responder STA shall reject the request with status code 44 (“Unsupported RSNE version”).
>
> Otherwise, the TDLS responder STA processes the message as follows:
>
> > If the contents of the RSNE do not indicate AKM of TPK Handshake (suite type 00-0F-AC:7), the TDLS responder STA shall reject the request with status code 43 (“Invalid AKMP”).

If none of the pairwise cipher suites are acceptable, or pairwise ciphers include WEP-40, WEP-104, or TKIP, then the TDLS responder STA shall reject the TDLS Setup Request with status code 42 ("Invalid pairwise cipher").

If the RSN Capabilities field has not set the subfields according to the described rules for this message, then the TDLS responder STA rejects with status code 45 ("Invalid RSNE capabilities").

If the suggested lifetime is unacceptable or below the default value, the TDLS responder STA shall reject the TDLS Setup Request with status code 6 ("Unacceptable lifetime").

If the contents of the FTE are not as per specified for this message, then the TDLS responder STA shall reject the TDLS Setup Request with status code 55 ("Invalid FTE").

The TDLS responder STA shall ignore the remaining fields in the RSNE, FTE, and Timeout Interval element.

Otherwise, the TDLS responder STA shall respond as specified in 10.22.4.

11.6.9.4.3 TPK Handshake Message 2

If the TDLS responder STA validates the TPK Handshake Message 1 for this TDLS instance, the TDLS responder STA may respond with TPK Handshake Message 2. To do so, the TDLS responder STA shall add an RSNE, FTE, and Timeout Interval element to its TDLS Setup Response frame. The elements shall be formatted as follows:

The RSNE shall include the following:

Include a pairwise cipher suite from one of those presented in RSNE of Message 1 of this sequence in the pairwise cipher suite list, and set the pairwise cipher suite count to 1.

The version number shall be the minimum of the maximum version supported by the TDLS responder STA and the version number received in the RSNE of Message 1.

All other RSNE fields shall be same as those received in Message 1.

The Timeout Interval element shall be the same as that received in the TPK Handshake Message 1.

The FTE shall include the following:

ANonce shall be set to a value chosen randomly by the TDLS responder STA, following the recommendations of 11.6.5.

SNonce shall be same as that received in Message 1 of this sequence

The MIC shall be calculated on the concatenation, in the following order, of:

TDLS initiator STA MAC address (6 octets)

TDLS responder STA MAC address (6 octets)

Transaction Sequence number (1 octet) which shall be set to the value 2

Link Identifier element

RSNE

Timeout Interval element

FTE, with the MIC field of the FTE set to 0.

The MIC shall be calculated using the TPK-KCK and the AES-128-CMAC algorithm. The output of the AES-128-CMAC shall be 128 bits.

All other fields shall be set to 0.

The TDLS responder STA shall use the MLME-SETKEYS.request primitive to configure the Temporal Key into its STA prior to sending Message 2.

The TDLS responder STA sends Message 2 to the TDLS initiator STA. The TDLS initiator STA shall process Message 2 as follows:

If the TDLS initiator STA Address and TDLS responder STA Address of the Link Identifier element do not match those for an outstanding TDLS Setup Request, the TDLS initiator STA shall silently discard the received TDLS Setup Response frame.

If the SNonce field of the FTE does not match that of an outstanding request to the TDLS responder STA, then the TDLS initiator STA shall silently discard the received TDLS Setup Response frame.

Otherwise, the TDLS initiator STA shall compute the TPK and then validate the MIC in the FTE as specified in MIC calculation procedure for TPK Handshake Message 2. If invalid, the TDLS initiator STA shall discard the message.

If the version of the RSNE is 0 or is greater than the version of the RSNE sent in Message 1, then the TDLS initiator STA shall reject the response with status code 44 (“Unsupported RSNE version”). Otherwise,

If the contents of the RSNE, with the exception of the pairwise cipher suite count and pairwise cipher suite list are not the same as those sent by the TDLS initiator STA in Message 1 of this sequence, then the TDLS initiator STA shall reject the response with status code 72 (“Invalid contents of RSNE”).

If the pairwise cipher suite count is other than 1, then the TDLS initiator STA shall reject the response with status code 42 (“Invalid pairwise cipher”).

If the selected pairwise cipher suite was not included in the Initiator’s request, then the TDLS initiator STA shall reject the TDLS Setup Response with status code 42 (“Invalid pairwise cipher”).

If the Timeout Interval element is not the same as that sent in Message 1, the TDLS initiator STA shall reject the TDLS Setup Response with status code 6 (“Unacceptable lifetime”).

If the BSSID in the Link Identifier element is different from the one sent in Message 1, then the TDLS initiator STA shall reject the response with status code 7 (“Not in same BSS”).

If the TDLS initiator STA validates TDLS Message 2, the TDLS initiator STA shall create a TPKSA and respond with Message 3 as defined in 10.22.4. The TDLS initiator STA shall use the MLME-SETKEYS.request primitive to configure the Temporal Key into its STA prior to sending Message 3.

11.6.9.4.4 TPK Handshake Message 3

If the TDLS initiator STA responds to Message 2 for this TDLS instance, the TDLS initiator STA shall add an RSNE, FTE, and Timeout Interval element to its TDLS Setup Confirm frame. The elements shall be formatted as follows:

The RSNE shall be the same as the RSNE received in Message 2.

The Timeout Interval element shall be the same as that received in the TPK Handshake message 2.

With the exception of the MIC field, the contents of the FTE shall be the same as the FTE received in Message 2.

The MIC shall be calculated on the concatenation, in the following order, of:

TDLS initiator STA MAC address (6 octets)

TDLS responder STA MAC address (6 octets)

Transaction Sequence number (1 octet), which shall be set to the value 3

Link Identifier element

RSNE

Timeout Interval element

FTE, with the MIC field of the FTE set to 0.

The MIC shall be calculated using the TPK-KCK and the AES-128-CMAC algorithm. The output of the AES-128-CMAC shall be 128 bits.

All other fields shall be set to 0.

The TDLS initiator STA sends Message 3 to the TDLS responder STA. The TDLS responder STA shall process Message 3 as follows:

If the Source and Destination Addresses of the Link Identifier element do not match those for an outstanding TDLS Setup Request, the TDLS responder STA shall discard the message.

If the ANonce and SNonce fields of the FTE do not match that of an outstanding request to the TDLS initiator STA, then the TDLS responder STA shall discard the message.

Otherwise, the TDLS responder STA shall validate the MIC in the FTE as specified in the MIC calculation procedure for TPK Handshake Message 3. If invalid, the TDLS responder STA shall discard the message.

The TDLS responder STA shall discard the message, the TDLS responder STA shall abandon the TPK Handshake identified by the <ANonce, SNonce> combination, and delete existing TPK Handshake Key state for this sequence if any of the following checks fail:

The contents of the RSNE are not the same as that sent by the TDLS responder STA in Message 2

The Timeout Interval element is not the same as that sent in Message 2

The BSSID from the Link Identifier element is not the same as that sent in Message 2

On successful processing of Message 3, the TPK Handshake is considered successful.

The TPKSA shall be deleted by the TDLS responder STA if it does not receive a valid TPK Handshake Message 3 from the TDLS Initiator STA within dot11TDLSResponseTimeout.

11.6.9.5 Supplicant state machine procedures

The following list summarizes the procedures used by the Supplicant state machine:

- **STADisconnect** – The Supplicant invokes this procedure to disassociate and deauthenticate its STA from the AP.
- **MIC**(*x*) – The Supplicant invokes this procedure to compute a MIC of the data *x*.
- **CheckMIC**() – The Supplicant invokes this procedure to verify a MIC computed by the MIC() function.
- **StaProcessEAPOL-Key –** The Supplicant invokes this procedure to process a received EAPOL-Key frame. The pseudo-code for this procedure is as follows:

  StaProcessEAPOL-Key (*S*, *M*, *A*, *I*, *K*, *RSC*, *ANonce*, RSC, *MIC*, RSNE, *GTK*[*N]*, *IGTK*[*M*], IPN)

```
TPTK ← PTK
TSNonce ← 0
PRSC ← 0
UpdatePTK ← 0
State ← UNKNOWN
if M = 1 then
    if Check MIC(PTK, EAPOL-Key frame) fails then
        State ← FAILED
    else
        State ← MICOK
    endif
endif
if K = P then
    if State ≠ FAILED then
        if PSK exists then – PSK is a preshared key
```

```
                PMK ← PSK
            else
                PMK ← L(MSK, 0, 256)
            endif
            TSNonce ← SNonce
            if ANonce ≠ PreANonce then
                TPTK ← Calc PTK(PMK, ANonce, TSNonce)
                PreANonce ← ANonce
            endif
            if State = MICOK then
                PTK ← TPTK
                UpdatePTK ← I
                if UpdatePTK = 1 then
                    if no GTK then
                        PRSC ← RSC
                    endif
                    if MLME-SETKEYS.request(0, TRUE, PRSC, PTK) fails then
                        invoke MLME-DEAUTHENTICATE.request
                    endif
                    MLME-SETPROTECTION.request(TA, Rx)
                endif
                if GTK then
                    if (GTK[N] ← Decrypt GTK) succeeds then
                        if MLME-SETKEYS.request(N, 0, RSC, GTK[N]) fails then
                            invoke MLME-DEAUTHENTICATE.request
                        endif
                    else
                        State ← FAILED
                    endif
                endif
                if IGTK then
                    if (IGTK[M] ← Decrypt IGTK) succeeds then
                        if MLME-SETKEYS.request(M, 0, IPN, IGTK[M]) fails then
                            invoke MLME-DEAUTHENTICATE.request
                        endif
                    else
                        State ← FAILED
                    endif
                endif
            endif
        endif
else if KeyData = GTK then
    if State = MICOK then
        if (GTK[N] ← Decrypt GTK) succeeds then
            if MLME-SETKEYS.request(N, T, RSC, GTK[N]) fails then
                invoke MLME-DEAUTHENTICATE.request
            endif
        else
            State ← FAILED
        endif
        if (IGTK[M] ← Decrypt IGTK) succeeds then
            if MLME-SETKEYS.request(M, T, IPN, IGTK[M]) fails then
                invoke MLME-DEAUTHENTICATE request
            endif
        else
            State ← FAILED
        endif
    else
        State ← FAILED
    endif
endif
if A = 1 && State ≠ Failed then
    Send EAPOL-Key(0,1,0,0,K,0,0,TSNonce,MIC(TPTK),RSNE)
endif
if UpdatePTK = 1 then
    MLME-SETPROTECTION.request(TA, Tx_Rx)
endif
if State = MICOK && S = 1 then
    MLME-SETPROTECTION.request(TA, Tx_Rx)
    if IBSS then
        keycount++
        if keycount = 2 then
            802.1X::portValid← TRUE
        endif
    else
        802.1X::portValid← TRUE
    endif
endif
```

Here UNKNOWN, MICOK, and FAILED are values of the variable State used in the Supplicant pseudo-code. State is used to decide how to do the key processing. MICOK is set to 1 when the MIC of the EAPOL-Key has been checked and is valid. FAILED is used when a failure has occurred in processing the EAPOL-Key frame. UNKNOWN is the initial value of the variable State.

When processing 4-Way Handshake Message 3, the GTK and IGTK are decrypted from the EAPOL-Key frame and installed. The PTK shall be installed before the GTK and IGTK.

The Key Replay Counter field used by the Supplicant for EAPOL-Key frames that are sent in response to a received EAPOL-Key frame shall be the received Key Replay Counter field. Invalid EAPOL-Key frames such as invalid MIC, GTK without a MIC, etc., shall be ignored.

NOTE 1—TPTK is used to stop attackers changing the PTK on the Supplicant by sending the first message of the 4-Way Handshake. An attacker can still affect the 4-Way Handshake while the 4-Way Handshake is being carried out.

NOTE 2—The PMK is supplied by the authentication method used with IEEE Std 802.1X-2004 if preshared mode is not used.

NOTE 3—A PTK is configured into the encryption/integrity engine depending on the Tx/Rx bit, but if configured, is always a transmit key. A GTK is configured into the encryption/integrity engine independent of the state of the Tx/Rx bit, but whether the GTK is used as a transmit key is dependent on the state of the Tx/Rx bit.

- **CalcGTK**(x) – Generates the GTK.
- **DecryptGTK(x)** – Decrypt the GTK from the EAPOL-Key frame.
- **DecryptIGTK(x)** – Decrypt the IGTK from the EAPOL-Key frame.

11.6.9.6 Supplicant PeerKey state machine states

Figure 11-42 depicts the PeerKey Handshake Supplicant key management state machine. The following list summarizes the states the Supplicant state machine uses to support the PeerKey Handshake:

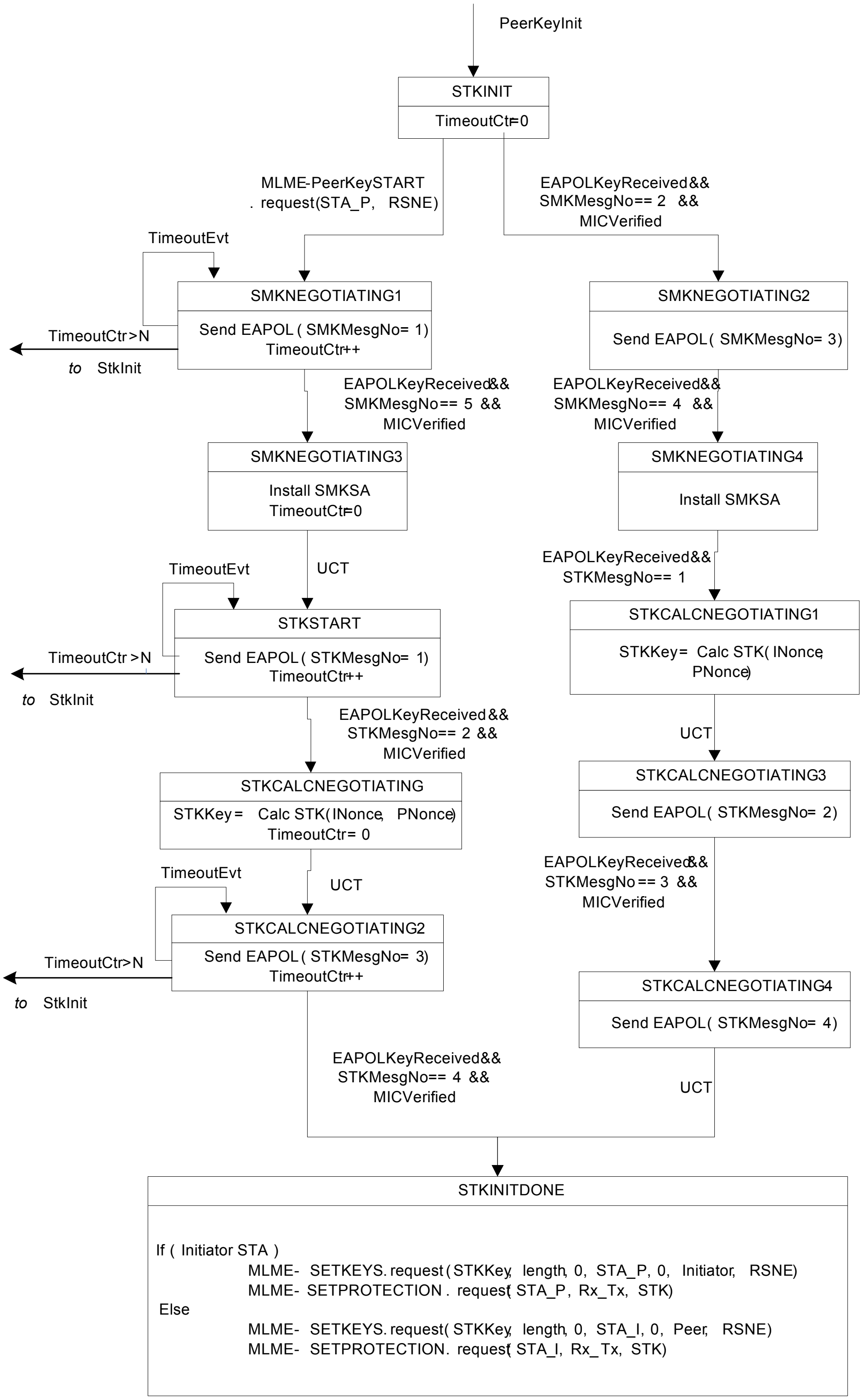


Figure 11-42—PeerKey Handshake Supplicant key management state machine

- **STKINIT:** This state is the idle state and is entered when the IEEE 802.1X Supplicant completes successful Authentication.
- **SMKNEGOTIATING1:** This state is entered when the MLME-STKSTART.request primitive is received for the SMK Handshake by the initiator STA.
- **SMKNEGOTIATING2:** This state is entered when the first EAPOL-Key frame of the SMK Handshake is received by the peer STA.
- **SMKNEGOTIATING3:** This state is entered when the fifth EAPOL-Key frame of the SMK Handshake is received by the initiator STA.
- **SMKNEGOTIATING4:** This state is entered when the fourth EAPOL-Key frame of the SMK Handshake is received by the peer STA.
- **STKSTART:** Once the SMKSA is created, the initiator STA enters this state. This is the start of 4-Way STK Handshake.
- **STKCALCNEGOTIATING:** This state is entered when the second EAPOL-Key frame of the 4-Way STK Handshake is received by the initiator STA and the MIC is verified.
- **STKCALCNEGOTIATING1:** This state is entered when the first EAPOL-Key frame of the 4-Way STK Handshake is received by the peer STA and the MIC is verified.
- **STKCALCNEGOTIATING2:** This state is entered unconditionally by the initiator STA.
- **STKCALCNEGOTIATING3:** This state is entered unconditionally by the peer STA.
- **STKCALCNEGOTIATING4:** This state is entered when the third EAPOL-Key frame of the 4-Way STK Handshake is received by the peer STA and the MIC is verified.
- **STKINITDONE:** This state is entered by the initiator STA when the fourth EAPOL-Key frame of the 4-Way STK Handshake is received. This state is entered by the peer STA when the fourth EAPOL-Key frame of the 4-Way STK Handshake is sent.

11.6.9.7 Supplicant PeerKey state machine variables

The following list summarizes the variables used by the Supplicant state machine:

- *PeerKeyInit* **–** This variable is used to initialize the PeerKey state machine.
- *TimeoutEvt* **–** This variable is set to TRUE if the EAPOL-Key frame sent fails to obtain a response from the Supplicant. The variable may be set by management action or set by the operation of a timeout while in the different states.
- *TimeoutCtr* **–** This variable maintains the count of EAPOL-Key receive timeouts. It is incremented each time a timeout occurs on EAPOL-Key receive event and is initialized to 0. The Key Replay Counter field value for the EAPOL-Key frame shall be incremented on each transmission of the EAPOL-Key frame.
- *MICVerified* **–** This variable is set to TRUE if the MIC on the received EAPOL-Key frame is verified and is correct. Any EAPOL-Key frames with an invalid MIC are dropped and ignored.
- *SMKMesgNo* **–** This variable indicates SMK Handshake EAPOL-Key frame types. Details for each message type (1-5) are provided in 11.6.8.
- *STKMesgNo* **–** This variable indicates 4-Way STK Handshake EAPOL-Key frame types. Details for each message type (1-4) are provided in 11.6.6.
- *STA_P* **–** This variable indicates the MAC address of the peer STA participating in the PeerKey Handshake.
- *STA_I* **–** This variable indicates the MAC address of the initiator STA participating in the PeerKey Handshake.
- *STKKey* **–** The STK Key generated as a result of the 4-Way STK Handshake.
- *EAPOLKeyReceived* – The Supplicant sets this variable to TRUE when it receives an EAPOL-Key frame.

11.6.10 RSNA Supplicant key management state machine

11.6.10.1 General

The Supplicant shall reinitialize the Supplicant state machine whenever its system initializes. A Supplicant enters the AUTHENTICATION state on an event from the MAC that requests another STA to be authenticated. A Supplicant enters the STAKEYSTART state on receiving an EAPOL-Key frame from the Authenticator. If the MIC or any of the EAPOL-Key frames fails, the Supplicant silently discards the frame. Figure 11-43 depicts the RSNA Supplicant state machine.

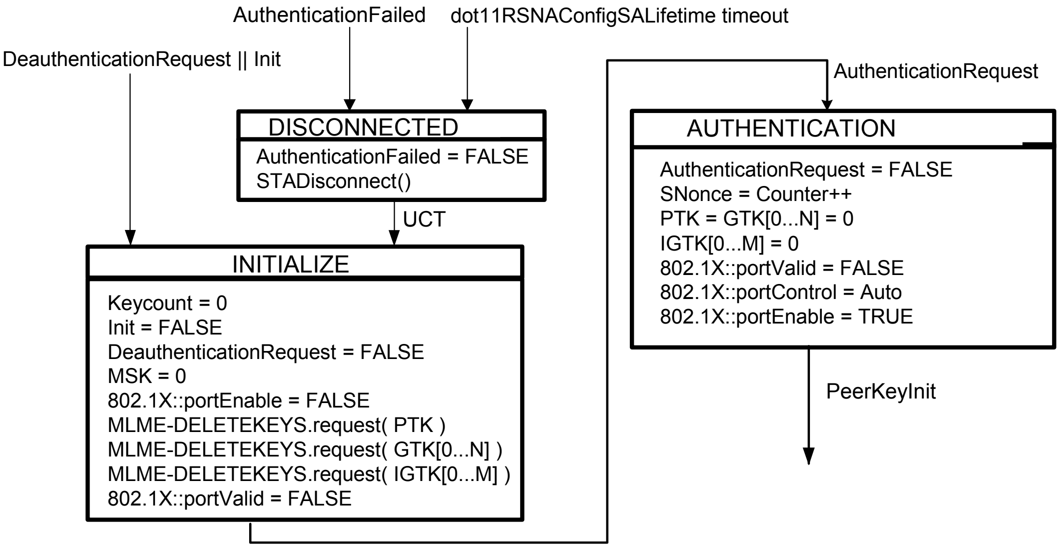

AuthenticationFailed
dot11RSNAConfigSALifetime timeout
DeauthenticationRequest || Init
AuthenticationRequest
DISCONNECTED
AuthenticationFailed = FALSE
STADisconnect()
UCT
AUTHENTICATION
AuthenticationRequest = FALSE
SNonce = Counter++
PTK = GTK[0...N] = 0
IGTK[0...M] = 0
802.1X::portValid = FALSE
802.1X::portControl = Auto
802.1X::portEnable = TRUE
INITIALIZE
Keycount = 0
Init = FALSE
DeauthenticationRequest = FALSE
MSK = 0
802.1X::portEnable = FALSE
MLME-DELETEKEYS.request(PTK)
MLME-DELETEKEYS.request(GTK[0...N])
MLME-DELETEKEYS.request(IGTK[0...M])
802.1X::portValid = FALSE
PeerKeyInit

Figure 11-43—RSNA Supplicant key management state machine

Unconditional transfer (UCT) means the event triggers an immediate transition.

This state machine does not use timeouts or retries. The IEEE 802.1X state machine has timeouts that recover from authentication failures, etc.

In order to authenticate an Authenticator, the management entity sends an authentication request event. This might be before or after the STA associates to the AP. In an IBSS environment, the event is generated when a Probe Response frame is received.

11.6.10.2 Supplicant state machine states

The following list summarizes the states of the Supplicant state machine:

- **AUTHENTICATION**: A Supplicant enters this state when it sends an IEEE 802.1X AuthenticationRequest to authenticate to an SSID.
- **DISCONNECTED**: A Supplicant enters this state when IEEE 802.1X authentication fails. The Supplicant executes StaDisconnect and enters the INITIALIZE state.
- **INITIALIZE**: A Supplicant enters this state from the DISCONNECTED state, when it receives Disassociation or Deauthentication messages or when the STA initializes, causing the Supplicant to initialize the key state variables.
- **STAKEYSTART:** A Supplicant enters this state when it receives an EAPOL-Key frame. All the information to process the EAPOL-Key frame is in the message and is described in the StaProcessEAPOL-Key procedure.

11.6.10.3 Supplicant state machine variables

The following list summarizes the variables used by the Supplicant state machine:

- *DeauthenticationRequest* – The Supplicant sets this variable to TRUE if the Supplicant's STA reports it has received Disassociation or Deauthentication messages.
- *AuthenticationRequest* – The Supplicant sets this variable to TRUE if its STA's IEEE 802.11 management entity reports that an SSID is to be authenticated. This might be on association or at other times.
- *AuthenticationFailed* – The Supplicant sets this variable to TRUE if the IEEE 802.1X authentication failed. The Supplicant uses the MLME-DISASSOCIATE.request primitive to cause its STA to disassociate from the Authenticator's STA.
- *EAPOLKeyReceived* – The Supplicant sets this variable to TRUE when it receives an EAPOL-Key frame.
- *IntegrityFailed* **–** The Supplicant sets this variable to TRUE when its STA reports that a fatal data integrity error (e.g., Michael failure) has occurred.

  NOTE—A Michael failure is not the same as MICVerified because IntegrityFailed is generated if the Michael integrity check fails; MICVerified is generated from validating the EAPOL-Key integrity check. Note also the STA does not generate this event for ciphers other than TKIP because countermeasures are not required.

- *MICVerified* **–** The Supplicant sets this variable to TRUE if the MIC on the received EAPOL-Key frame verifies as correct. The Supplicant silently discards any EAPOL-Key frame received with an invalid MIC.
- *Counter* **–** The Supplicant uses this variable as a global counter used for generating nonces.
- *SNonce* **–** This variable represents the Supplicant's nonce.
- *PTK* **–** This variable represents the current PTK.
- *TPTK* **–** This variable represents the current PTK until Message 3 of the 4-Way Handshake arrives and is verified.
- *GTK[]* **–** This variable represents the current GTKs for each group key index.
- *IGTK[]* – This variable represents the current IGTKs for each group management key index.
- *PMK* **–** This variable represents the current PMK.
- *keycount* – This variable is used in IBSS mode to decide when all the keys have been delivered and an IBSS link is secure.
- *802.1X::XXX* **–** This variable denotes another IEEE 802.1X state variable *XXX* not specified in this standard.

11.6.11 RSNA Authenticator key management state machine

11.6.11.1 General

There is one state diagram for the Authenticator. In an ESS, the Authenticator is always on the AP; and in an IBSS environment, the Authenticator is on every STA.

The state diagram shown in parts in Figure 11-44 to Figure 11-47 consists of the following states:

a) The AUTHENTICATION, AUTHENTICATION2, INITPMK, INITPSK, PTKSTART, PTKCALCNEGOTIATING, PTKCALCNEGOTIATING2, PTKINITNEGOTIATING, PTKINITDONE, DISCONNECT, DISCONNECTED, and INITIALIZE states. These states handle the initialization, 4-Way Handshake, tear-down, and general clean-up. These states are per associated STA.

b) The IDLE, REKEYNEGOTIATING, KEYERROR, and REKEYESTABLISHED states. These states handle the transfer of the GTK to the associated client. These states are per associated STA.

c) The GTK_INIT, SETKEYS, and SETKEYSDONE states. These states change the GTK when required, trigger all the PTK group key state machines, and update the IEEE 802.11 MAC in the Authenticator's AP when all STAs have the updated GTK. These states are global to the Authenticator.

Because there are two GTKs, responsibility for updating these keys is given to the group key state machine (see Figure 11-46). In other words, this state machine determines which GTK is in use at any time.

When a second STA associates, the group key state machine is already initialized, and a GTK is already available and in use.

When the GTK is to be updated the variable GTKReKey is set to 1. The SETKEYS state updates the GTK and triggers all the PTK group key state machines that currently exist—one per associated STA. Each PTK group key state machine sends the GTK to its STA. When all the STAs have received the GTK (or failed to receive the key), the SETKEYSDONE state is executed which updates the APs encryption/integrity engine with the new key.

Both the PTK state machine and the PTK group key state machine use received EAPOL-Key frames as an event to change states. The PTK state machine only uses EAPOL-Key frames with the Key Type field equal to Pairwise, and the PTK group key state machine only uses EAPOL-Key frames with the Key Type field equal to Group.

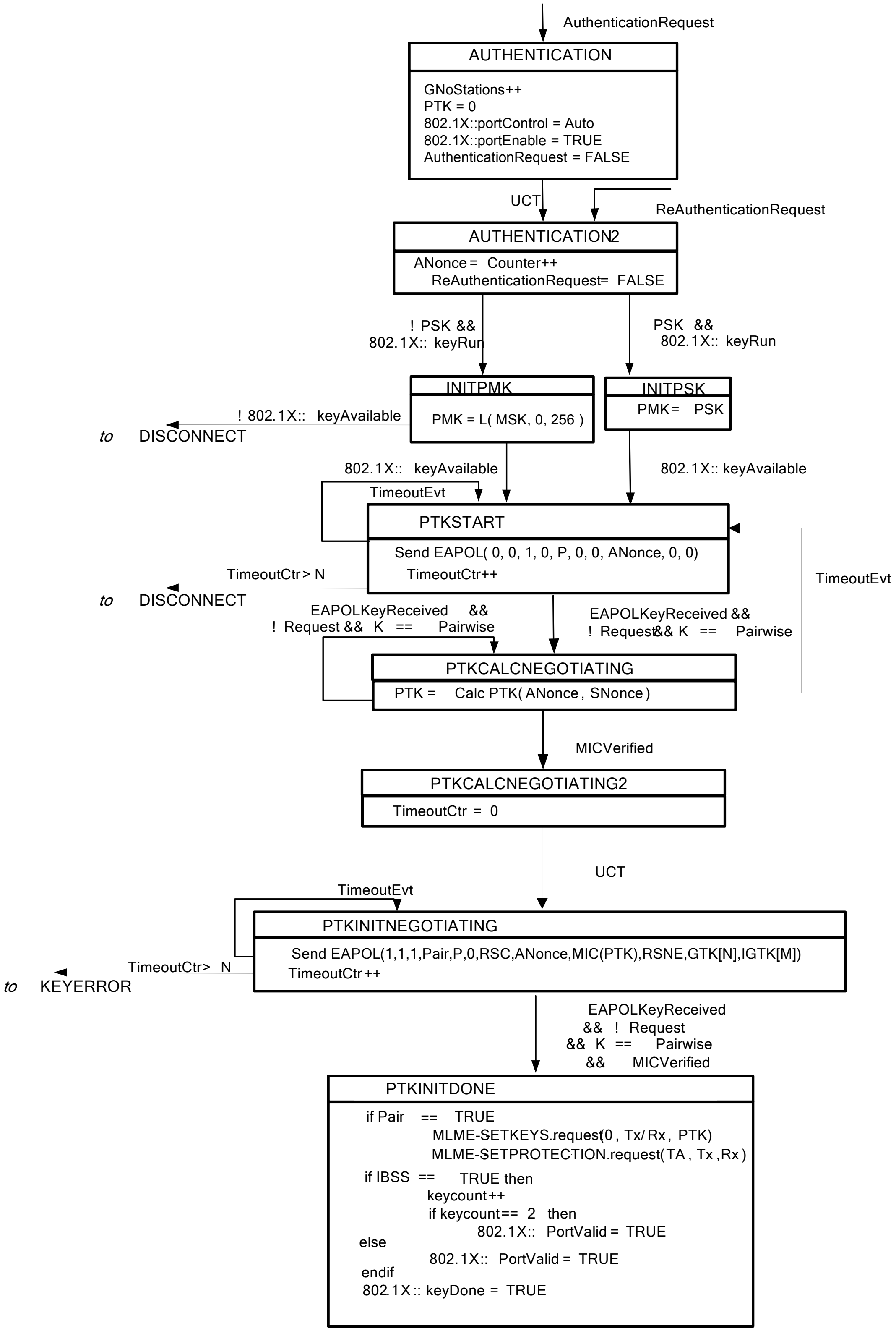


Figure 11-44—Authenticator state machines, part 1

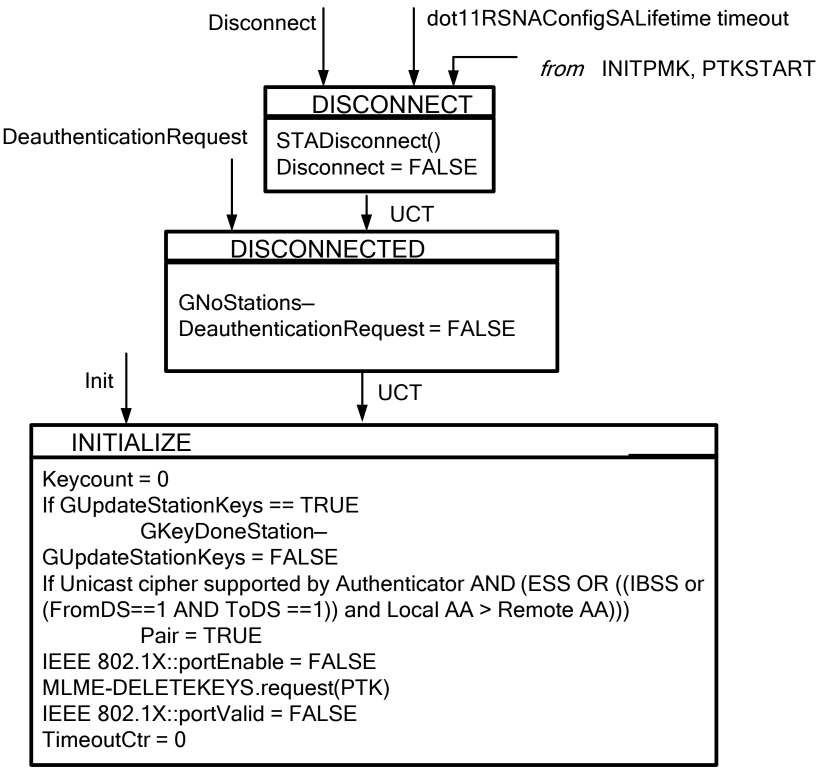

Disconnect
dot11RSNAConfigSALifetime timeout
from INITPMK, PTKSTART
DISCONNECT
STADisconnect()
Disconnect = FALSE
DeauthenticationRequest
UCT
DISCONNECTED
GNoStations--
DeauthenticationRequest = FALSE
Init
UCT
INITIALIZE
Keycount = 0
If GUpdateStationKeys == TRUE
GKeyDoneStation--
GUpdateStationKeys = FALSE
If Unicast cipher supported by Authenticator AND (ESS OR ((IBSS or (FromDS==1 AND ToDS ==1)) and Local AA > Remote AA)))
Pair = TRUE
IEEE 802.1X::portEnable = FALSE
MLME-DELETEKEYS.request(PTK)
IEEE 802.1X::portValid = FALSE
TimeoutCtr = 0

Figure 11-45—Authenticator state machines, part 2

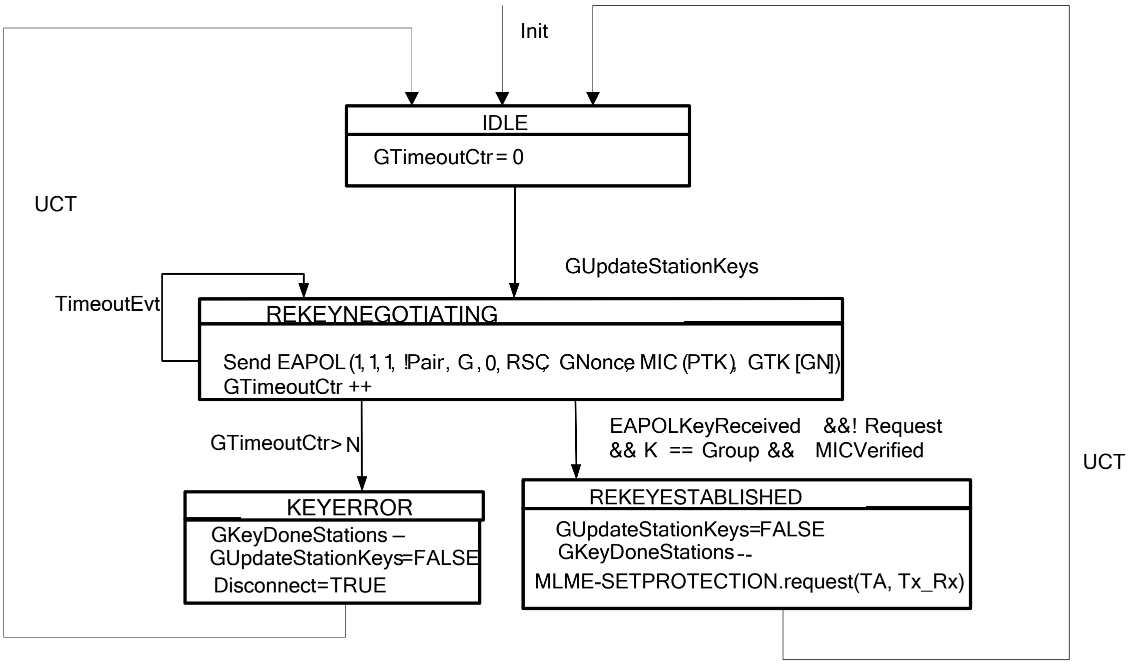


Figure 11-46—Authenticator state machines, part 3

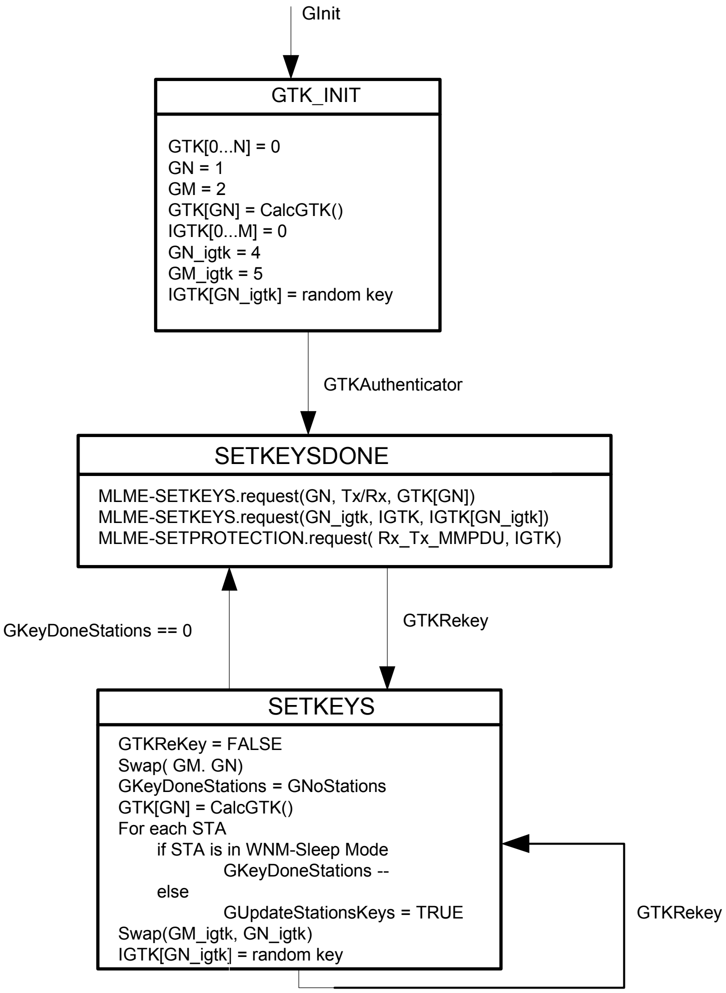

GInit
GTK_INIT
GTK[0...N] = 0
GN = 1
GM = 2
GTK[GN] = CalcGTK()
IGTK[0...M] = 0
GN_igtk = 4
GM_igtk = 5
IGTK[GN_igtk] = random key
GTKAuthenticator
SETKEYSDONE
MLME-SETKEYS.request(GN, Tx/Rx, GTK[GN])
MLME-SETKEYS.request(GN_igtk, IGTK, IGTK[GN_igtk])
MLME-SETPROTECTION.request(Rx_Tx_MMPDU, IGTK)
GKeyDoneStations == 0
GTKRekey
SETKEYS
GTKReKey = FALSE
Swap(GM. GN)
GKeyDoneStations = GNoStations
GTK[GN] = CalcGTK()
For each STA
if STA is in WNM-Sleep Mode
GKeyDoneStations --
else
GUpdateStationsKeys = TRUE
Swap(GM_igtk, GN_igtk)
IGTK[GN_igtk] = random key
GTKRekey

Figure 11-47—Authenticator state machines, part 4

11.6.11.2 Authenticator state machine states

11.6.11.2.1 Authenticator state machine: 4-Way Handshake (per STA)

The following list summarizes the states the Authenticator state machine uses to support the 4-Way Handshake:

— **AUTHENTICATION**: This state is entered when an AuthenticationRequest is sent from the management entity to authenticate a BSSID.

— **AUTHENTICATION2**: This state is entered from the AUTHENTICATION state or from the PTKINITDONE state.

— **DISCONNECT**: This state is entered if an EAPOL-Key frame is received and fails its MIC check. It sends a Deauthentication message to the STA and enters the INITIALIZE state.

— **DISCONNECTED**: This state is entered when Disassociation or Deauthentication messages are received.

— **INITIALIZE**: This state is entered from the DISCONNECTED state, when a deauthentication request event occurs, or when the STA initializes. The state initializes the key state variables.

- **INITPMK**: This state is entered when the IEEE 802.1X backend AS completes successfully. If a PMK is supplied, it goes to the PTKSTART state; otherwise, it goes to the DISCONNECTED state.
- **INITPSK:** This state is entered when a PSK is configured.
- **PTKCALCNEGOTIATING:** This state is entered when the second EAPOL-Key frame for the 4-Way Handshake is received with the key type of Pairwise.
- **PTKCALNEGOTIATING2:** This state is entered when the MIC for the second EAPOL-Key frame of the 4-Way Handshake is verified.
- **PTKINITNEGOTIATING:** This state is entered when the MIC for the second EAPOL-Key frame for the 4-Way Handshake is verified. When Message 3 of the 4-Way Handshake is sent in state PTKINITNEGOTIATING, the encrypted GTK shall be sent at the end of the data field, and the GTK length is put in the GTK Length field.
- **PTKINITDONE:** This state is entered when the last EAPOL-Key frame for the 4-Way Handshake is received with the key type of Pairwise. This state may call SetPTK; if this call fails, the AP should detect and recover from the situation, for example, by doing a disconnect event for this association.
- **PTKSTART:** This state is entered from INITPMK or INITPSK to start the 4-Way Handshake or if no response to the 4-Way Handshake occurs.

11.6.11.2.2 Authenticator state machine: Group Key Handshake (per STA)

The following list summarizes the states the Authenticator state machine uses to support the Group Key Handshake:

- **IDLE:** This state is entered when no Group Key Handshake is occurring.
- **KEYERROR:** This state is entered if the EAPOL-Key acknowledgment for the Group Key Handshake is not received.
- **REKEYESTABLISHED:** This state is entered when an EAPOL-Key frame is received from the Supplicant with the Key Type subfield equal to Group.
- **REKEYNEGOTIATING:** This state is entered when the GTK is to be sent to the Supplicant.

NOTE—The TxRx flag for sending a GTK is always the opposite of whether the pairwise key is used for data encryption/integrity or not. If a pairwise key is used for encryption/integrity, then the STA never transmits with the GTK; otherwise, the STA uses the GTK for transmit.

11.6.11.2.3 Authenticator state machine: Group Key Handshake (global)

The following list summarizes the states the Authenticator state machine uses to coordinate a group key update of all STAs:

- **GTK_INIT:** This state is entered on system initialization.
- **SETKEYS:** This state is entered if the GTK is to be updated on all Supplicants.
- **SETKEYSDONE:** This state is entered if the GTK has been updated on all Supplicants.

NOTE—SETKEYSDONE calls SetGTK to set the GTK for all associated STAs that are not in WNM-Sleep Mode. If this fails, all communication via this key fails, and the AP needs to detect and recover from this situation. A STA that is in WNM-Sleep Mode will not have the current GTK installed when it wakes up and will need to get new GTK as described in the WNM-Sleep Mode procedures in 10.2.1.18.

11.6.11.3 Authenticator state machine variables

The following list summarizes the variables used by the Authenticator state machine:

- *AuthenticationRequest* – This variable is set to TRUE by the STA's IEEE 802.11 management entity in order to authenticate an association. This can be set to TRUE when the STA associates or at other times.

- *ReAuthenticationRequest* – This variable is set to TRUE if the IEEE 802.1X Authenticator received an eapStart or 802.1X::reAuthenticate is 1.
- *DeauthenticationRequest* – This variable is set to TRUE if a Disassociation or Deauthentication message is received.
- *Disconnect* – This variable is set to TRUE when the STA should initiate a deauthentication.
- *EAPOLKeyReceived* – This variable is set to TRUE when an EAPOL-Key frame is received. EAPOL-Key frames that are received in response to an EAPOL-Key frame sent by the Authenticator shall contain the same Key Replay Counter field value as the Key Replay Counter field in the transmitted message. EAPOL-Key frames that contain different Key Replay Counter field values should be discarded. An EAPOL-Key frame that is sent by the Supplicant in response to an EAPOL-Key frame from the Authenticator shall not have the Ack bit set to 1. EAPOL-Key frames sent by the Supplicant not in response to an EAPOL-Key frame from the Authenticator shall have the Request bit set to 1.

  EAPOL-Key frames with a key type of Pairwise and a nonzero key index should be ignored.

  EAPOL-Key frames with a key type of Group and an invalid key index should be ignored.

  NOTE—When an EAPOL-Key frame in which the Ack bit is 0 is received, then it is expected as a reply to a message that the Authenticator sent, and the replay counter is checked against the replay counter used in the sent EAPOL-Key frame. When an EAPOL-Key frame in which the Request bit is 1 is received, then a replay counter for these messages is used that is a different replay counter than the replay counter used for sending messages to the Supplicant.

- *GTimeoutCtr* – This variable maintains the count of EAPOL-Key receive timeouts for the Group Key Handshake. It is incremented each time a timeout occurs on EAPOL-Key receive event and is initialized to 0. Annex C details the timeout values. The Key Replay Counter field value for the EAPOL-Key frame shall be incremented on each transmission of the EAPOL-Key frame.
- *GInit* – This variable is used to initialize the group key state machine. This is a group variable.
- *Init* – This variable is used to initialize per-STA state machine
- *TimeoutEvt* – This variable is set to TRUE if the EAPOL-Key frame sent out fails to obtain a response from the Supplicant. The variable may be set to 1 by management action or set to 1 by the operation of a timeout while in the PTKSTART and REKEYNEGOTIATING states.
- *TimeoutCtr* **–** This variable maintains the count of EAPOL-Key receive timeouts. It is incremented each time a timeout occurs on EAPOL-Key receive event and is initialized to 0. Annex C contains details of the timeout values. The Key Replay Counter field value for the EAPOL-Key frame shall be incremented on each transmission of the EAPOL-Key frame.
- *MICVerified* – This variable is set to TRUE if the MIC on the received EAPOL-Key frame is verified and is correct. Any EAPOL-Key frames with an invalid MIC are dropped and ignored.
- *GTKAuthenticator* – This variable is set to TRUE if the Authenticator is on an AP or it is the designated Authenticator for an IBSS.
- *GKeyDoneStations* – Count of number of STAs left to have their GTK updated. This is a global variable.
- *GTKRekey* – This variable is set to TRUE when a Group Key Handshake is required. This is a global variable.
- *GUpdateStationKeys* – This variable is set to TRUE when a new GTK is available to be sent to Supplicants.
- *GNoStations* – This variable counts the number of Authenticators so it is known how many Supplicants need to be sent the GTK. This is a global variable.
- *Counter* – This variable is the global STA key counter.
- *ANonce* – This variable holds the current nonce to be used if the STA is an Authenticator.
- *GN*, *GM* – These are the current key indices for GTKs. Swap(GM, GN) means that the global key index in GN is swapped with the global key index in GM, so now GM and GN are reversed.

- *PTK* – This variable is the current PTK.
- *GTK[]* – This variable is the current GTKs for each GTK index.
- *PMK* – This variable is the buffer holding the current PMK.
- *802.1X::XXX* – This variable is the IEEE 802.1X state variable *XXX*.
- *keycount* – This variable is used in IBSS mode to decide when all the keys have been delivered and an IBSS link is secure.
- *WNM-Sleep Mode* – This variable is true when the non-AP STA is in the WNM-Sleep Mode, as described in 10.2.1.18. Otherwise, it is false.

11.6.11.4 Authenticator state machine procedures

The following list summarizes the procedures used by the Authenticator state machine:

- **STADisconnect**() – Execution of this procedure deauthenticates the STA.
- **CalcGTK**(x) – Generates the GTK.
- **MIC**(x) – Computes a MIC over the plaintext data.

11.7 Mapping EAPOL keys to IEEE 802.11 keys

11.7.1 Mapping PTK to TKIP keys

See 11.6.1.3 for the definition of the EAPOL temporal key derived from PTK.

A STA shall use bits 0–127 of the temporal key as its input to the TKIP Phase 1 and Phase 2 mixing functions.

A STA shall use bits 128–191 of the temporal key as the Michael key for MSDUs from the Authenticator's STA to the Supplicant's STA.

A STA shall use bits 192–255 of the temporal key as the Michael key for MSDUs from the Supplicant's STA to the Authenticator's STA.

11.7.2 Mapping GTK to TKIP keys

See 11.6.1.4 for the definition of the EAPOL temporal key derived from GTK.

A STA shall use bits 0–127 of the temporal key as the input to the TKIP Phase 1 and Phase 2 mixing functions.

A STA shall use bits 128–191 of the temporal key as the Michael key for MSDUs from the Authenticator's STA to the Supplicant's STA.

A STA shall use bits 192–255 of the temporal key as the Michael key for MSDUs from the Supplicant's STA to the Authenticator's STA.

11.7.3 Mapping PTK to CCMP keys

See 11.6.1.3 for the definition of the EAPOL temporal key derived from PTK.

A STA shall use the temporal key as the CCMP key for MPDUs between the two communicating STAs.

11.7.4 Mapping GTK to CCMP keys

See 11.6.1.4 for the definition of the EAPOL temporal key derived from GTK.

A STA shall use the temporal key as the CCMP key.

11.7.5 Mapping GTK to WEP-40 keys

See 11.6.1.4 for the definition of the EAPOL temporal key derived from GTK.

A STA shall use bits 0–39 of the temporal key as the WEP-40 key.

11.7.6 Mapping GTK to WEP-104 keys

See 11.6.1.4 for the definition of the EAPOL temporal key derived from GTK.

A STA shall use bits 0–103 of the temporal key as the WEP-104 key.

11.7.7 Mapping IGTK to BIP keys

See 11.6.1.5 for the definition of the IGTK key. A STA shall use bits 0–127 of the IGTK as the AES-128-CMAC key.

11.8 Per-frame pseudo-code

11.8.1 WEP frame pseudo-code

An MPDU of type Data with the Protected Frame subfield of the Frame Control field equal to 1 is called a WEP MPDU. Other MPDUs of type Data are called non-WEP MPDUs.

A STA shall not transmit WEP-encapsulated MPDUs when dot11PrivacyInvoked is false. This MIB variable does not affect the reception of frames containing all or part of an MSDU or MMPDU.

```
if dot11PrivacyInvoked is “false” then
    the MPDU is transmitted without WEP cryptographic encapsulation
else
    if (the MPDU has an individual RA and there is an entry in dot11WEPKeyMappings for that
      RA) then
        if that entry has WEPOn equal to “false” then
            the MPDU is transmitted without WEP cryptographic encapsulation
        else
            if that entry contains a key that is null then
                discard the MPDU’s entire MSDU and generate an MA-UNITDATA-
                  STATUS.indication primitive to notify LLC that the MSDU was
                  undeliverable due to a null WEP key
            else
                encrypt the MPDU using that entry’s key, setting the Key ID subfield of the IV
                  field to 0
            endif
        endif
    else
        if (the MPDU has a group RA and the Privacy subfield of the Capability Information field
          in this BSS is 0) then
            the MPDU is transmitted without WEP cryptographic encapsulation
        else
            if dot11WEPDefaultKeys[dot11WEPDefaultKeyID] is null then
                discard the MPDU's entire MSDU and generate an MA-UNITDATA-
                  STATUS.indication primitive to notify LLC that the MSDU was
                  undeliverable due to a null WEP key
            else
                WEP-encapsulate the MPDU using the key dot11WEPDefaultKeys-
                  [dot11WEPDefaultKeyID], setting the Key ID subfield of the IV field to
                  dot11WEPDefaultKeyID
            endif
        endif
    endif
endif
```

When the boolean dot11ExcludeUnencrypted is true, non-WEP MPDUs shall not be indicated at the MAC service interface, and only MSDUs successfully reassembled from successfully decrypted MPDUs shall be indicated at the MAC service interface. When receiving a frame of type Data, the values of dot11PrivacyOptionImplemented, dot11WEPKeyMappings, dot11WEPDefaultKeys, dot11WEPDefaultKeyID, and dot11ExcludeUnencrypted in effect at the time the PHY-RXSTART.indication primitive is received by the MAC shall be used according to the following decision tree:

```
if the Protected Frame subfield of the Frame Control field is 0 then
    if dot11ExcludeUnencrypted is "true" then
        discard the frame body without indication to LLC and increment
          dot11WEPExcludedCount
    else
        receive the frame without WEP decapsulation
    endif
else
    if dot11PrivacyOptionImplemented is "true" then
        if (the MPDU has individual RA and there is an entry in dot11WEPKeyMappings
              matching the MPDU's TA) then
            if that entry has WEPOn equal to "false" then
                discard the frame body and increment dot11WEPUndecryptableCount
            else
                if that entry contains a key that is null then
                    discard the frame body and increment dot11WEPUndecryptableCount
                else
                    WEP-decapsulate with that key, incrementing dot11WEPICVErrorCount if
                      the ICV check fails
                endif
            endif
        else
            if dot11WEPDefaultKeys[Key ID] is null then
                discard the frame body and increment dot11WEPUndecryptableCount
            else
                WEP-decapsulate with dot11WEPDefaultKeys[Key ID], incrementing
                  dot11WEPICVErrorCount if the ICV check fails
            endif
        endif
    else
        discard the frame body and increment dot11WEPUndecryptableCount
    endif
endif
```

11.8.2 RSNA frame pseudo-code

11.8.2.1 General

STAs transmit protected MSDUs, A-MSDUs, and robust management frames to an RA when temporal keys are configured and an MLME-SETPROTECTION.request primitive has been invoked with ProtectType parameter Tx or Rx_Tx to that RA. STAs expect to receive protected MSDUs, A-MSDUs, and robust management frames from a TA when temporal keys are configured and an MLME-SET-PROTECTION.request primitive has been invoked with ProtectType parameter Rx or Rx_Tx from that TA. MSDUs, A-MSDUs, and robust management frames that do not match these conditions are sent in the clear and are received in the clear.

11.8.2.2 Per-MSDU/Per-A-MSDU Tx pseudo-code

```
if dot11RSNAActivated = true then
    if MSDU or A-MSDU has an individual RA and Protection for RA is off for Tx then
        transmit the MSDU or A-MSDU without protections
    else if (MPDU has individual RA and Pairwise key exists for the MPDU's RA) or (MPDU has
      a group addressed RA and network type is IBSS and IBSS GTK exists for MPDU's TA)
      then
        // If we find a suitable Pairwise or GTK for the mode we are in…
        if key is a null key then
            discard the entire MSDU or A-MSDU and generate one or more MA-UNITDATA-
              STATUS.indication primitives to notify LLC that the MSDUs were undeliverable
              due to a null key
        else
            // Note that it is assumed that no entry in the key
            // mapping table is of an unsupported cipher type
            Set the Key ID subfield of the IV field to 0.
            if cipher type of entry is AES-CCM then
                Transmit the MSDU or A-MSDU, to be protected after fragmentation using
                AES-CCM
            else if cipher type of entry is TKIP then
                Compute MIC using Michael algorithm and entry's Tx MIC key.
                Append MIC to MSDU
                Transmit the MSDU, to be protected with TKIP
            else if cipher type of entry is WEP then
                Transmit the MSDU, to be protected with WEP
            endif
        endif
    else // Else we did not find a key but we are protected, so handle the default key case or discard
        if GTK entry for Key ID contains null then
            discard the MSDU or A-MSDU and generate one or more MA-UNITDATA-
              STATUS.indication primitives to notify the LLC that the MSDUs were
              undeliverable due to a null GTK
        else if GTK entry for Key ID is not null then
            Set the Key ID subfield of the IV field to the Key ID.
            if MPDU has an individual RA and cipher type of entry is not TKIP then
                discard the entire MSDU or A-MSDU and generate one or more MA-
                  UNITDATA-STATUS.indication primitives to notify the LLC that the
                  MSDUs were undeliverable due to a null key
            else if cipher type of entry is AES-CCM then
                Transmit the MSDU or A-MSDU, to be protected after fragmentation using
                AES-CCM
```

```
                    else if cipher type of entry is TKIP then
                        Compute MIC using Michael algorithm and entry's Tx MIC key.
                        Append MIC to MSDU
                        Transmit the MSDU, to be protected with TKIP
                    else if cipher type of entry is WEP then
                        Transmit the MSDU, to be protected with WEP
                    endif
                endif
            endif
        endif
```

11.8.2.3 Per-MMPDU Tx pseudo-code

```
if ((dot11RSNAActivated = TRUE) and (frame is a robust management frame)) then
    if ((dot11RSNAProtectedManagementFramesActivated = FALSE) then
        Transmit the MMPDU without protection
    else // dot11RSNAProtectedManagementFramesActivated = TRUE
        if (dot11RSNAUnprotectedManagementFramesAllowed = TRUE) then
            if (MMPDU has an individual RA) then
                if (peer STA advertised MFPC = 1) then
                    if (Pairwise key exists for the MMPDU's RA) then
                      // Note that it is assumed that no entry in the key
                      // mapping table is of an unsupported cipher.
                      Transmit the MMPDU, to be protected after fragmentation
                      // see 11.8.2.5
                    else if (robust Action frame) then
                      // pairwise key was not found
                      Discard the MMPDU and generate an MLME.confirm primitive to
                      notify the SME that the MMPDU was not delivered
                    else // Disassociation or Deauthentication
                      Transmit the MMPDU without protection
                    endif
                else // (peer STA didn't advertised MFPC = 1)
                    Transmit the MMPDU without protection
                endif
            else // MMPDU has a group RA
                if (IGTK exists) then
                    // if we find a suitable IGTK
                    Transmit the MMPDU with protection
                    // See 11.8.2.5
                else if (MMPDU is Disassociate ||Deauthenticate ||(not a robust Action frame))
                then
                    Transmit the MMPDU without protection
                else
                    Discard the MMPDU and generate an MLME.confirm primitive to notify
                    the SME that the MMPDU was undeliverable
                    endif
                endif
            else // dot11RSNAUnprotectedManagementFramesAllowed = FALSE
                if (MMPDU has an individual RA) then
                    if (peer STA advertised MFPC = 1) then
                        if (Pairwise key exists for the MMPDU's RA) then
                            // Note that it is assumed that no entry in the key
                            // mapping table is of an unsupported cipher.
                            Transmit the MMPDU, to be protected after fragmentation
                            // see 11.8.2.5
                        else if (robust Action frame) then
                            // pairwise key was not found
                            Discard the MMPDU and generate an MLME.confirm primitive to
                            notify the SME that the MMPDU was not delivered
                        else // FrameControlSubType is Disassociation or Deauthentication
                            Transmit the MMPDU without protection
                        endif
                    else // peer STA didn't advertise MFPC = 1
                        Discard the MMPDU and generate an MLME.confirm primitive to notify
                        the SME that the MMPDU was not delivered
                    endif
                else // MMPDU has a group RA
                    if (IGTK exists) then
                        // if we find a suitable IGTK
                        Transmit the MMPDU with protection
                        // See 11.8.2.5
                    else if (MMPDU is Disassociate || Deauthenticate || (not a robust Action frame))
                    then
                        Transmit the MMPDU without protection
                    else
                        Discard the MMPDU and generate an MLME.confirm primitive to notify
                        the SME that the MMPDU was undeliverable
                    endif
                endif
            endif
        endif
    else // (dot11RSNAActivated = FALSE) or (not a robust management frame)
        Use 11.8.2.2 to transmit the frame
    endif
```

11.8.2.4 Per-MPDU Tx pseudo-code

```
    if dot11RSNAActivated = TRUE then
        if MPDU is member of an MSDU that is to be transmitted without protections
            transmit the MPDU without protections
        else if MSDU or A-MSDU that MPDU is a member of is to be protected using AES-CCM
            Protect the MPDU using entry’s key and AES-CCM
            Transmit the MPDU
        else if MSDU that MPDU is a member of is to be protected using TKIP
            Protect the MPDU using TKIP encryption
            Transmit the MPDU
        else if MSDU that MPDU is a member of is to be protected using WEP
            Encrypt the MPDU using entry’s key and WEP
            Transmit the MPDU
        else
            // should not arrive here
        endif
    endif
```

11.8.2.5 Per-MPDU Tx pseudo-code for MMPDU

```
if ((dot11RSNAActivated = TRUE) then
    if (MPDU is member of an MMPDU that is to be transmitted without protection) then
        Transmit the MPDU without protection
    else if (MPDU has an individual RA) then
        Protect the MPDU using entry's TK and selected cipher from RSNE
        Transmit the MPDU
    else
        // MPDU has a group RA
        Protect the MPDU using IGTK and BIP
        Transmit the MPDU
    endif
endif
```

11.8.2.6 Per-MPDU Rx pseudo-code

```
if dot11RSNAActivated = TRUE then
    if the Protected Frame subfield of the Frame Control field is 0 then
        if Protection for TA is off for Rx then
            Receive the unencrypted MPDU without protections
        else
            discard the frame body without indication to LLC and increment
              dot11WEPExcludedCount
        endif
    else if Protection is true for TA then
        if ((MPDU has individual RA and Pairwise key exists for the MPDU’s TA) or (MPDU
           has a group addressed RA and network type is IBSS and IBSS GTK exists for
           MPDU’s RA)) then
            if MPDU has individual RA then
                lookup pairwise key using Key ID from MPDU
            else
                lookup group key using Key ID from MPDU
            endif
            if key is null then
                discard the frame body and increment dot11WEPUndecryptableCount
            else if entry has an AES-CCM key then
```

```
                    decrypt frame using AES-CCM key
                    discard the frame if the integrity check fails and increment dot11RSNAStats-
                        CCMPDecryptErrors
                **else if** entry has a TKIP key **then**
                    prepare a temporal key from the TA, TKIP key and PN
                    decrypt the frame using ARC4
                    discard the frame if the ICV fails and increment dot11RSNAStatsTKIPLocal-
                        MicFailures
                **else if** entry has a WEP key **then**
                    decrypt the frame using WEP decryption
                    discard the frame if the ICV fails and increment dot11WEPICVErrorCount
                **else**
                    discard the frame body **and** increment dot11WEPUndecryptableCount
                **endif**
            **else if** GTK for the Key ID does not exist **then**
                discard the frame body **and** increment dot11WEPUndecryptableCount
            **else if** GTK for the Key ID is null **then**
                discard the frame body **and** increment dot11WEPUndecryptableCount
            **else if** the GTK for the Key ID is a CCM key **then**
                decrypt frame using AES-CCM key
                discard the frame if the integrity check fails and increment dot11RSNAStatsCCMP-
                    DecryptErrors
            **else if** the GTK for the Key ID is a TKIP key **then**
                prepare a temporal key from the TA, TKIP key and PN
                decrypt the frame using ARC4
                discard the frame if the ICV fails and increment dot11RSNAStatsTKIPICVErrors
            **else if** the GTK for the Key ID is a WEP key **then**
                decrypt the frame using WEP decryption
                discard the frame if the ICV fails and increment dot11WEPICVErrorCount
            **endif**
        **else**
            MLME-PROTECTEDFRAMEDROPPED.indication
            discard the frame body **and** increment dot11WEPUndecryptableCount
        **endif**
    **endif**
```

11.8.2.7 Per-MPDU Rx pseudo-code for an MMPDU

```
if ((dot11RSNAActivated = TRUE) and (frame is a robust management frame)) then
    if ((dot11RSNAProtectedManagementFramesActivated = FALSE) then
        if (Protected Frame subfield of the Frame Control field is equal to 1) then
            Discard the frame
        else
            Receive the MMPDU
        endif
    else // dot11RSNAProtectedManagementFramesActivated = TRUE
        if (dot11RSNAUnprotectedManagementFramesAllowed = TRUE) then
            if (STA with frame TA advertised MFPC = 0) then
                if (Protected Frame subfield of the Frame Control field is equal to 1) then
                    Discard the frame
                else
                    Make frame available for further processing
                endif
            else // STA with frame TA advertised MFPC = 1
                if (MMPDU has an individual RA) then
                    if (Pairwise key does not exist) then
                        if (frame is a Disassociation or Deauthentication) then
                            if (Protected Frame subfield of the Frame Control field is equal to 0) then
                                Make the MPDU available for further processing
                            else // encrypted
                                Discard the frame
                            endif
                        else // frame is not a Disassociation or Deauthenticate
                            Discard the frame
                        endif
                    else if (security association has an AES-CCM key) then
                        if (Protected Frame subfield of the Frame Control field is equal to 0) then
                            //unprotected frame
                            Discard the frame
                        else // frame is encrypted
                            if (PN is not sequential) then
                                Discard the MPDU as a replay
                                Increment dot11RSNAStatsCCMPReplays
                            else
                                Decrypt frame using AES-CCM key
                                if (the integrity check fails) then
                                    Discard the frame
                                    Increment dot11RSNAStatsCCMPDecryptErrors
                                else
                                    Make the MPDU available for further processing
                                endif
                            endif
                        endif
                    else // key for some other cipher—for future expansion
                    endif
                else // MMPDU has a group RA
                    if (IGTK does not exist) then
                        if (Disassociation or Deauthentication) then
                            Make frame available for further processing
                        else
                            Discard the frame
                        endif
                    else // IGTK exists
                        if (MME is not present) then
```

```
                    Discard the frame
                else // MME is present
                    if (AES-128-CMAC IGTK) then
                        if (IPN is not valid) then
                            Discard the frame as a replay
                            Increment dot11RSNAStatsCMACReplay
                        else if (integrity check fails) then
                            Discard the frame
                            Increment dot11RSNAStatsCMACICVError
                        else
                            Make frame available for further processing
                        endif
                    else // some other kind of key—for the future
                    endif
                endif
            endif
        endif
    endif
else // dot11RSNAUnprotectedManagementFramesAllowed = FALSE
    if (MMPDU has an individual RA) then
        if (peer STA advertised MFPC = 1) then
            if (Pairwise key exists for the MMPDU's RA) then
                if (security association has an AES-CCM key) then
                    if (Protected Frame subfield of the Frame Control field is equal to 0) then
                        Discard the frame
                    else // frame is encrypted
                        if (PN is not sequential) then
                            Discard the MPDU as a replay
                            Increment dot11RSNAStatsCCMPReplays
                        else
                            Decrypt frame using AES-CCM key
                            if (the integrity check fails) then
                                Discard the frame
                                Increment dot11RSNAStatsCCMPDecryptErrors
                            else
                                Make the MPDU available for further processing
                            endif
                        endif
                    endif
                else // key for some other cipher—for future expansion
                endif
            else if (Protected Frame subfield of the Frame Control field is set to 1) then
                Discard the frame
```

```
                    else if (Deauthenticate || Disassociate) then
                        Make frame available for processing
                    else
                        Discard the frame
                    endif
                else // peer STA didn't advertise MFPC = 1
                    Discard the frame
                endif
            else // MMPDU has a group RA
                if (IGTK exists) then
                    if (MME is not present) then
                        Discard the frame
                    else // MME is present
                        if (AES-128-CMAC IGTK) then
                            if (PN is not valid) then
                                Discard the frame as a replay
                                Increment dot11RSNAStatsCMACReplay
                            else if (security association has an AES-128-CMAC IGTK) then
                                Discard the frame
                                Increment dot11RSNAStatsCMACICVError
                            else
                                Make frame available for further processing
                            endif
                        else // some other kind of key—for the future
                        endif
                    endif
                else // IGTK does not exist
                    if (Disassociation or Deauthentication) then
                        Make frame available for further processing
                    else
                        Discard the frame
                    endif
                endif
            endif
        endif
    endif
else // (dot11RSNAActivated = FALSE) or (not a robust management frame)
    Use 11.8.2.6 to receive the frame
endif
```

11.8.2.8 Per-MSDU/Per-A-MSDU Rx pseudo-code

```
if dot11RSNAActivated = TRUE then
    if the frame was not protected then
```

```
            Receive the MSDU or A-MSDU unprotected
            Make MSDU(s) available to higher layers
        else// Have a protected MSDU or A-MSDU
            if Pairwise key is an AES-CCM key then
                Accept the MSDU or A-MSDU if its MPDUs had sequential PNs (or if it consists of
                    only one MPDU), otherwise discard the MSDU or A-MSDU as a replay attack and
                    increment dot11RSNAStatsCCMPReplays
                Make MSDU(s) available to higher layers
            else if Pairwise key is a TKIP key then
                Compute the MIC using the Michael algorithm
                Compare the received MIC to the computed MIC
                discard the frame if the MIC fails increment dot11RSNAStatsTKIPLocalMIC-
                    Failures and invoke countermeasures if appropriate
                compare TSC to replay counter, if replay check fails increment dot11RSNA-
                    StatsTKIPReplays
                otherwise accept the MSDU
                Make MSDU available to higher layers
            else if dot11WEPKeyMappings has a WEP key then
                Accept the MSDU since the decryption took place at the MPDU
                Make MSDU available to higher layers
            endif
        endif
    endif
```

11.8.2.9 Per-MMPDU Rx pseudo-code

```
    if (dot11RSNAActivated = TRUE) then
        if (dot11RSNAProtectedManagmentFramesActivated = TRUE) then
            if (the MPDU was not protected) then
                Receive the MMPDU unprotected
                Make the MMPDU available to higher layers
            else //Have a protected MMPDU
                if ((MMPDU has individual RA) and (security association has an AES-CCM key))
                then
                    if (the MPDU has only one MPDU or multiple MPDUs with sequential PNs)
                    then
                        Receive the MMPDU protected
                        Make the MMPDU available to higher layers
                    else
                        Discard the MMPDU as a replay
                        Increment dot11RSNAStatsRobustMgmtCCMPReplays
                    endif
                else if ((MPDU has group addressed RA) and (security association has an AES-128-
                CMAC IGTK)) then
                    Receive the MMPDU
                    Make the MMPDU available to higher layers
                else
                    if (any other cipher exists) then
                        Process the frame using other cipher
                    else
                        Discard the frame
                    endif
                endif
            endif
        endif
    endif
```

11.9 Authenticated mesh peering exchange (AMPE)

The authenticated mesh peering exchange is defined in 13.5.

12. Fast BSS transition

12.1 Overview

Fast BSS transition seeks to reduce the length of time that connectivity is lost between a STA and the DS during a BSS transition. The FT protocols are part of the reassociation service and only apply to STA transitions between APs within the same mobility domain within the same ESS.

The FT protocols require information to be exchanged during the initial association (or a later reassociation) between a STA (known as the *FT Originator* (FTO)) and AP. The initial exchange is referred to as the *FT initial mobility domain association*. Subsequent reassociations to APs within the same mobility domain may make use of the FT protocols.

Two FT protocols are defined:

- *FT Protocol.* This protocol is executed when an FTO makes a transition to a target AP and does not require a resource request prior to its transition.
- *FT Resource Request Protocol.* This protocol is executed when an FTO requires a resource request prior to its transition.

For an FTO to move from its current AP to a target AP utilizing the FT protocols, the message exchanges are performed using one of two methods:

- *Over-the-Air.* The FTO communicates directly with the target AP using IEEE 802.11 authentication with the FT authentication algorithm.
- *Over-the-DS.* The FTO communicates with the target AP via the current AP. The communication between the FTO and the target AP is carried in FT Action frames between the FTO and the current AP. Between the current AP and target AP, communication is via an encapsulation method described in 12.10.3. The current AP converts between the two encapsulations.

APs advertise both capabilities and policies for supporting the FT protocols and methods.

Throughout this clause, the notation *Authentication-Request* refers to an Authentication frame with the Authentication Transaction Sequence Number field equal to 1; *Authentication-Response* refers to an Authentication frame with the Authentication Transaction Sequence Number field equal to 2; *Authentication-Confirm* refers to an Authentication frame with the Authentication Transaction Sequence Number field equal to 3; *Authentication-Ack* refers to an Authentication frame with the Authentication Transaction Sequence Number field equal to 4. The first parameter to the above four messages is the authentication algorithm, such as Open System authentication algorithm (i.e., *Open* in figures in this clause) or FT authentication algorithm (i.e., *FTAA* in figures in this clause).

12.2 Key holders

12.2.1 Introduction

The FT key holder architecture, shown in Figure 12-1, describes the FT key management entities and is defined in the context of the IEEE 802.11 basic reference model (see Figure 4-14 in 4.9).

The R0KH and R1KH are part of AP's SME RSNA key management. The computation of PMK-R0 and PMK-R1, and all the intermediate results in the computations, shall be restricted to the R0KH. The computation of PTK, and all intermediate results in its computation, shall be restricted to the R1KH.

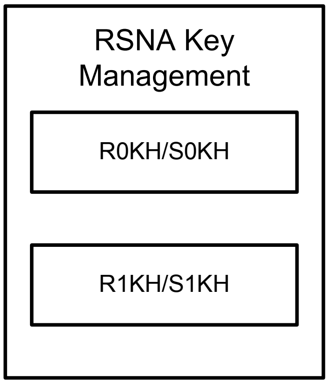


Figure 12-1—FT key holder architecture

The S0KH and S1KH are part of the FTO's SME RSNA key management. The computation of PMK-R0 and PMK-R1, and all the intermediate results in the computations, shall be restricted to the S0KH. The computation of PTK, and all intermediate results in its computation, shall be restricted to the S1KH.

12.2.2 Authenticator key holders

The R0KH and R1KH are responsible for the derivation of keys in the FT key hierarchy. For fast BSS transition, the functions of the IEEE 802.1X Authenticator are distributed among the R0KH and R1KHs.

The R0KH interacts with the IEEE 802.1X Authenticator to receive the MSK resulting from an EAP authentication. The R1KH interacts with the IEEE 802.1X Authenticator to open the Controlled Port. Both the R0KH and R1KH interactions with the IEEE 802.1X Authenticator occur within the SME.

The R0KH derives the PMK-R0 for use in the mobility domain utilizing the MSK (when the AKM negotiated is 00-0F-AC:3), the PSK (when the AKM negotiated is 00-0F-AC:4) or the PMK (when the AKM negotiated is 00-0F-AC:9). The R0KH shall be responsible for deriving a PMK-R1 for each R1KH within the mobility domain.

The R1KH and S1KH each derive the PTK.

Each R0KH-ID and R1KH-ID is assumed to be expressed as a unique identifier within the mobility domain. This identifier is communicated to the FTO and other key holders. The R0KH-ID is bound into the PMK-R0 derivation and the R1KH-ID is bound into the PMK-R1 derivation.

The R0KH shall meet the following requirements:

- The R0KH shall be collocated with the network access server (NAS) Client functionality of the IEEE 802.1X Authenticator.
- The R0KH-ID shall be set to the identity of the co-resident NAS Client (e.g., NAS-Identifier as defined in IETF RFC 2865 if RADIUS is used as the backend protocol). R0KH-ID shall not be longer than 48 octets to fit in the length limitation of the FTE.
- When the PMK-R0 lifetime expires, the R0KH shall delete the PMK-R0 security association and shall revoke within the R0KH all PMK-R1s derived from the PMK-R0.
- The R0KH shall not expose the PMK-R0 to other parties.
- The R0KH shall not expose the PMK-R1 to parties other than the authorized R1KH.

The R1KH shall meet the following requirements:

- The R1KH-ID shall be set to a MAC address of the physical entity that stores the PMK-R1 and uses it to generate the PTK. That same MAC address shall be used to advertise the PMK-R1 identity to the STA and the R0KH.
- The R1KH shall derive and distribute the GTK and IGTK to all connected STAs.
- When the PMK-R1 lifetime expires, the R1KH shall delete the PMK-R1 PMKSA and shall revoke all PTKSAs derived from the PMK-R1 using the MLME-DELETEKEYS primitive.
- The R1KH shall not expose the PMK-R1 to other parties.

dot11FTR0KeyHolderID and dot11FTR1KeyHolderID shall contain the values of R0KH-ID and R1KH-ID as defined in this clause, respectively.

The R0KH and the R1KH are assumed to have a secure channel between them that can be used to exchange cryptographic keys without exposure to any intermediate parties. The cryptographic strength of the secure channel between the R0KH and R1KH is assumed to be greater than or equal to the cryptographic strength of the channels for which the keys are used. This standard assumes that the key transfer includes the PMK-R1, the PMK-R1 PMKSA, the PMK-R1 context, and the associated key authorizations. The protocol for distribution of keying material from the R0KH to the R1KH is outside the scope of this standard.

The PMK-R1 distribution from the R0KH to the R1KHs within the same mobility domain shall satisfy the following assumptions:

- The R0KH authenticates a potential R1KH with the same identity as is included in the PMK-R1 derivation. The cryptographic strength of the authentication is assumed to be greater than or equal to the cryptographic strength of the authentication between the Supplicant and AS.
- The authorization of holding a PMK-R1 is based on the authentication of the R1KH.
- The protected channel provides confidentiality and integrity protection.

12.2.3 Supplicant key holders

The S0KH and S1KH are responsible for the derivation of keys in the FT key hierarchy. The S0KH and S1KH are entities that are assumed to physically reside in the Supplicant.

The S0KH interacts with the IEEE 802.1X functional block (see Figure 4-14 in 4.9) to receive the MSK resulting from an EAP authentication. The S1KH interacts with 802.1X to open the Controlled Port. Both the S0KH and S1KH interactions with 802.1X occur within the SME of a STA.

The S0KH derives the PMK-R0 for use in the mobility domain utilizing the MSK (when the AKM negotiated is 00-0F-AC:3), the PSK (when the AKM negotiated is 00-0F-AC:4) or the PMK (when the AKM negotiated is 00-0F-AC:9).

The S1KH shall derive the PTK mutually with the R1KH.

The S0KH and S1KH shall be identified by the SPA. The S0KH shall not expose the PMK-R0 to other parties and shall not expose the PMK-R1 to parties other than the authorized S1KH. The S1KH shall not expose the PMK-R1 to other parties.

12.3 Capability and policy advertisement

The FT capability is advertised in the Beacon and Probe Response frames by including the MDE. The MDE is advertised in the Beacon and Probe Response frames to indicate the MDID, FT capability, and the FT policy.

The MDID field shall be the value of dot11FTMobilityDomainID. The Fast BSS Transition Policy bits in the MDE, i.e., Fast BSS Transition over DS subfield and Resource Request Protocol Capability subfield, shall be set by dot11FTOverDSActivated and dot11FTResourceRequestSupported, respectively.

NOTE—It is assumed by this standard that the Fast BSS Transition Policy bits in the MDE are administered consistently across the mobility domain.

The capability is advertised in the Neighbor Report element. See 10.11 and 8.4.2.39.

If an FTE is included in a Request element in a Probe Request frame, the FTE in the Probe Response frame shall contain the R0KH-ID and R1KH-ID (from dot11FTR0KeyHolderID and dot11FTR1KeyHolderID), and all other fields shall be set to 0.

12.4 FT initial mobility domain association

12.4.1 Overview

The FT initial mobility domain association is the first (re)association in the mobility domain, where the SME of the STA enables its future use of the FT procedures.

FT initial mobility domain association is typically the first association within the ESS. In addition to association frames, reassociation frames are supported in the initial mobility domain association to enable both FT and non-FT APs to be present in a single ESS.

12.4.2 FT initial mobility domain association in an RSN

A STA indicates its support for the FT procedures by including the MDE in the (Re)Association Request frame and indicates its support of security by including the RSNE. The AP responds by including the FTE, MDE, and RSNE in the (Re)Association Response frame. After a successful IEEE 802.1X authentication (if needed) or SAE authentication, the STA and AP perform an FT 4-Way Handshake. At the end of the sequence, the IEEE 802.1X Controlled Port is opened, and the FT key hierarchy has been established. The message flow is shown in Figure 12-2.

A STA initiates the FT initial mobility domain association procedures by performing an IEEE 802.11 authentication using the Open System authentication algorithm.

STA→AP: Authentication-Request (Open System authentication algorithm)
AP→STA: Authentication-Response (Open System authentication algorithm, Status)

The SME of the STA initiates the authentication exchange, through the use of the MLME-AUTHENTICATE.request primitive, and the SME of the AP responds with MLME-AUTHENTICATE.response primitive. See 10.3.4.

Upon successful IEEE 802.11 Open System authentication, (if the suite type is 00-0F-AC:3 or 00-0F-AC:4) or SAE authentication (if the suite type is 00-0F-AC:9), the STA shall send a (Re)Association Request frame to the AP that includes the MDE. The contents of the MDE shall be the values advertised by the AP in its Beacon or Probe Response frames. Additionally, the STA includes its security capabilities in the RSNE.

STA→AP: (Re)Association Request (MDE, RSNE)
AP→STA: (Re)Association Response (MDE, FTE[R1KH-ID, R0KH-ID])

The SME of the STA initiates the (re)association through the use of the MLME-ASSOCIATE.request or MLME-REASSOCIATE.request primitive. The SME of the AP responds to the indication with MLME-ASSOCIATE.response or MLME-REASSOCIATE.response primitive. See 10.3.5.

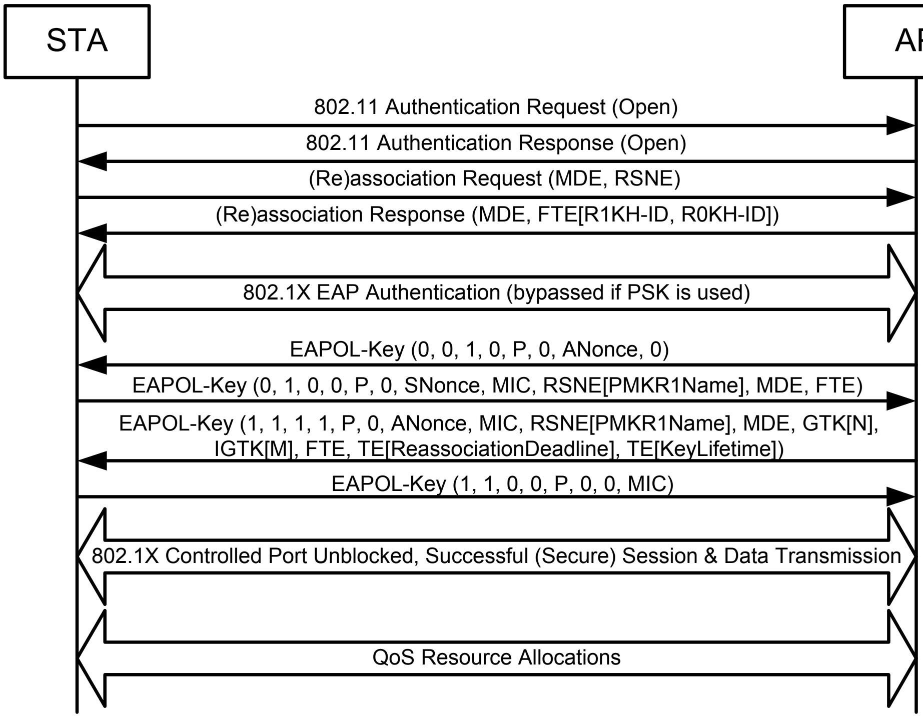

STA
AP
802.11 Authentication Request (Open)
802.11 Authentication Response (Open)
(Re)association Request (MDE, RSNE)
(Re)association Response (MDE, FTE[R1KH-ID, R0KH-ID])
802.1X EAP Authentication (bypassed if PSK is used)
EAPOL-Key (0, 0, 1, 0, P, 0, ANonce, 0)
EAPOL-Key (0, 1, 0, 0, P, 0, SNonce, MIC, RSNE[PMKR1Name], MDE, FTE)
EAPOL-Key (1, 1, 1, 1, P, 0, ANonce, MIC, RSNE[PMKR1Name], MDE, GTK[N], IGTK[M], FTE, TE[ReassociationDeadline], TE[KeyLifetime])
EAPOL-Key (1, 1, 0, 0, P, 0, 0, MIC)
802.1X Controlled Port Unblocked, Successful (Secure) Session & Data Transmission
QoS Resource Allocations

Figure 12-2—FT initial mobility domain association in an RSN

If the contents of the MDE received by the AP do not match the contents advertised in the Beacon and Probe Response frames, the AP shall reject the (Re)Association Request frame with status code 54 (i.e., Invalid MDE). If an MDE is present in the (Re)Association Request frame and the contents of the RSNE do not indicate a negotiated AKM of Fast BSS Transition (suite type 00-0F-AC:3, 00-0F-AC:4, or 00-0F-AC:9), the AP shall reject the (Re)Association Request frame with status code 43 (i.e., Invalid AKMP).

The (Re)Association Response frame from the AP shall contain an MDE, with contents as presented in Beacon and Probe Response frames. The FTE shall include the key holder identities of the AP, the R0KH-ID and R1KH-ID, set to the values of dot11FTR0KeyHolderID and dot11FTR1KeyHolderID, respectively. The FTE shall have a MIC element count of zero (i.e., no MIC present) and have ANonce, SNonce, and MIC fields set to 0.

On successful (re)association, the S0KH on the STA and the R0KH on the AP then proceed with an IEEE 802.1X authentication using EAPOL messages carried in IEEE 802.11 data frames if SAE authentication was not performed (i.e., if the suite type is not 00-0F-AC:9). The S0KH shall use the value of R0KH-ID as the endpoint identifier of the NAS Client (NAS-Identifier if RADIUS is used) in the exchange as defined in IETF RFC 3748-2004 [B38].

If IEEE 802.1X authentication was performed, then upon successful completion of authentication, the R0KH receives the MSK and authorization attributes. If SAE authentication was performed, the R0KH receives the PMK, resulting in the successful completion of SAE. If a key hierarchy already exists for this STA belonging to the same mobility domain (i.e., having the same MDID), the R0KH shall delete the existing PMK-R0 security association and PMK-R1 security associations. It then calculates the PMK-R0, PMKR0Name, and PMK-R1 and makes the PMK-R1 available to the R1KH of the AP with which the STA is associated.

If the SME of the STA cannot authenticate the AS, then it shall disassociate with an MLME-DISASSOCIATE.request primitive. If the AS signals the Authenticator that the STA cannot be authenticated, then the SME of the AP shall disassociate with an MLME-DISASSOCIATE.request primitive.

If the MSK lifetime attribute is provided by the AS, the lifetime of the PMK-R0 shall not be more than the lifetime of the MSK. If the MSK lifetime attribute is not provided, the PMK-R0 lifetime shall be dot11FTR0KeyLifetime. For PSK, the PMK-R0 lifetime shall be dot11FTR0KeyLifetime. The lifetime of the PMK-R1s and PTK shall be the same as the lifetime of PMK-R0. When the key lifetime expires, each key holder shall delete its respective PMK-R0, PMK-R1, and PTK SAs.

The R1KH and S1KH then perform an FT 4-Way Handshake. The EAPOL-Key frame notation is defined in 11.6.4.

```
R1KH→S1KH: EAPOL-Key(0, 0, 1, 0, P, 0, 0, ANonce, 0)
S1KH→R1KH: EAPOL-Key(0, 1, 0, 0, P, 0, 0, SNonce, MIC, RSNE[PMKR1Name], MDE, FTE)
R1KH→S1KH: EAPOL-Key(1, 1, 1, 1, P, 0, 0, ANonce, MIC, RSNE[PMKR1Name], MDE,
                                  GTK[N], IGTK[M], FTE, TIE[ReassociationDeadline],
                                  TIE[KeyLifetime])
S1KH→R1KH: EAPOL-Key(1, 1, 0, 0, P, 0, 0, 0, MIC)
```

The message sequence is similar to that of 11.6.6. The contents of each message shall be as described in 11.6.6 except as follows:

- Message 2: the S1KH shall include the PMKR1Name in the PMKID field of the RSNE. The PMKR1Name shall be as calculated by the S1KH according to the procedures of 11.6.1.7.4; all other fields of the RSNE shall be identical to the RSNE present in the (Re)Association Request frame. The S1KH shall include the FTE and MDE; the FTE and MDE shall be the same as those provided in the AP's (Re)Association Response frame.
- Message 3: the R1KH shall include the PMKR1Name in the PMKID field of the RSNE. The PMKR1Name shall be as calculated by the R1KH according to the procedures of 11.6.1.7.4 and shall be the same as the PMKR1Name in Message 2; all other fields of the RSNE shall be identical to the RSNE present in the Beacon or Probe Response frames. The R1KH shall also include the FTE, the MDE, the reassociation deadline timeout in the TIE[ReassociationDeadline], and the PTK key lifetime in the TIE[KeyLifetime]. The FTE and MDE shall be the same as in the (Re)Association Response frame. The reassociation deadline shall be set to the minimum of dot11FTReassociationDeadline and the key lifetime.

It is assumed by this standard that the reassociation deadline is administered consistently across the mobility domain. The mechanism for such consistent administration is outside the scope of this standard.

The PTK shall be calculated by the R1KH and S1KH according to the procedures given in 11.6.1.7.5.

Upon completion of a successful FT 4-Way Handshake, the IEEE 802.1X Controlled Port shall be opened on both the non-AP STA and the AP. Subsequent EAPOL-Key frames shall use the key replay counter to detect replayed messages.

Upon completion of a successful FT 4-Way Handshake, the PTK key lifetime timer is initiated to ensure that the lifetime of the PTKSA is no longer than the value provided in the TIE[KeyLifetime] sent in Message 3.

Once the PTKSA key lifetime expires, as indicated by the TIE[KeyLifetime], to continue its association in the mobility domain the STA shall perform the FT initial mobility domain association procedures. If the AP sends a Deauthentication or Disassociation frame to the STA with reason code 2 (i.e., Previous authentication no longer valid), then to continue its association in the mobility domain, the STA shall perform the FT initial mobility domain association procedures with any AP in the mobility domain. If the

Supplicant EAPOL state machines are triggered to send an EAPOL-Start packet after a successful initial mobility domain association, the STA shall perform the FT initial mobility domain association procedures.

12.4.3 FT initial mobility domain association in a non-RSN

In this sequence, the STA utilizes the FT procedures by including the MDE in the (Re)Association Request frame. The AP responds by including the MDE in the (Re)Association Response frame. The message flow is shown in Figure 12-3.

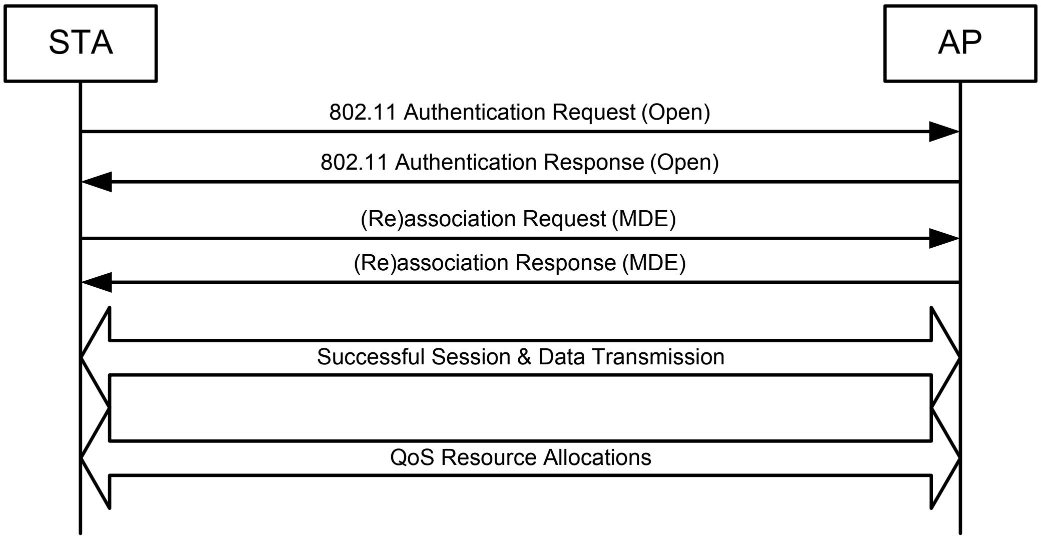

STA
AP
802.11 Authentication Request (Open)
802.11 Authentication Response (Open)
(Re)association Request (MDE)
(Re)association Response (MDE)
Successful Session & Data Transmission
QoS Resource Allocations

Figure 12-3—FT initial mobility domain association in a non-RSN

The STA initiates the FT initial mobility domain association procedures by performing an IEEE 802.11 authentication using the Open System authentication algorithm.

STA→AP: Authentication-Request (Open System authentication algorithm)
AP→STA: Authentication-Response (Open System authentication algorithm, Status)

The SME of the STA initiates the authentication exchange through the use of the primitive MLME-AUTHENTICATE.request primitive, and the SME of the AP responds with MLME-AUTHENTICATE.response primitive. See 10.3.4.

Upon successful IEEE 802.11 Open System authentication, the STA shall send a (Re)Association Request frame to the AP and shall include the MDE. The contents of the MDE shall be the values advertised by the AP in its Beacon or Probe Response frames.

STA→AP: (Re)Association Request (MDE)
AP→STA: (Re)Association Response (MDE)

The SME of the STA initiates the (Re)association through the use of the MLME-ASSOCIATE.request or MLME-REASSOCIATE.request primitive. The SME of the AP responds to the indication with MLME-ASSOCIATE.response or MLME-REASSOCIATE.response primitive. See 10.3.5.

If the contents of the MDE received by the AP do not match the contents advertised in the Beacon and Probe Response frames, the AP shall reject the (Re)Association Request frame with status code 54 (i.e., Invalid MDE).

The (Re)Association Response frame from the AP shall contain an MDE, with contents as presented in Beacon and Probe Response frames.

On successful (re)association, the AP and the non-AP STA shall transition to State 4 (as defined in 10.3) to enable data frame transmission.

12.5 FT Protocol

12.5.1 Overview

STAs with dot11FastBSSTransitionActivated equal to true shall support the FT Protocol.

The FT Protocol supports resource requests as part of the reassociation. The optional FT Resource Request Protocol (see 12.6) supports resource requests prior to reassociation.

A STA shall not use any authentication algorithm except the FT authentication algorithm when using the FT Protocol.

12.5.2 Over-the-air FT Protocol authentication in an RSN

The over-the-air FT Protocol in an RSN is shown in Figure 12-4.

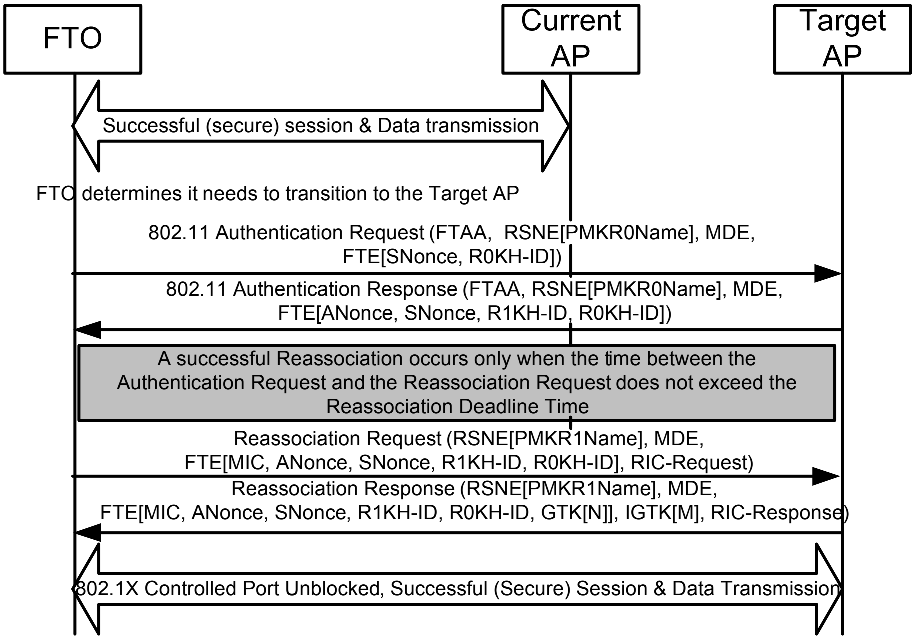

FTO
Current AP
Target AP
Successful (secure) session & Data transmission
FTO determines it needs to transition to the Target AP
802.11 Authentication Request (FTAA, RSNE[PMKR0Name], MDE, FTE[SNonce, R0KH-ID])
802.11 Authentication Response (FTAA, RSNE[PMKR0Name], MDE, FTE[ANonce, SNonce, R1KH-ID, R0KH-ID])
A successful Reassociation occurs only when the time between the Authentication Request and the Reassociation Request does not exceed the Reassociation Deadline Time
Reassociation Request (RSNE[PMKR1Name], MDE, FTE[MIC, ANonce, SNonce, R1KH-ID, R0KH-ID], RIC-Request)
Reassociation Response (RSNE[PMKR1Name], MDE, FTE[MIC, ANonce, SNonce, R1KH-ID, R0KH-ID, GTK[N]], IGTK[M], RIC-Response)
802.1X Controlled Port Unblocked, Successful (Secure) Session & Data Transmission

Figure 12-4—Over-the-air FT Protocol in an RSN

The FTO and AP use the FT authentication sequence to specify the PMK-R1 security association and to provide values of SNonce and ANonce that enable a liveness proof, replay protection, and PTK key separation. This exchange enables a fresh PTK to be computed in advance of reassociation. The PTKSA is used to protect the subsequent reassociation transaction, including the optional RIC-Request.

To perform an over-the-air fast BSS transition to a target AP, the FTO and target AP shall perform the following exchange:

FTO→Target AP: Authentication-Request (FTAA, 0, RSNE[PMKR0Name], MDE, FTE[SNonce, R0KH-ID])

Target AP→FTO: Authentication-Response (FTAA, Status, RSNE[PMKR0Name], MDE, FTE[ANonce, SNonce, R1KH-ID, R0KH-ID])

The SME of the FTO initiates the authentication exchange, through the use of the MLME-AUTHENTICATE.request primitive, and the SME of the AP responds with an MLME-AUTHENTICATE.response primitive. See 10.3.4. The MLME primitives for Authentication when the FT authentication algorithm is selected use only Authentication transaction sequence number values 1 and 2.

In the Authentication Request frame, the SA field of the message header shall be set to the MAC address of the FTO, and the DA field of the message header shall be set to the BSSID of the target AP. The elements in the frame, and their required contents, shall be as given in 12.8.2.

If the contents of the MDE received by the AP do not match the contents advertised in the Beacon and Probe Response frames, the AP shall reject the Authentication Request with status code 54 (i.e., Invalid MDE). If the Authentication Request frame contains an authentication algorithm equal to FT authentication and the contents of the RSNE do not indicate a negotiated AKM of Fast BSS Transition (suite type 00-0F-AC:3 or 00-0F-AC:4), the AP shall reject the Authentication Request with status code 43 (i.e., Invalid AKMP). If the FTE in the FT Request frame contains an invalid R0KH-ID, the AP shall reject the FT Request frame with status code 55 (i.e., Invalid FTE). If the RSNE in the Authentication Request frame contains an invalid PMKR0Name and the AP has determined that it is an invalid PMKR0Name, the AP shall reject the Authentication Request with status code 53 (i.e., Invalid PMKID). If the requested R0KH is not reachable, the AP shall respond to the Authentication Request with status code 28 (i.e., R0KH unreachable). If the FTO selects a pairwise cipher suite in the RSNE that is different from the ones used in the Initial mobility domain association, then the AP shall reject the Authentication Request with status code 19 (i.e., Invalid Pairwise Cipher). Subsequent to a rejection of an Authentication Request, the FTO may retry the Authentication Request.

In the Authentication Response frame, the SA field of the message header shall be set to the BSSID of the target AP, and the DA field of the message header shall be set to the MAC address of the FTO. The Status Code field shall be a value from the options listed in 8.4.1.9. The elements in the frame, and their required contents, shall be as given in 12.8.3.

The R1KH of the target AP uses the value of PMKR0Name and other information in the frame to calculate PMKR1Name. If the target AP does not have the key identified by PMKR1Name, it may retrieve that key from the R0KH identified by the FTO. See 12.2. Upon receiving a new PMK-R1 for a STA, the target AP shall delete the prior PMK-R1 security association and PTKSAs derived from the prior PMK-R1.

The FTO and the target AP compute the PTK and PTKName using the PMK-R1, PMKR1Name, ANonce, and SNonce, as specified in 11.6.1.7.5. The PTKSA shall be deleted by the target AP if it does not receive a Reassociation Request frame from the FTO within the reassociation deadline timeout value.

If the FTO does not receive a response to the Authentication Request frame, it may reissue the request following the restrictions given for Authentication frames in 10.3. If the Status Code field value returned by the target AP is 0, indicating success, the FTO and target AP transition to State 2 (as defined in 10.3); the FTO may continue with reassociation (12.7.1). Handling of errors returned in the Status Code field shall be as specified in 10.3.

12.5.3 Over-the-DS FT Protocol authentication in an RSN

A STA shall not initiate an over-the-DS FT authentication to a target AP whose MDE contains the Fast BSS Transition over DS bit equal to 0.

The over-the-DS FT Protocol in an RSN is shown in Figure 12-5.

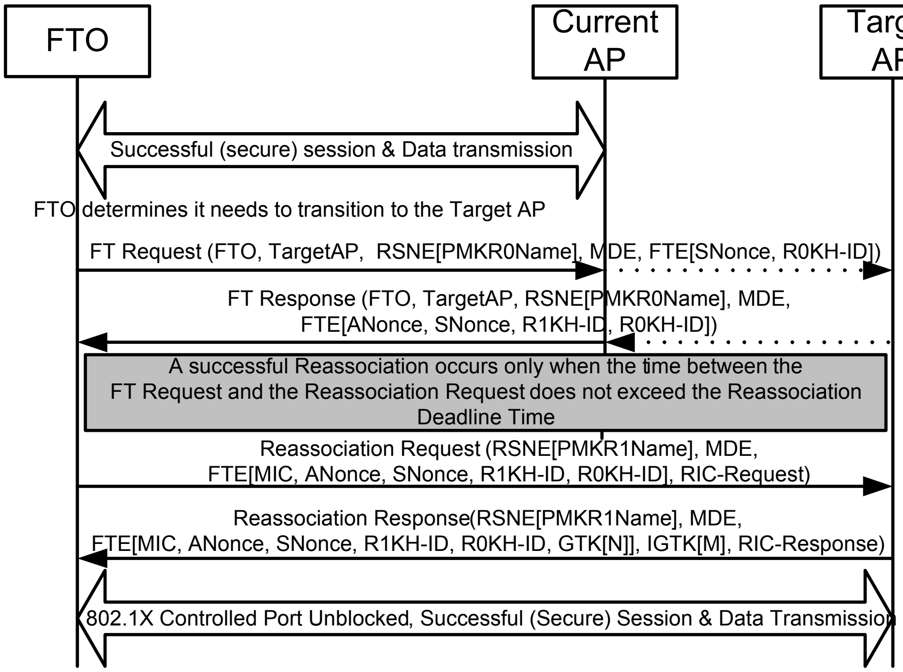

FTO
Current AP
Target AP
Successful (secure) session & Data transmission
FTO determines it needs to transition to the Target AP
FT Request (FTO, TargetAP, RSNE[PMKR0Name], MDE, FTE[SNonce, R0KH-ID])
FT Response (FTO, TargetAP, RSNE[PMKR0Name], MDE, FTE[ANonce, SNonce, R1KH-ID, R0KH-ID])
A successful Reassociation occurs only when the time between the FT Request and the Reassociation Request does not exceed the Reassociation Deadline Time
Reassociation Request (RSNE[PMKR1Name], MDE, FTE[MIC, ANonce, SNonce, R1KH-ID, R0KH-ID], RIC-Request)
Reassociation Response(RSNE[PMKR1Name], MDE, FTE[MIC, ANonce, SNonce, R1KH-ID, R0KH-ID, GTK[N]], IGTK[M], RIC-Response)
802.1X Controlled Port Unblocked, Successful (Secure) Session & Data Transmission

Figure 12-5—Over-the-DS FT Protocol in an RSN

To perform an over-the-DS fast BSS transition to a target AP, the FTO and the target AP (through the current AP) shall perform the following exchange:

FTO→Target AP: FT Request (FTO address, TargetAP address, RSNE[PMKR0Name], MDE, FTE[SNonce, R0KH-ID])

Target AP→FTO: FT Response (FTO address, TargetAP address, Status, RSNE[PMKR0Name], MDE, FTE[ANonce, SNonce, R1KH-ID, R0KH-ID])

The SME of the FTO initiates the FT Request frame to the target AP by issuing a MLME-REMOTE-REQUEST.request primitive with parameters including the contents of the FT Request frame (FT Action frame with an FT Action field value indicating FT Request) to be sent. The MAC of the FTO transmits this Action frame. For processing at the current AP and target AP, see 12.10. When the MAC of the FTO receives the FT Response frame (FT Action frame with an FT Action field value indicating FT Response), it passes it to the SME by use of MLME-REMOTE-REQUEST.indication primitive, with parameters including the contents of the received Action frame. The MLME interfaces on the FTO, current AP, and the target AP for executing the over-the-DS fast BSS transition are shown in Figure 12-6.

The STA Address field of the FT Request frame shall be set to the MAC address of the FTO, and the Target AP Address field of the FT Request frame shall be set to the BSSID of the target AP. The elements in the FT Request frame, and their required contents, shall be as given in 12.8.2.

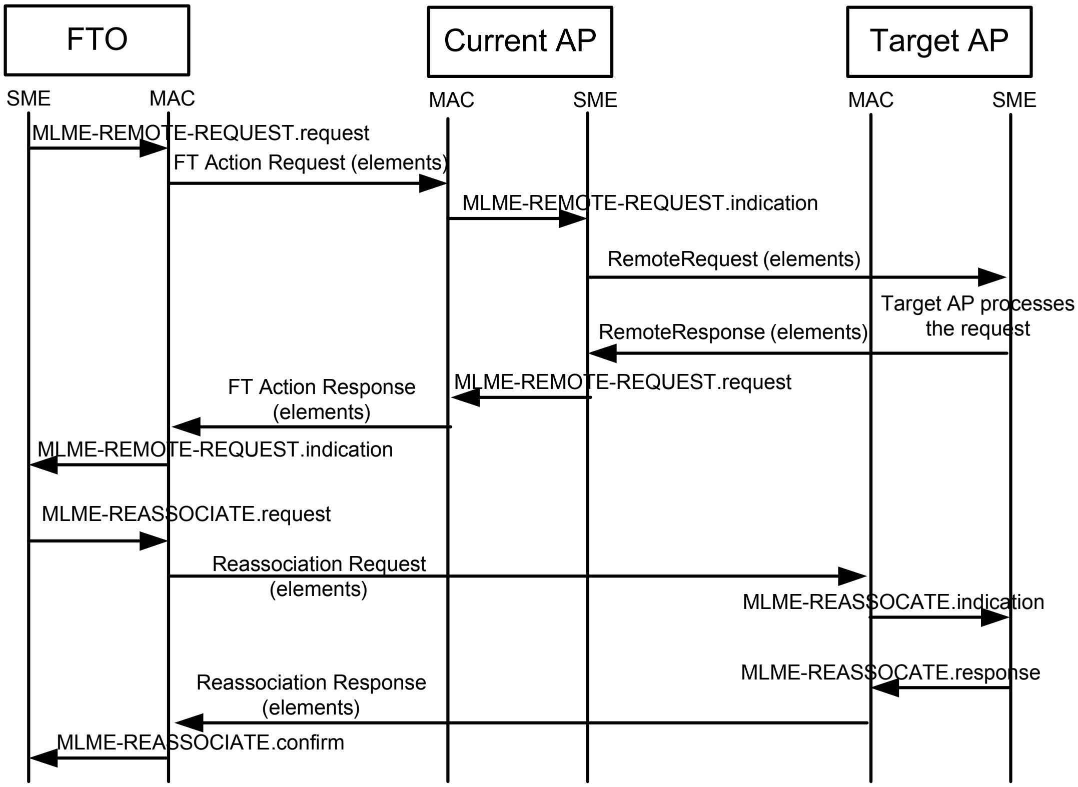

FTO
Current AP
Target AP
SME
MAC
MAC
SME
MAC
SME
MLME-REMOTE-REQUEST.request
FT Action Request (elements)
MLME-REMOTE-REQUEST.indication
RemoteRequest (elements)
Target AP processes the request
RemoteResponse (elements)
FT Action Response (elements)
MLME-REMOTE-REQUEST.request
MLME-REMOTE-REQUEST.indication
MLME-REASSOCIATE.request
Reassociation Request (elements)
MLME-REASSOCATE.indication
MLME-REASSOCATE.response
Reassociation Response (elements)
MLME-REASSOCIATE.confirm

Figure 12-6—MLME interfaces for over-the-DS FT Protocol messages

If the contents of the MDE received by the target AP do not match the contents advertised in the Beacon and Probe Response frames, the target AP shall reject the FT Request frame with status code 54 (i.e., Invalid MDE). If the contents of the RSNE do not indicate a negotiated AKM of Fast BSS Transition (suite type 00-0F-AC:3, 00-0F-AC:4, or 00-0F-AC:9), the AP shall reject the FT Request frame with status code 43 (i.e., Invalid AKMP). If the FTE in the FT Request frame contains an invalid R0KH-ID, the AP shall reject the FT Request frame with status code 55 (i.e., Invalid FTE). If the RSNE in the FT Request frame contains an invalid PMKR0Name, and the AP has determined that it is an invalid PMKR0Name, the AP shall reject the Authentication Request with status code 53 (i.e., Invalid PMKID). If the requested R0KH is not reachable, the AP shall respond to the FT Request frame with status code 28 (i.e., R0KH unreachable). The AP may reject the FT Request frame for limiting the FTO's reassociation to this AP by using the status code 37 ("This request has been declined"). If the FTO selects a pairwise cipher suite in the RSNE that is different from the ones used in the initial mobility domain association, then the AP shall reject the FT Request frame with status code 19 (i.e., Invalid Pairwise Cipher).

The STA Address field of the FT Response frame shall be set to the MAC address of the FTO, and the Target AP Address field of the FT Response frame shall be set to the BSSID of the target AP. The elements in the FT Response frame, and their required contents, shall be as given in 12.8.3. The Status Code field shall be a value from the options listed in 8.4.1.9.

The R1KH of the target AP uses the value of PMKR0Name and other information from the frame to calculate PMKR1Name. If the target AP does not have the key identified by PMKR1Name, it may retrieve that key from the R0KH identified by the STA. See 12.2. Upon receiving a new PMK-R1 for a STA, the target AP shall delete the prior PMK-R1 security association and PTKSAs derived from the prior PMK-R1.

The FTO and the target AP compute the PTK and PTKName using the PMK-R1, PMKR1Name, ANonce, and SNonce, as specified in 11.6.1.7.5. The PTKSA shall be deleted by the target AP if it does not receive a Reassociation Request frame from the FTO within the reassociation deadline timeout value.

If the FTO does not receive a response to the FT Request frame, it may reissue the request following the restrictions given for Authentication frames in 10.3. If the Status Code field value returned by the target AP is 0, indicating success, the FTO and target AP transition to State 2 (as defined in 10.3); the FTO may continue with reassociation (12.7.1). Handling of errors returned in the Status Code field shall be as specified for Authentication frames in 10.3.

12.5.4 Over-the-air FT Protocol authentication in a non-RSN

The over-the-air FT Protocol in a non-RSN is shown in Figure 12-7.

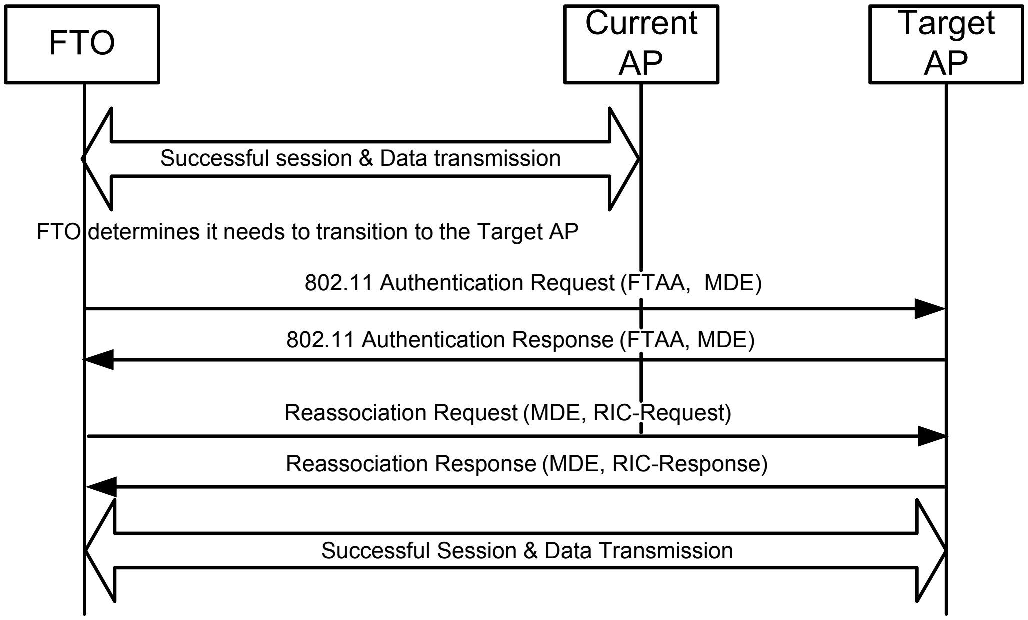

FTO
Current AP
Target AP
Successful session & Data transmission
FTO determines it needs to transition to the Target AP
802.11 Authentication Request (FTAA, MDE)
802.11 Authentication Response (FTAA, MDE)
Reassociation Request (MDE, RIC-Request)
Reassociation Response (MDE, RIC-Response)
Successful Session & Data Transmission

Figure 12-7—Over-the-air FT Protocol in a non-RSN

To perform an over-the-air fast BSS transition to a target AP in a non-RSN, the FTO and target AP shall perform the following exchange:

FTO→Target AP: Authentication-Request (FTAA, 0, MDE)
Target AP→FTO: Authentication-Response (FTAA, Status, MDE)

In the Authentication Request frame, the SA field of the message header shall be set to the MAC address of the FTO, and the DA field of the message header shall be set to the BSSID of the target AP. The elements in the frame, and their required contents, shall be as given in 12.8.2.

If the contents of the MDE received by the target AP do not match the contents advertised in the Beacon and Probe Response frames, the target AP shall reject the Authentication Request with status code 54 (i.e., Invalid MDE).

In the Authentication Response frame, the SA field of the message header shall be set to the BSSID of the target AP, and the DA field of the message header shall be set to the MAC address of the FTO. The Status Code field shall be a value from the options listed in 8.4.1.9. The elements in the frame, and their required contents, shall be as given in 12.8.3.

If the FTO does not receive a response to the Authentication Request frame, it may reissue the request following the restrictions given for Authentication frames in 10.3. If the Status Code field value returned by the target AP is 0, indicating success, the FTO and target AP transition to State 2 (as defined in 10.3); the

FTO may continue with reassociation (12.7.2). Handling of errors returned in the Status Code field shall be as specified in 10.3.

12.5.5 Over-the-DS FT Protocol authentication in a non-RSN

The over-the-DS FT Protocol in a non-RSN is shown in Figure 12-8.

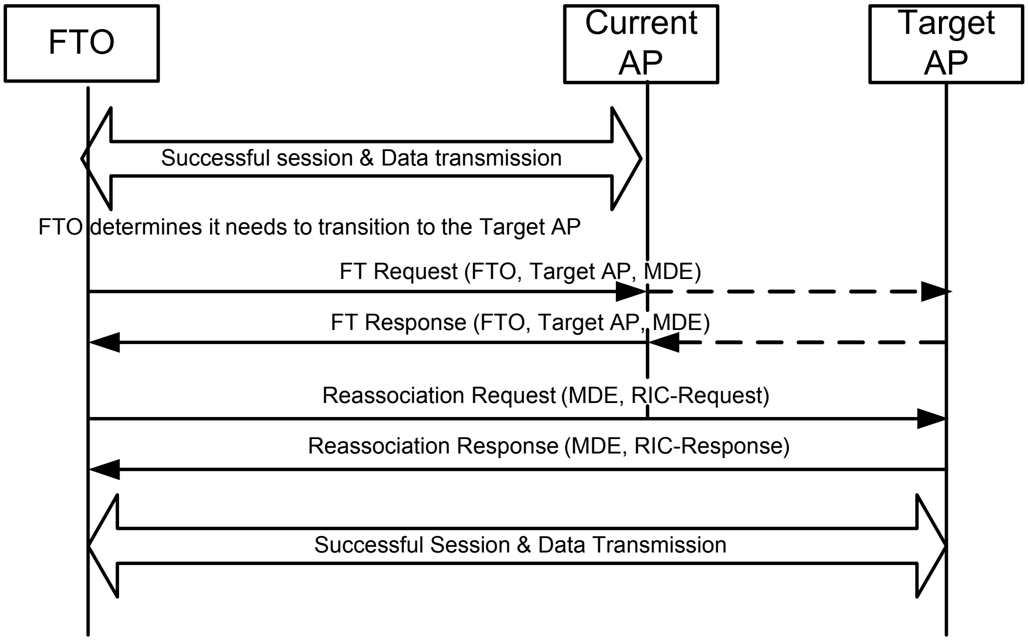

FTO
Current AP
Target AP
Successful session & Data transmission
FTO determines it needs to transition to the Target AP
FT Request (FTO, Target AP, MDE)
FT Response (FTO, Target AP, MDE)
Reassociation Request (MDE, RIC-Request)
Reassociation Response (MDE, RIC-Response)
Successful Session & Data Transmission

Figure 12-8—Over-the-DS FT Protocol in a non-RSN

To perform an over-the-DS fast BSS transition to a target AP in a non-RSN, the FTO and the target AP (through the current AP) shall perform the following exchange:

FTO→Target AP: FT Request(FTO, TargetAP, MDE)

Target AP→FTO: FT Response(FTO, TargetAP, Status, MDE)

The STA Address field of the FT Request frame shall be set to the MAC address of the FTO, and the Target AP Address field of the FT Request frame shall be set to the BSSID of the target AP. The elements in the FT Request frame, and their required contents, shall be as given in 12.8.2.

If the contents of the MDE received by the target AP do not match the contents advertised in the Beacon and Probe Response frames, the target AP shall reject the FT Request frame with status code 54 (i.e., Invalid MDE).

The STA Address field of the FT Response frame shall be set to the MAC address of the FTO, and the Target AP Address field of the FT Response frame shall be set to the BSSID of the target AP. The elements in the FT Response frame, and their required contents, shall be as given in 12.8.3. The Status Code field shall be a value from the options listed in 8.4.1.9.

If the FTO does not receive a response to the FT Request frame, it may reissue the request following the restrictions given for Authentication frames in 10.3. If the Status Code field value returned by the target AP is 0, indicating success, the FTO and target AP transition to State 2 (as defined in 10.3); the FTO may continue with reassociation (12.7.2). Handling of errors returned in the Status Code field shall be as specified for Authentication frames in 10.3.

12.6 FT Resource Request Protocol

12.6.1 Overview

The FT Resource Request Protocol involves an additional message exchange after the Authentication Request/Response frame, or FT Request/Response frame, and prior to reassociation.

APs capable of fast BSS transition may allow FTOs to request resources prior to reassociation. Availability of the FT Resource Request Protocol is advertised by the target AP in the MDE. If the Resource Request Protocol Capability subfield is 0, then the FTO shall not send an Authentication Confirm nor FT Confirm frame to the AP. An AP that receives an Authentication Confirm or FT Confirm frame from a STA and does not support the FT Resource Request Protocol shall respond with status code 38 (i.e., the request has not been successful as one or more parameters have invalid values).

The additional message exchange for the FT Resource Request Protocol shall be performed using the same method (over-the-air or over-the-DS) as was used for the Authentication Request/Response frame or FT Request/Response frame. An AP that receives an FT Confirm frame that did not previously receive an FT Request frame from the same STA shall reject the request with status code 52 (i.e., Invalid FT Action Frame Count). An AP that receives an Authentication Confirm frame that did not previously receive an Authentication Request frame from the same STA shall reject the request with status code 14 (i.e., Received an Authentication frame with authentication transaction sequence number out of expected sequence).

12.6.2 Over-the-air fast BSS transition with resource request

The over-the-air FT Resource Request Protocol in an RSN is shown in Figure 12-9.

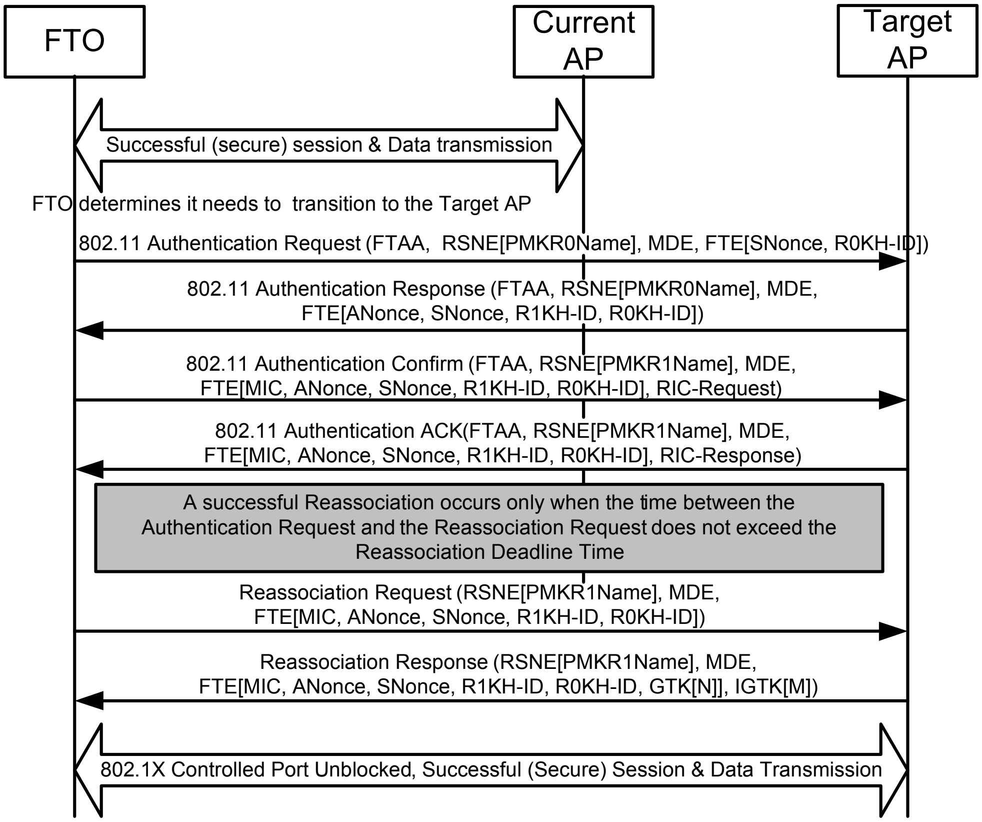

FTO
Current AP
Target AP
Successful (secure) session & Data transmission
FTO determines it needs to transition to the Target AP
802.11 Authentication Request (FTAA, RSNE[PMKR0Name], MDE, FTE[SNonce, R0KH-ID])
802.11 Authentication Response (FTAA, RSNE[PMKR0Name], MDE, FTE[ANonce, SNonce, R1KH-ID, R0KH-ID])
802.11 Authentication Confirm (FTAA, RSNE[PMKR1Name], MDE, FTE[MIC, ANonce, SNonce, R1KH-ID, R0KH-ID], RIC-Request)
802.11 Authentication ACK(FTAA, RSNE[PMKR1Name], MDE, FTE[MIC, ANonce, SNonce, R1KH-ID, R0KH-ID], RIC-Response)
A successful Reassociation occurs only when the time between the Authentication Request and the Reassociation Request does not exceed the Reassociation Deadline Time
Reassociation Request (RSNE[PMKR1Name], MDE, FTE[MIC, ANonce, SNonce, R1KH-ID, R0KH-ID])
Reassociation Response (RSNE[PMKR1Name], MDE, FTE[MIC, ANonce, SNonce, R1KH-ID, R0KH-ID, GTK[N]], IGTK[M])
802.1X Controlled Port Unblocked, Successful (Secure) Session & Data Transmission

Figure 12-9—Over-the-air FT Resource Request Protocol in an RSN

The over-the-air FT Resource Request Protocol in a non-RSN is shown in Figure 12-10.

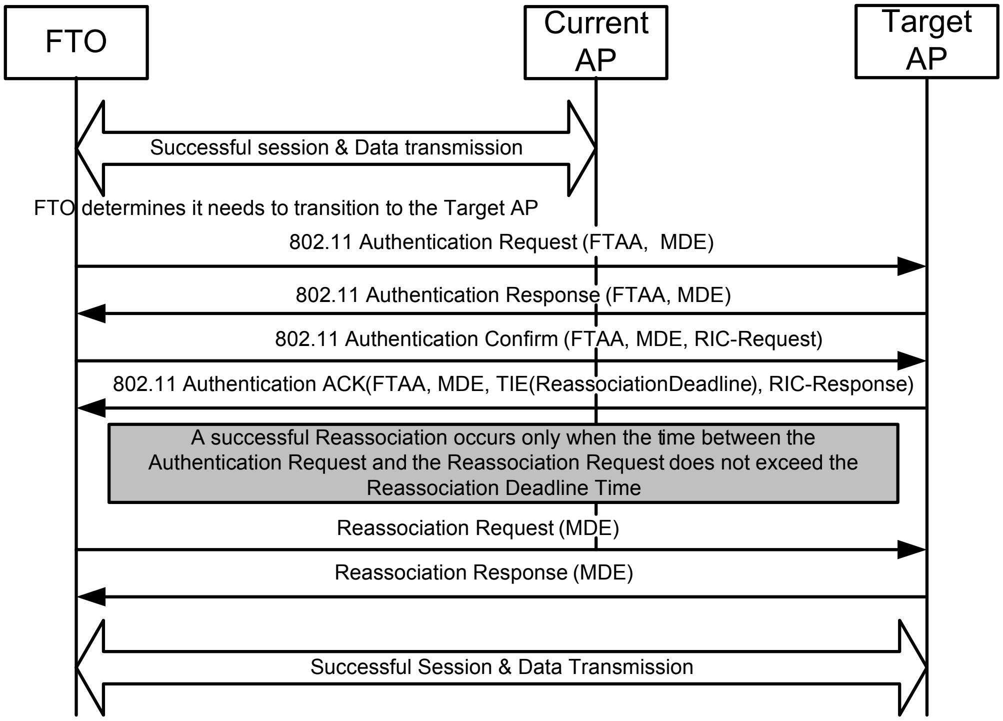

FTO
Current AP
Target AP
Successful session & Data transmission
FTO determines it needs to transition to the Target AP
802.11 Authentication Request (FTAA, MDE)
802.11 Authentication Response (FTAA, MDE)
802.11 Authentication Confirm (FTAA, MDE, RIC-Request)
802.11 Authentication ACK(FTAA, MDE, TIE(ReassociationDeadline), RIC-Response)
A successful Reassociation occurs only when the time between the Authentication Request and the Reassociation Request does not exceed the Reassociation Deadline Time
Reassociation Request (MDE)
Reassociation Response (MDE)
Successful Session & Data Transmission

Figure 12-10—Over-the-air FT Resource Request Protocol in a non-RSN

To perform an over-the-air FT Resource Request Protocol to a target AP, after completing the Authentication Request/Response exchange given in 12.5.2 or 12.5.4, the FTO and target AP shall perform the following exchange:

FTO→Target AP: Authentication-Confirm (FTAA, 0, RSNE[PMKR1Name], MDE, FTE[MIC, ANonce, SNonce, R1KH-ID, R0KH-ID], RIC-Request)

Target AP→FTO: Authentication-Ack (FTAA, Status, RSNE[PMKR1Name], MDE, FTE[MIC, ANonce, SNonce, R1KH-ID, R0KH-ID], RIC-Response)

The SME of the FTO initiates the resource request exchange through the use of the primitive MLME-RESOURCE-REQUEST.request primitive, and the SME of the AP responds with MLME-RESOURCE-REQUEST.response primitive.

In the Authentication Confirm frame, the SA field of the message header shall be set to the MAC address of the FTO, and the DA field of the message header shall be set to the BSSID of the target AP. In a non-RSN, the FTE and RSNE shall not be present. The elements in the frame, the element contents, and MIC calculation shall be as given in 12.8.4.

If the contents of the MDE received by the target AP do not match the contents advertised in the Beacon and Probe Response frames, the target AP shall reject the Authentication Confirm frame with status code 54 (i.e., Invalid MDE).

In an RSN, the R1KH of the target AP verifies the MIC in the FTE in the Authentication Confirm frame and shall discard the request if it is incorrect. If the FTE in the Authentication Confirm frame contains a different R0KH-ID, R1KH-ID, ANonce, or SNonce, the AP shall reject the Authentication Confirm frame with status code 55 (i.e., Invalid FTE). If the RSNE in the Authentication Confirm frame contains an invalid

PMKR1Name, the AP shall reject the Authentication Confirm frame with status code 53 (i.e., Invalid PMKID).

In the Authentication Ack frame, the SA field of the message header shall be set to the BSSID of the target AP, and the DA field of the message header shall be set to the MAC address of the FTO. In a non-RSN, the FTE and RSNE shall not be present. The Status Code field shall be a value from the options listed in 8.4.1.9. The elements in the frame, the element contents, and MIC calculation shall be as given in 12.8.5.

In an RSN, the S1KH of the FTO verifies the MIC in the FTE in the Authentication Ack frame and shall discard the response if the MIC is incorrect.

The FTO may make a request for resources by including a RIC-Request (see 12.11) in the Authentication Confirm frame. The RIC-Request is generated by the procedures of 12.11.3.1, and the RIC-Response is generated by the procedures of 12.11.3.2.

If the value of the Status Code field returned by the target AP in the Authentication Ack frame is nonzero, then the FTO shall abandon this transition attempt.

In an RSN, on successful completion of the FT authentication exchange of the FT Resource Request Protocol, the PTKSA has been established and proven live. The key replay counter shall be initialized to 0, and the subsequent EAPOL-Key frames (e.g., GTK and IGTK updates) shall use the key replay counter to ensure they are not replayed. The PTKSA shall be deleted by the target AP if it does not receive a Reassociation Request frame from the FTO within the reassociation deadline timeout value.

In a non-RSN, the Authentication Ack frame contains a TIE with a reassociation deadline. If the FTO does not send a Reassociation Request frame to the target AP within that interval, the FTO shall abandon this transition attempt.

The exchange between the FTO and the target AP may continue with reassociation (12.7.1 or 12.7.2).

12.6.3 Over-the-DS fast BSS transition with resource request

The over-the-DS FT Resource Request Protocol in an RSN is shown in Figure 12-11.

The over-the-DS FT Resource Request Protocol in a non-RSN is shown in Figure 12-12.

To perform an Over-the-DS FT Resource Request Protocol to a target AP, after completing the FT Request/Response frame exchange given in 12.5.3 or 12.5.5, the FTO and target AP (through the current AP) shall perform the following exchange, using the mechanism described in 12.10:

FTO→Target AP: FT Confirm (FTO, TargetAP, RSNE[PMKR1Name], MDE, FTE[MIC, ANonce, SNonce, R1KH-ID, R0KH-ID], RIC-Request)

Target AP→FTO: FT Ack (FTO, TargetAP, Status, RSNE[PMKR1Name], MDE, FTE[MIC, ANonce, SNonce, R1KH-ID, R0KH-ID], TIE[ReassociationDeadline], RIC-Response)

The SME of the FTO initiates the FT Confirm frame to the target AP by issuing a MLME-REMOTE-REQUEST.request primitive with parameters including the contents of the FT Confirm frame (FT Action frame with an FT Action field value indicating FT Confirm) to be sent. The MAC of the FTO transmits this Action frame. For processing at the current AP and target AP, see 12.10. When the MAC of the FTO receives the FT Ack frame (FT Action frame with an FT Action field value indicating FT Ack), it passes it to the SME by use of an MLME-REMOTE-REQUEST.indication primitive, with parameters including the contents of the received Action frame.

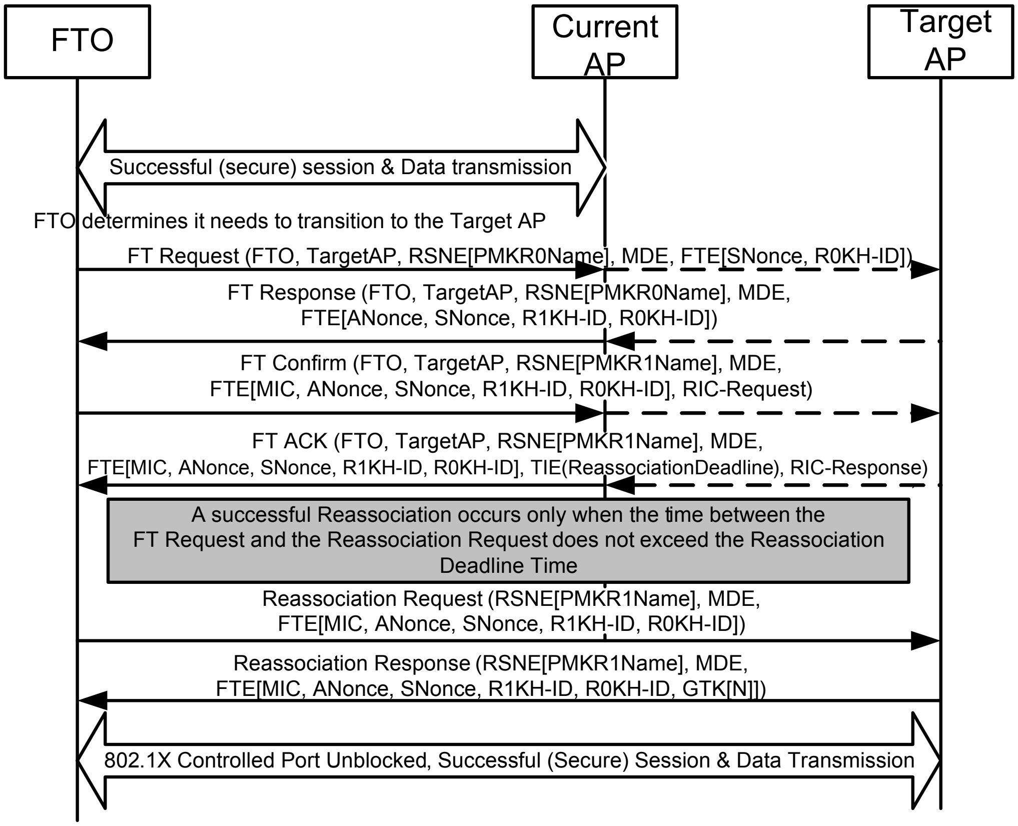

FTO
Current AP
Target AP
Successful (secure) session & Data transmission
FTO determines it needs to transition to the Target AP
FT Request (FTO, TargetAP, RSNE[PMKR0Name], MDE, FTE[SNonce, R0KH-ID])
FT Response (FTO, TargetAP, RSNE[PMKR0Name], MDE, FTE[ANonce, SNonce, R1KH-ID, R0KH-ID])
FT Confirm (FTO, TargetAP, RSNE[PMKR1Name], MDE, FTE[MIC, ANonce, SNonce, R1KH-ID, R0KH-ID], RIC-Request)
FT ACK (FTO, TargetAP, RSNE[PMKR1Name], MDE, FTE[MIC, ANonce, SNonce, R1KH-ID, R0KH-ID], TIE(ReassociationDeadline), RIC-Response)
A successful Reassociation occurs only when the time between the FT Request and the Reassociation Request does not exceed the Reassociation Deadline Time
Reassociation Request (RSNE[PMKR1Name], MDE, FTE[MIC, ANonce, SNonce, R1KH-ID, R0KH-ID])
Reassociation Response (RSNE[PMKR1Name], MDE, FTE[MIC, ANonce, SNonce, R1KH-ID, R0KH-ID, GTK[N]])
802.1X Controlled Port Unblocked, Successful (Secure) Session & Data Transmission

Figure 12-11—Over-the-DS FT Resource Request Protocol in an RSN

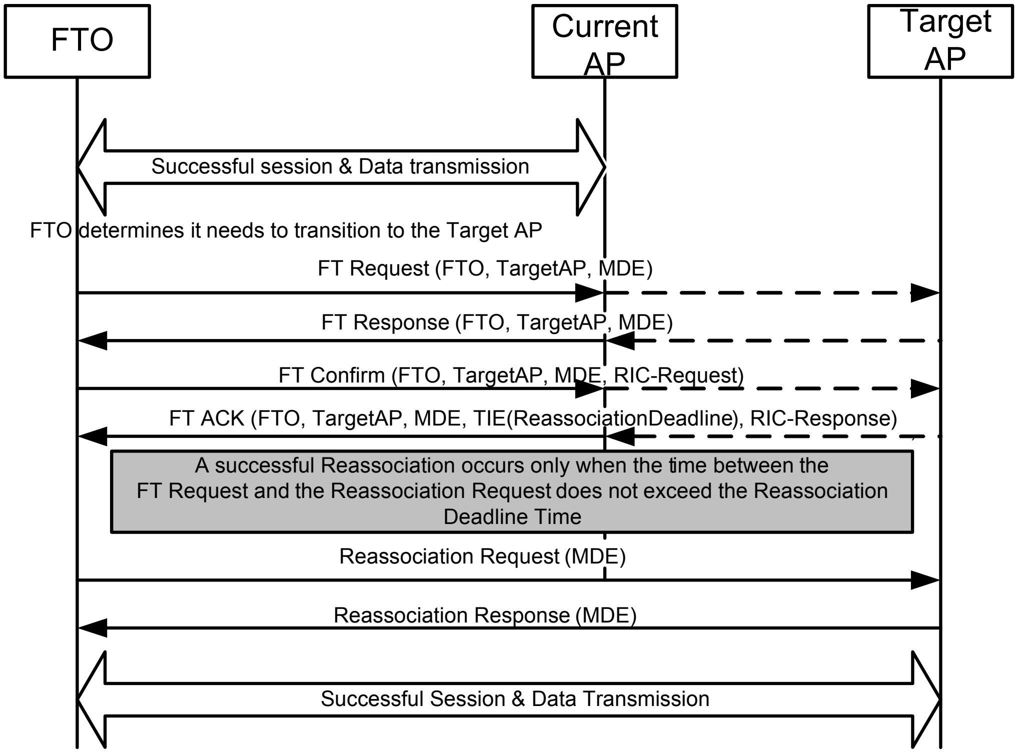

FTO
Current AP
Target AP
Successful session & Data transmission
FTO determines it needs to transition to the Target AP
FT Request (FTO, TargetAP, MDE)
FT Response (FTO, TargetAP, MDE)
FT Confirm (FTO, TargetAP, MDE, RIC-Request)
FT ACK (FTO, TargetAP, MDE, TIE(ReassociationDeadline), RIC-Response)
A successful Reassociation occurs only when the time between the FT Request and the Reassociation Request does not exceed the Reassociation Deadline Time
Reassociation Request (MDE)
Reassociation Response (MDE)
Successful Session & Data Transmission

Figure 12-12—Over-the-DS FT Resource Request Protocol in a non-RSN

The STA Address field of the FT Confirm frame shall be set to the MAC address of the FTO, and the Target AP Address field of the FT Confirm frame shall be set to the BSSID of the target AP. The elements in the FT Confirm frame, the element contents, and the MIC calculation shall be as given in 12.8.4. In a non-RSN, the FTE and RSNE shall not be present.

If the contents of the MDE received by the target AP do not match the contents advertised in the Beacon and Probe Response frames, the target AP shall reject the FT Confirm frame with status code 54 (i.e., Invalid MDE).

In an RSN, the R1KH of the target AP verifies the MIC in the FTE and shall discard the request if it is incorrect. If the FTE in the FT Confirm frame contains a different R0KH-ID, R1KH-ID, ANonce, or SNonce from the values sent in the FT Response frame, the AP shall reject the FT Confirm frame with status code 55 (i.e., Invalid FTE). If the RSNE in the FT Confirm frame contains an invalid PMKR1Name, the AP shall reject the FT Confirm frame with status code 53 (i.e., Invalid PMKID).

The STA Address field of the FT Ack frame shall be set to the MAC address of the FTO, and the Target AP Address field of the FT Ack frame shall be set to the BSSID of the target AP. The elements in the FT Ack frame, the element contents, and the MIC calculation shall be as given in 12.8.5. In a non-RSN, the FTE and RSNE shall not be present. The Status Code field value shall be a value from the options listed in 8.4.1.9, and a TIE may appear.

In an RSN, the S1KH of the FTO verifies the MIC in the FTE in the FT Ack frame and shall discard the response if the MIC is incorrect.

The FTO may make a request for resources by including a RIC-Request (see 12.11) in the FT Confirm frame. The RIC-Request is generated by the procedures of 12.11.3.1, and the RIC-Response is generated by the procedures of 12.11.3.2.

In order to recover from over-the-DS packet losses, the FTO may retransmit the FT Confirm frame until the reassociation deadline time is reached. If the FTO does not receive a response to the FT Confirm frame or if the value of the Status Code field returned by the target AP in the FT Ack frame is nonzero, then the FTO shall abandon this transition attempt.

In an RSN, on successful completion of the FT Confirm/Acknowledgment frame exchange, the PTKSA has been established and proven live. The key replay counter shall be initialized to 0, and the subsequent EAPOL-Key frames (e.g., GTK and IGTK updates) shall use the key replay counter to ensure they are not replayed. The PTKSA shall be deleted by the target AP if it does not receive a Reassociation Request frame from the FTO within the reassociation deadline timeout value. Resource request procedures are specified in 12.11.

In a non-RSN, the FT Ack frame contains a TIE with a reassociation deadline. If the FTO does not send a Reassociation Request frame to the target AP within that interval, the FTO shall abandon this transition attempt.

The exchange between the FTO and the target AP may continue with reassociation (12.7.1 or 12.7.2).

12.7 FT reassociation

12.7.1 FT reassociation in an RSN

If the FTO does not send a Reassociation Request frame to the target AP within the reassociation deadline interval received during the FT initial mobility domain association, the target AP may delete the PTKSA, and the FTO shall abandon this transition attempt.

The FTO shall perform a reassociation directly with the target AP via the following exchange:

FTO→Target AP: Reassociation Request(RSNE[PMKR1Name], MDE, FTE[MIC, ANonce, SNonce, R1KH-ID, R0KH-ID], RIC-Request)

Target AP→FTO: Reassociation Response(RSNE[PMKR1Name], MDE, FTE[MIC, ANonce, SNonce, R1KH-ID, R0KH-ID, GTK[N], IGTK[M]], RIC-Response)

The SME of the FTO initiates the reassociation through the use of the MLME-REASSOCIATE.request primitive. The SME of the AP responds to the indication with MLME-REASSOCIATE.response primitive. See 10.3.5.

In the Reassociation Request frame, the SA field of the message header shall be set to the MAC address of the FTO, and the DA field of the message header shall be set to the BSSID of the target AP. The elements in the frame, the element contents, and the MIC calculation shall be as given in 12.8.4.

The R1KH of the target AP verifies the MIC in the FTE in the Reassociation Request frame and shall discard the request if the MIC is incorrect. If the contents of the MDE received by the target AP do not match the contents advertised in the Beacon and Probe Response frames, the target AP shall reject the Reassociation Request frame with status code 54 (i.e., Invalid MDE). If the FTE in the Reassociation Request frame contains a different R0KH-ID, R1KH-ID, ANonce, or SNonce, the AP shall reject the Reassociation Request frame with status code 55 (i.e., Invalid FTE). If the RSNE in the Reassociation Request frame contains an invalid PMKR1Name, the AP shall reject the Reassociation Request frame with status code 53 (i.e., Invalid PMKID).

In the Reassociation Response frame, the SA field of the message header shall be set to the BSSID of the target AP, and the DA field of the message header shall be set to the MAC address of the FTO. The Status Code field shall be a value from the options listed in 8.4.1.9. The elements in the frame, the element contents, and the MIC calculation shall be as given in 12.8.5.

The S1KH of the FTO verifies the MIC in the FTE in the Reassociation Response frame and shall discard the response if the MIC is incorrect.

If an FTO is performing a reassociation exchange as part of the FT Resource Request Protocol, then the FTO shall not include the RIC-Request in the Reassociation Request frame, and the AP shall not include the RIC-Response in the Reassociation Response frame. If the reassociation exchange is part of the FT Resource Request Protocol and the AP is unable to honor the resources that have been placed in the accepted state for that FTO, then the AP shall reject the Reassociation Request frame and may use status code 33 (i.e., Association denied because QoS AP has insufficient bandwidth to handle another QoS STA).

If the FTO did not utilize the FT Resource Request Protocol, the FTO may make a request for resources by including a RIC-Request (see 12.11) in the Reassociation Request frame. The RIC-Request is generated by the procedures of 12.11.3.1, and the RIC-Response is generated by the procedures of 12.11.3.2.

If the Status Code field value returned by the target AP in the response is 1 (i.e., Unspecified failure), 14 (i.e., Authentication transaction sequence number out of sequence), or 16 (i.e., Authentication rejected due to timeout waiting for next frame in sequence), then the FTO shall abandon this transition attempt. Handling of other errors returned in the Status Code field shall be as specified in 10.3.

Upon a successful reassociation, the PTKSA has been established and proven live. The SME of the AP shall open the IEEE 802.1X Controlled Port. The FTO shall transition to State 4 (as defined in 10.3). If the target AP is distinct from the previous AP, the FTO shall enter State 1 with respect to the previous AP.

Upon a successful reassociation, the FTO shall delete any corresponding PTKSA with its previous AP. The SME of the FTO shall issue an MLME-DELETEKEYS.request primitive to delete the pairwise keys with the previous AP, and the FTO and the AP shall issue a MLME-SETKEYS.request primitive and MLME-SETPROTECTION.request primitive to install the pairwise keys. The PTK key lifetime timer shall be initialized with the value calculated as the difference between the TIE[KeyLifetime] sent in Message 3 of the FT initial mobility domain association and the time since the completion of the FT 4-Way Handshake during the FT initial mobility domain association.

When the IEEE 802.1X Controlled Port is opened, the EAPOL-Key frame replay counter shall be initialized to 0. The R1KH shall increment the key replay counter on each successive EAPOL-Key frame that it transmits.

12.7.2 FT reassociation in a non-RSN

The FTO shall perform a reassociation with the target AP via the following exchange:

FTO→Target AP: Reassociation Request(MDE, RIC-Request)
Target AP→FTO: Reassociation Response(MDE, RIC-Response)

The SME of the FTO initiates the reassociation through the use of the MLME-REASSOCIATE.request primitive. The SME of the AP responds to the indication with MLME-REASSOCIATE.response primitive. See 10.3.5.

In the Reassociation Request frame, the SA field of the message header shall be set to the MAC address of the FTO, and the DA field of the message header shall be set to the BSSID of the target AP. The elements in Reassociation Request frame, and their required contents, shall be as given in 12.8.4.

If the contents of the MDE received by the target AP do not match the contents advertised in the Beacon and Probe Response frames, the target AP shall reject the Reassociation Request frame with status code 54 (i.e., Invalid MDE).

In the Reassociation Response frame, the SA field of the message header shall be set to the BSSID of the target AP, and the DA field of the message header shall be set to the MAC address of the FTO. The elements in Reassociation Response frame, and their required contents, shall be as given in 12.8.5. The Status Code field shall be a value from the options listed in 8.4.1.9.

If the FTO is performing a reassociation exchange as part of the FT Resource Request Protocol, then the FTO shall not include the RIC-Request in the Reassociation Request frame, and the AP shall not include the RIC-Response in the Reassociation Response frame.

If the FTO did not utilize the FT Resource Request Protocol, the FTO may make a request for resources by including a RIC-Request (see 12.11) in the Reassociation Request frame. The RIC-Request is generated by the procedures of 12.11.3.1, and the RIC-Response is generated by the procedures of 12.11.3.2.

If the Status Code field value returned by the target AP in the response is 1 (i.e., Unspecified failure), 14 (i.e., Authentication transaction sequence number out of sequence), or 16 (i.e., Authentication rejected due to timeout waiting for next frame in sequence), then the FTO shall abandon this transition attempt. Handling of other errors returned in the Status Code field shall be as specified in 10.3.

If the AP has dot11RSNAActivated equal to true, upon a successful reassociation, the SME shall open the IEEE 802.1X Controlled Port.

Upon a successful reassociation, the target AP and the FTO shall transition to State 4 (as defined in 10.3). If the target AP is distinct from the previous AP, then the FTO shall enter State 1 with respect to the previous AP.

12.8 FT authentication sequence

12.8.1 Overview

The FT authentication sequence comprises four sets of FT elements. Each set of FT elements is referred to in 12.8 as a *message*. These messages are included in the FT Protocol frames or FT Resource Request Protocol frames to initiate a fast BSS transition. The FT authentication sequence is always initiated by the FTO and responded to by the target AP.

In an RSN, the first two messages in the sequence allow the FTO and target AP to provide association instance identifiers, SNonce and ANonce, respectively. SNonce and ANonce are chosen randomly or pseudorandomly and are used to generate a fresh PTK. The first two messages also enable the target AP to provision the PMK-R1 and the FTO and target AP to compute the PTK. The third and fourth messages demonstrate liveness of the peer, authenticate the elements, and enable an authenticated resource request.

When an FTO invokes the FT Protocol, then the first two messages of the sequence are both carried in Authentication frames or both carried in Action frames, and these messages are described in 12.8.2 and 12.8.3. The third and fourth messages in the sequence are carried in the Reassociation Request and Reassociation Response frames and are described in 12.8.4 and 12.8.5.

When the FTO invokes the FT Resource Request Protocol, then the first four messages of the sequence are all carried in Authentication frames or all carried in Action frames, and these messages are described in 12.8.2 to 12.8.5. The fifth and sixth frames of the FT Resource Request Protocol are carried in the Reassociation Request frame and Reassociation Response frame and are described in 12.8.4 and 12.8.5.

Regardless of the transport mechanism, the information contained in the FT authentication sequence consists of the set of elements shown in Table 12-1.

Table 12-1—FT authentication elements

Information	Presence in Authentication Sequence messages	Description
RSN	The RSNE is present if dot11RSNAActivated is true.	8.4.2.27
Mobility Domain	The Mobility Domain element is present.	8.4.2.49
Fast BSS Transition	The Fast BSS Transition element is present if dot11RSNAActivated is true.	8.4.2.50
Timeout Interval (reassociation deadline)	The Timeout Interval element is optionally present in the fourth message of the sequence if dot11RSNAActivated is true.	8.4.2.51
RIC	The RIC Data element is optionally present in the third and fourth messages.	8.4.2.52

The first message is used by the FTO to initiate a fast BSS transition. When RSNA is enabled, the FTO shall include the R0KH-ID and the SNonce in the FTE and the PMKR0Name in the RSNE. The target AP can use the PMKR0Name to derive the PMKR1Name, and if the target AP does not have the PMK-R1 identified by PMKR1Name, it may attempt to retrieve that key from the R0KH identified by R0KH-ID. See 12.2. The FTO includes a fresh SNonce as its contribution to the association instance identifier and to provide key separation of the derived PTK; it is selected randomly to serve as a challenge that demonstrates the liveness of the peer in the fourth message.

The second message is used by the target AP to respond to the requesting FTO. The target AP provides the key holder identifiers and key names used to generate the PTK. The target AP also includes a fresh ANonce as its contribution to the association instance identifier and to provide key separation of the derived PTK. The response includes a status code.

In an RSN, the third message is used by the FTO to assert to the target AP that it has a valid PTK. If no resources are required, then the FTO omits inclusion of the RIC.

The fourth message is used by the target AP to respond to the requesting FTO. This message serves as final confirmation of the transition, establishes that the AP possesses the PMK-R1 and is participating in this association instance, and protects against downgrade attacks. Note, however, that the RIC is absent if no resources were requested in the third message. This also includes a status code and may include a reassociation deadline.

12.8.2 FT authentication sequence: contents of first message

The RSNE shall be present only if dot11RSNAActivated is true. If present, the RSNE shall be set as follows:

- Version field shall be set to 1.
- PMKID Count field shall be set to 1.
- PMKID List field shall contain the PMKR0Name.
- All other fields shall be as specified in 8.4.2.27 and 11.5.3.

The MDE shall contain the MDID field and the FT Capability and Policy field settings obtained from the target AP, as advertised by the target AP in Beacon and Probe Response frames. The MDID shall be identical to that obtained during the FT initial mobility domain association exchange.

The FTE shall be present only if dot11RSNAActivated is true. If present, the FTE shall be set as follows:

- R0KH-ID shall be the value of R0KH-ID obtained by the FTO during its FT initial mobility domain association exchange.
- SNonce shall be set to a value chosen randomly by the FTO, following the recommendations of 11.6.5.
- All other fields shall be set to 0.

12.8.3 FT authentication sequence: contents of second message

If the status code is 0, then the following rules apply.

The RSNE shall be present only if dot11RSNAActivated is true. If present, the RSNE shall be set as follows:

- Version field shall be set to 1.
- PMKID Count field shall be set to 1.
- PMKID List field shall be set to the value contained in the first message of this sequence.
- All other fields shall be identical to the contents of the RSNE advertised by the AP in Beacon and Probe Response frames.

The MDE shall contain the MDID and FT Capability and Policy fields. This element shall be the same as the MDE advertised by the target AP in Beacon and Probe Response frames.

The FTE shall be present only if dot11RSNAActivated is true. If present, the FTE shall be set as follows:

- R0KH-ID shall be identical to the R0KH-ID provided by the FTO in the first message.

- R1KH-ID shall be set to the R1KH-ID of the target AP, from dot11FTR1KeyHolderID.
- ANonce shall be set to a value chosen randomly by the target AP, following the recommendations of 11.6.5.
- SNonce shall be set to the value contained in the first message of this sequence.
- All other fields shall be set to 0.

12.8.4 FT authentication sequence: contents of third message

The RSNE shall be present only if dot11RSNAActivated is true. If present, the RSNE shall be set as follows:

- Version field shall be set to 1.
- PMKID Count field shall be set to 1.
- PMKID field shall contain the PMKR1Name.
- All other fields shall be as specified in 8.4.2.27 and 11.5.3.

The MDE shall contain the MDID and FT Capability and Policy fields. This element shall be identical to the MDE contained in the first message of this sequence.

The FTE shall be present only if dot11RSNAActivated is true. If present, the FTE shall be set as follows:

- ANonce, SNonce, R0KH-ID, and R1KH-ID shall be set to the values contained in the second message of this sequence.
- The Element Count field of the MIC Control field shall be set to the number of elements protected in this frame (variable).
- When the negotiated AKM is 00-0F-AC:3, 00-0F-AC:4, or 00-0F-AC:9, the MIC shall be calculated using the KCK and the AES-128-CMAC algorithm. The output of the AES-128-CMAC shall be 128 bits.
- The MIC shall be calculated on the concatenation of the following data, in the order given here:
  - FTO’s MAC address (6 octets)
  - Target AP’s MAC address (6 octets)
  - Transaction sequence number (1 octet), which shall be set to the value 5 if this is a Reassociation Request frame and, otherwise, set to the value 3
  - RSNE
  - MDE
  - FTE, with the MIC field of the FTE set to 0
  - Contents of the RIC-Request (if present)
- All other fields shall be set to 0.

If resources are being requested by the FTO, then a sequence of elements forming the RIC-Request shall be included.

12.8.5 FT authentication sequence: contents of fourth message

If the status code is 0, then the following rules apply.

The RSNE shall be present only if dot11RSNAActivated is true. If present, the RSNE shall be set as follows:

- Version field shall be set to 1.
- PMKID Count field shall be set to 1.

- PMKID field shall contain the PMKR1Name
- All other fields shall be identical to the contents of the RSNE advertised by the target AP in Beacon and Probe Response frames.

The MDE shall contain the MDID and FT Capability and Policy fields. This element shall be identical to the MDE contained in the second message of this sequence.

The FTE shall be present only if dot11RSNAActivated is true. If present, the FTE shall be set as follows:

- ANonce, SNonce, R0KH-ID, and R1KH-ID shall be set to the values contained in the second message of this sequence.
- The Element Count field of the MIC Control field shall be set to the number of elements protected in this frame (variable).
- When this message of the authentication sequence appears in a Reassociation Response frame, the Optional Parameter(s) field in the FTE may include the GTK and IGTK subelements. If a GTK or an IGTK are included, the Key field of the subelement shall be encrypted using KEK and the NIST AES key wrap algorithm. The Key field shall be padded before encrypting if the key length is less than 16 octets or if it is not a multiple of 8. The padding consists of appending a single octet 0xdd followed by zero or more 0x00 octets. When processing a received message, the receiver shall ignore this trailing padding. Addition of padding does not change the value of the Key Length field. Note that the length of the encrypted Key field can be determined from the length of the GTK or IGTK subelement.
- When the negotiated AKM is 00-0F-AC:3, 00-0F-AC:4, or 00-0F-AC:9, the MIC shall be calculated using the KCK and the AES-128-CMAC algorithm. The output of the AES-128-CMAC algorithm shall be 128 bits.
- The MIC shall be calculated on the concatenation of the following data, in the order given here:
  - FTO’s MAC address (6 octets)
  - Target AP’s MAC address (6 octets)
  - Transaction sequence number (1 octet), which shall be set to the value 6 if this is a Reassociation Response frame or, otherwise, set to the value 4
  - RSNE
  - MDE
  - FTE, with the MIC field of the FTE set to 0
  - Contents of the RIC-Response (if present)
- All other fields shall be set to 0.

If this message is other than a Reassociation Response frame and dot11RSNAActivated is false, a TIE may appear. If this message is other than a Reassociation Response frame, includes a RIC-Response, and dot11RSNAActivated is false, then a timeout interval shall appear. If it appears, it shall be set as follows:

- Timeout Interval Type field shall be set to 1 (reassociation deadline)
- Timeout Interval Value field shall be set to the reassociation deadline time.

If resources were requested by the FTO, then a RIC-Response shall be included.

12.9 FT security architecture state machines

12.9.1 Introduction

The FT state machines describe the interaction between the RSNA key management and 802.11 architectural components.

RSNA key management uses the MA-UNITDATA service primitives to send/receive EAPOL-Key frames for FT initial association; MLME interfaces described below for FT methods; and MLME-SETKEYS, MLME-DELETEKEYS, and MLME-SETPROTECTION primitives. FT key management uses the following primitives for key management delivery and reception:

— The MLME-REMOTE-REQUEST primitives for FT key management over the DS
— The MLME-AUTHENTICATE primitives for FT key management over the air
— The MLME-RESOURCE-REQUEST primitives for FT resource request over the air
— The MLME-REASSOCIATE primitives for FT key management over the air and over the DS

Some of the state machine design considerations are as follows:

— Details of error handling are not included in the state machines. See 12.4, 12.5, and 12.6.
— Retransmission of FT Authentication and (Re)Association frames are not included in the state machines; see 12.5 and 12.6.

Various signals are used to communicate between the R0KH and the R1KH state machines. Note that these interactions may be between separate entities, rather than within a single SME. These interactions are as follows:

— In the R0KH state machine, FT-PMKR1-SA (PMKR1-SA) sends a PMK-R1 PMKSA to the R1KH.
— In the R1KH state machine, FT-FULL-AUTH (R1KH-ID) requests a key from the R0KH.
— In the R1KH state machine, FT-Transition-Auth (R1KH-ID) requests a key from the R0KH that was used for the initial mobility domain association.

The interactions between the R0KH and IEEE 802.1X, between the R1KH and IEEE 802.1X, and between the S1KH and IEEE 802.1X occur within the SME. At both the target AP and at the FTO, the R1KH and S1KH initialize the IEEE 802.1X EAPOL state machines in the respective SMEs. The Controlled Port is opened without an EAP exchange when the reassociation completes.

12.9.2 R0KH state machine

12.9.2.1 General

There is one R0KH state machine, which includes FT key management.

The state diagram in Figure 12-13 consists of a set of states that handle R0KH functions including key hierarchy instantiation, key generation, and cleanup. This state machine interacts with the R1KH state machine.

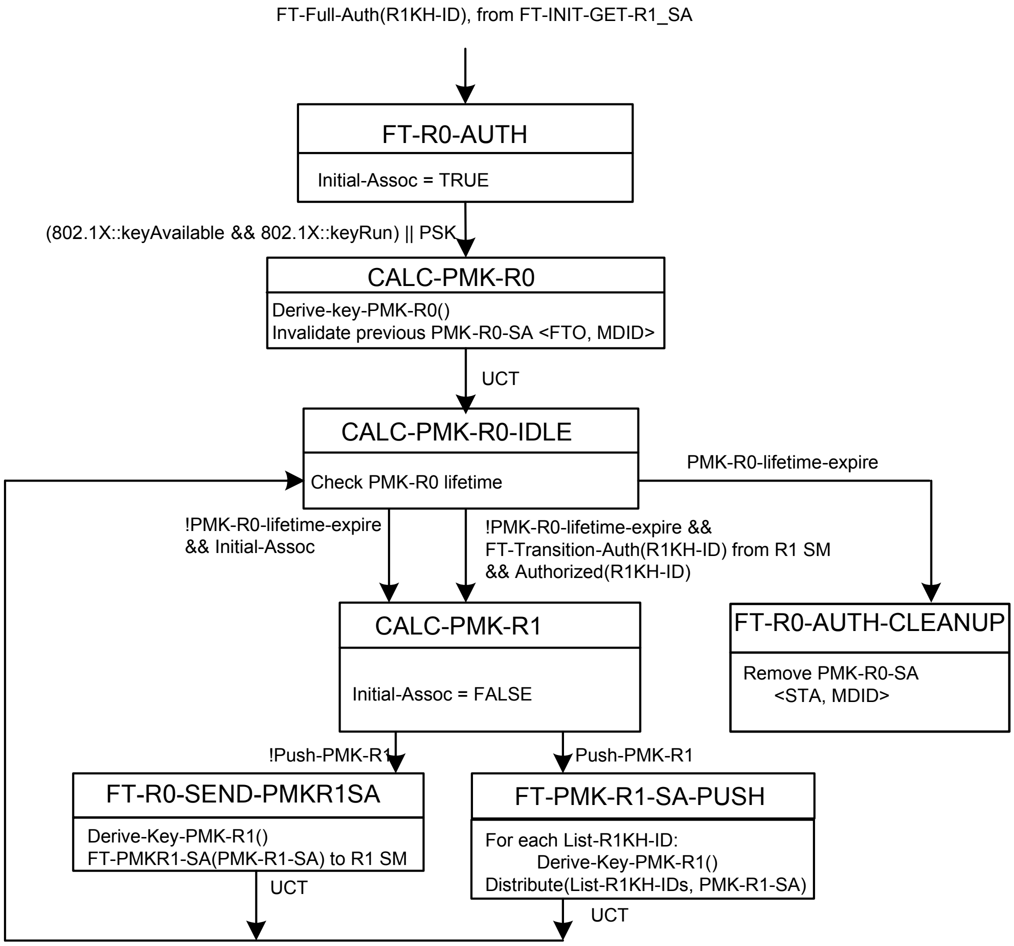


Figure 12-13—R0KH state machine

12.9.2.2 R0KH state machine states

The following list summarizes the states of the R0KH state machine:

- **CALC-PMK-R0**: This state is entered after the MSK from EAP authentication or PSK is available.
- **CALC-PMK-R0-IDLE**: This state is an intermediate state for the R0KH to wait for new requests from R1KHs.
- **CALC-PMK-R1**: For FT initial association, this state is entered as an unconditional transfer. For FT methods, this state is entered through the event from an R1KH state machine.
- **FT-PMK-R1-SA-PUSH**: This state is entered if Push-PMK-R1 is set to TRUE. PMK-R1s are derived and distributed to all the configured R1KHs.
- **FT-R0-AUTH-CLEANUP**: This state is entered when the PMK-R0 lifetime expires.
- **FT-R0-AUTH**: This state is entered through the event from the R1KH state machine. The R1KH state machine sends this event when it determines that a new PMK-R0 is needed.
- **FT-R0-SEND-PMKR1SA**: This state is entered from CALC-PMK-R1 when a request for the PMK-R1 security association is received from an R1KH. PMK-R1 security association is derived and distributed to the requesting R1KH.

12.9.2.3 R0KH state machine variables

The following list summarizes the variables used by the R0KH state machine:

- *Initial-Assoc* – This variable is used to indicate whether the current authentication is the initial association, in order to trigger the initial derivation of PMK-R1.
- *List-R1KH-IDs* – This variable contains a list of all of the R1KH-IDs in the mobility domain. This list is populated by the key distribution protocol as required in 12.2.
- *PMK-R0-lifetime-expire* – This variable is set to TRUE when PMK-R0 lifetime is deemed expired.
- *PSK* – This variable is set to TRUE when authentication is performed by use of a preshared key.
- *Push-PMK-R1* – This variable is set to TRUE when R0KH can push the PMK-R1 security associations to R1KHs.

12.9.2.4 R0KH state machine procedures

The following list summarizes the procedures used by the R0KH state machine:

- **Authorized(R1KH-ID)** – This procedure returns a value of true if the R1KH is a known key holder in the mobility domain.
- **Distribute (List-R1KH-IDs, PMK-R1 PMKSA)** – Distributes the PMK-R1-SAs for the current instance of the key hierarchy to the list of R1KH-IDs.
- **Derive-Key-PMK-R0()** – This procedure derives the PMK-R0 from the MSK or PSK, derives the PMKR0Name (as described in 11.6.1.7.3), and creates PMK-R0 security association.
- **Derive-Key-PMK-R1 (R1KH-ID)** – This procedure derives the PMK-R1 from PMK-R0, and R1KH-ID (as described in 11.6.1.7.4), for the R1KH identified by R1KH-ID, and creates PMK-R1 security association.

12.9.3 R1KH state machine

12.9.3.1 General

The R1KH state machine includes functions for FT initial association and FT protocols. The R1KH states performing FT initial association and the R1KH states performing FT protocol exchanges interact differently with the R0KH state machine.

The R1KH state machine and other portions of the SME are defined in Figure 12-14 and Figure 12-15 and consist of a set of states that handle FT initial mobility domain association, PMK-R1 reception, PTK handshake and session establishment, FT protocols (including resource requests), and cleanup. This state machine interacts with the R0KH state machine to generate a fresh FT key hierarchy for the initial mobility domain association and to get the PMK-R1 security association (PMK-R1 PMKSA) for the FT protocols. While the figures show the over-the-air message exchanges, the over-the-DS exchanges are handled similarly.

A new instance of the R1KH state machine is created each time initial mobility domain association or fast BSS transition is initiated.

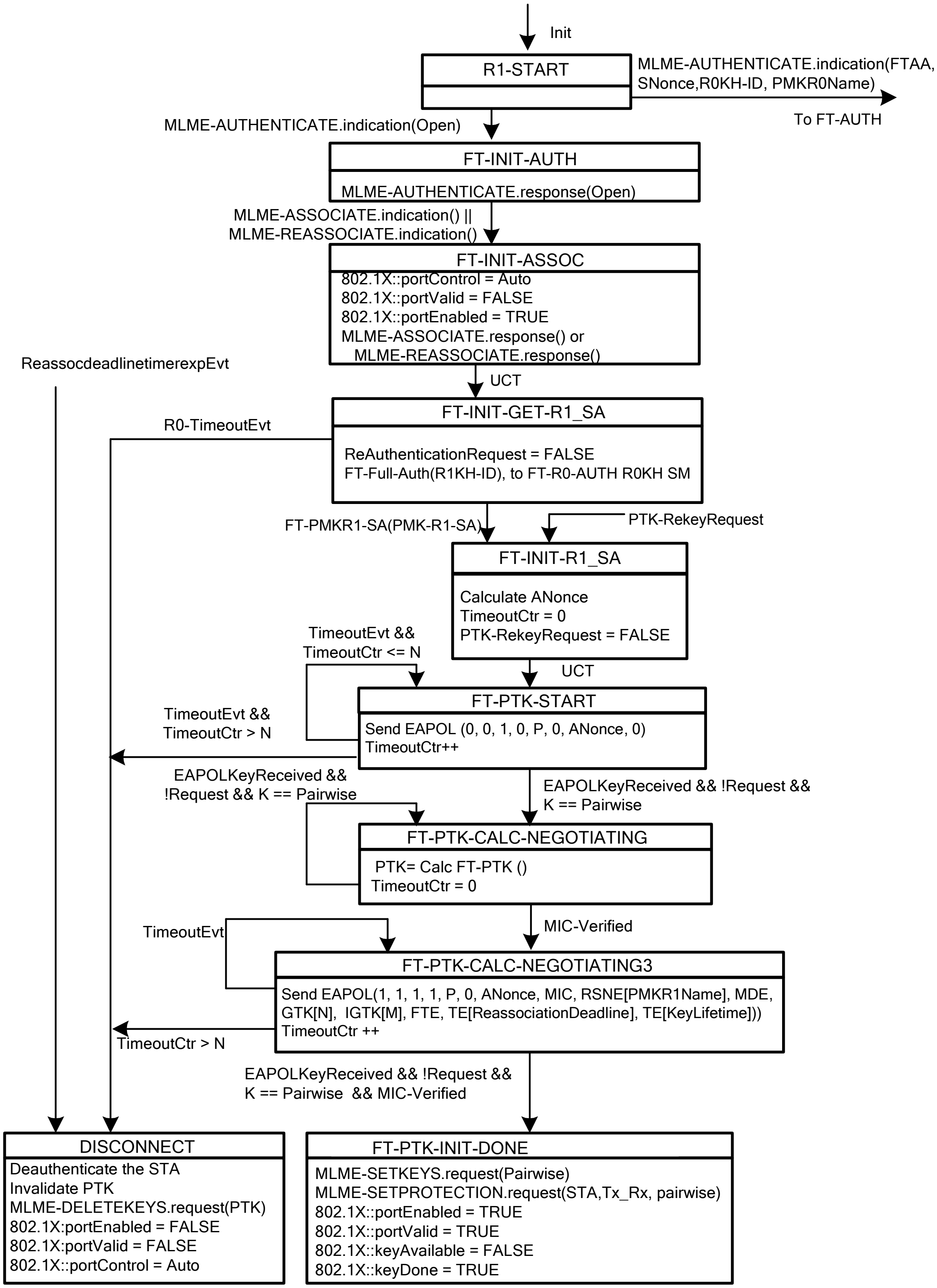

Init
R1-START
MLME-AUTHENTICATE.indication(FTAA, SNonce,R0KH-ID, PMKR0Name)
To FT-AUTH
MLME-AUTHENTICATE.indication(Open)
FT-INIT-AUTH
MLME-AUTHENTICATE.response(Open)
MLME-ASSOCIATE.indication() || MLME-REASSOCIATE.indication()
FT-INIT-ASSOC
802.1X::portControl = Auto
802.1X::portValid = FALSE
802.1X::portEnabled = TRUE
MLME-ASSOCIATE.response() or MLME-REASSOCIATE.response()
ReassocdeadlinetimerexpEvt
UCT
FT-INIT-GET-R1_SA
R0-TimeoutEvt
ReAuthenticationRequest = FALSE
FT-Full-Auth(R1KH-ID), to FT-R0-AUTH R0KH SM
FT-PMKR1-SA(PMK-R1-SA)
PTK-RekeyRequest
FT-INIT-R1_SA
Calculate ANonce
TimeoutCtr = 0
PTK-RekeyRequest = FALSE
TimeoutEvt && TimeoutCtr <= N
UCT
FT-PTK-START
TimeoutEvt && TimeoutCtr > N
Send EAPOL (0, 0, 1, 0, P, 0, ANonce, 0)
TimeoutCtr++
EAPOLKeyReceived && !Request && K == Pairwise
EAPOLKeyReceived && !Request && K == Pairwise
FT-PTK-CALC-NEGOTIATING
PTK= Calc FT-PTK ()
TimeoutCtr = 0
TimeoutEvt
MIC-Verified
FT-PTK-CALC-NEGOTIATING3
Send EAPOL(1, 1, 1, 1, P, 0, ANonce, MIC, RSNE[PMKR1Name], MDE, GTK[N], IGTK[M], FTE, TE[ReassociationDeadline], TE[KeyLifetime])
TimeoutCtr ++
TimeoutCtr > N
EAPOLKeyReceived && !Request && K == Pairwise && MIC-Verified
DISCONNECT
Deauthenticate the STA
Invalidate PTK
MLME-DELETEKEYS.request(PTK)
802.1X:portEnabled = FALSE
802.1X:portValid = FALSE
802.1X::portControl = Auto
FT-PTK-INIT-DONE
MLME-SETKEYS.request(Pairwise)
MLME-SETPROTECTION.request(STA,Tx_Rx, pairwise)
802.1X::portEnabled = TRUE
802.1X::portValid = TRUE
802.1X::keyAvailable = FALSE
802.1X::keyDone = TRUE

Figure 12-14—R1KH state machine, including portions of the SME (part 1)

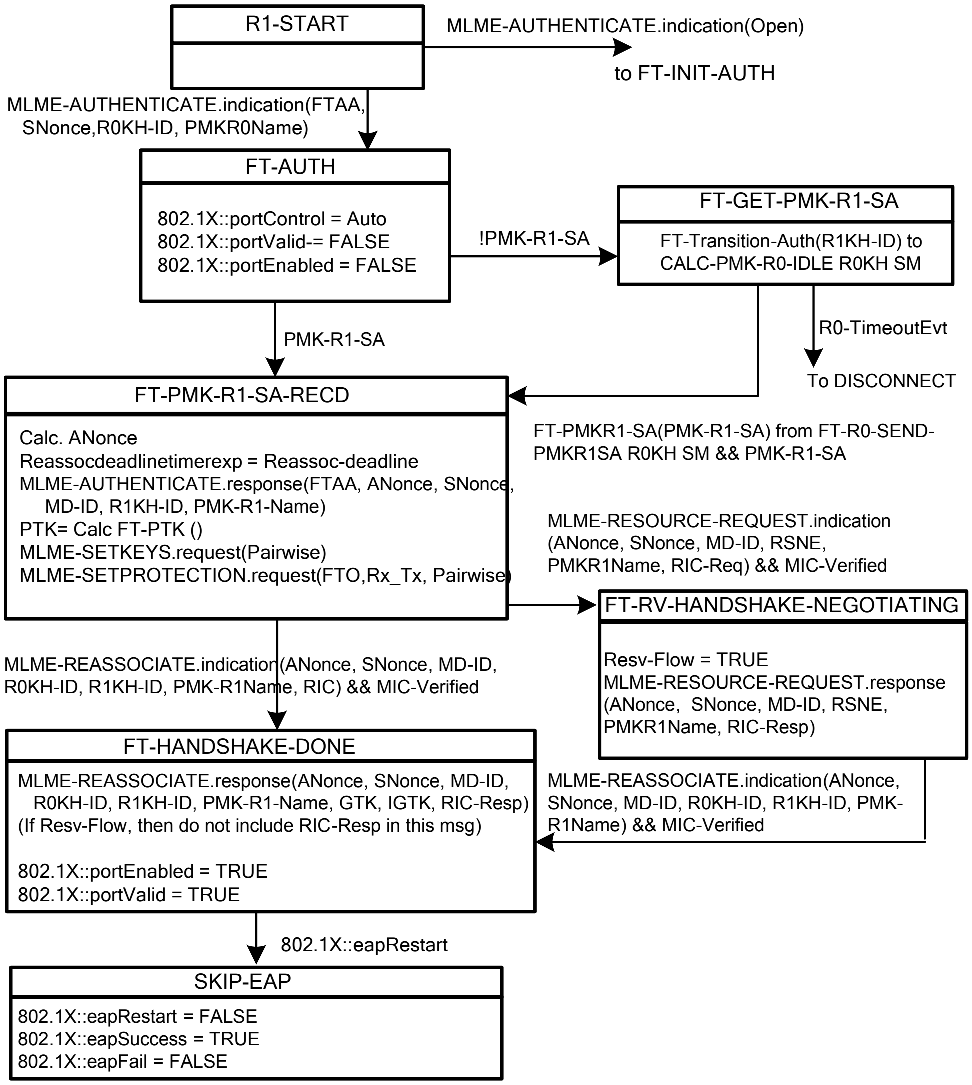

R1-START
MLME-AUTHENTICATE.indication(Open)
to FT-INIT-AUTH
MLME-AUTHENTICATE.indication(FTAA, SNonce,R0KH-ID, PMKR0Name)
FT-AUTH
802.1X::portControl = Auto
802.1X::portValid-= FALSE
802.1X::portEnabled = FALSE
!PMK-R1-SA
FT-GET-PMK-R1-SA
FT-Transition-Auth(R1KH-ID) to CALC-PMK-R0-IDLE R0KH SM
R0-TimeoutEvt
To DISCONNECT
PMK-R1-SA
FT-PMK-R1-SA-RECD
Calc. ANonce
Reassocdeadlinetimerexp = Reassoc-deadline
MLME-AUTHENTICATE.response(FTAA, ANonce, SNonce, MD-ID, R1KH-ID, PMK-R1-Name)
PTK= Calc FT-PTK ()
MLME-SETKEYS.request(Pairwise)
MLME-SETPROTECTION.request(FTO,Rx_Tx, Pairwise)
FT-PMKR1-SA(PMK-R1-SA) from FT-R0-SEND-PMKR1SA R0KH SM && PMK-R1-SA
MLME-RESOURCE-REQUEST.indication (ANonce, SNonce, MD-ID, RSNE, PMKR1Name, RIC-Req) && MIC-Verified
FT-RV-HANDSHAKE-NEGOTIATING
Resv-Flow = TRUE
MLME-RESOURCE-REQUEST.response (ANonce, SNonce, MD-ID, RSNE, PMKR1Name, RIC-Resp)
MLME-REASSOCIATE.indication(ANonce, SNonce, MD-ID, R0KH-ID, R1KH-ID, PMK-R1Name, RIC) && MIC-Verified
FT-HANDSHAKE-DONE
MLME-REASSOCIATE.response(ANonce, SNonce, MD-ID, R0KH-ID, R1KH-ID, PMK-R1-Name, GTK, IGTK, RIC-Resp)
(If Resv-Flow, then do not include RIC-Resp in this msg)
802.1X::portEnabled = TRUE
802.1X::portValid = TRUE
MLME-REASSOCIATE.indication(ANonce, SNonce, MD-ID, R0KH-ID, R1KH-ID, PMK-R1Name) && MIC-Verified
802.1X::eapRestart
SKIP-EAP
802.1X::eapRestart = FALSE
802.1X::eapSuccess = TRUE
802.1X::eapFail = FALSE

Figure 12-15—R1KH state machine, including portions of the SME (part 2)

12.9.3.2 R1KH state machine states

The following list summarizes the states of the R1KH state machine:

- **DISCONNECT**: This state is entered when the current session ends or when errors occur.
- **FT-AUTH**: This state is entered upon receipt of an indication that an FT Protocol or FT Resource Request Protocol is invoked.
- **FT-GET-PMK-R1-SA**: This state is entered when R1KH sends a message to the R0KH to get the PMK-R1-SA.
- **FT-HANDSHAKE-DONE**: This state is entered when reassociation indication parameters are validated. The Reassociation Response frame is then sent.
- **FT-INIT-ASSOC**: This state is entered upon receipt of a (Re)Association Request frame during initial mobility domain association.

- **FT-INIT-AUTH**: This state is entered upon receipt of an indication that initial association is invoked.
- **FT-INIT-GET-R1_SA**: This state is entered when the R1KH determines that a new key hierarchy is required.
- **FT-INIT-R1_SA**: This state is entered on receiving the PMK-R1-SA from the R0KH and when rekeying the PTK.
- **FT-PMK-R1-SA-RECD**: This state is entered on receiving the PMK-R1-SA from the R0KH. An FT Authenticate response is sent in this state. This state then calculates the PTK and delivers the key to the MAC.
- **FT-PTK-INIT-DONE**: This state is entered on successful validation of the fourth EAPOL-Key message. In this state, keys are provided to the MAC.
- **FT-PTK-CALC-NEGOTIATING**: This state is entered when a second EAPOL-Key message is received.
- **FT-PTK-CALC-NEGOTIATING3**: This state is entered on successful validation of the second EAPOL-Key message. In this state, the third EAPOL-Key message is sent.
- **FT-PTK-START**: This state is entered when the PMK-R1-SA is present. This state is the beginning of the 4-Way Handshake to derive a fresh PTK.
- **FT-RV-HANDSHAKE-NEGOTIATING**: This state is entered when an FT resource request is received. The FT resource response is sent.
- **R1-START**: This is the start of the R1KH state machine.
- **SKIP-EAP**: This state is entered after successful completion of the FT Protocol. In this state, the EAPOL state machine is triggered to open the IEEE 802.1X port.

12.9.3.3 R1KH state machine variables

The following list summarizes the variables used by the R1KH state machine:

- *Init* – This variable is set to TRUE to initialize the R1KH the state machine.
- *EAPOLKeyReceived* – This variable is set to TRUE when an EAPOL-Key message is received.
- *K* – This variable is one of the values of the Key Type bit in the EAPOL-Key frame received and is either Pairwise or Group.
- *MIC-Verified* – This variable is set to TRUE when the message authentication code integrity check passes.
- *Pairwise* – This variable is one of the values of the Key Type bit in the EAPOL-Key frame.
- *PMK-R1-SA* – This variable is set to TRUE when a valid PMK-R1-SA is present at the R1KH.
- *PTK-RekeyRequest* – This variable is set to TRUE when a PTK Key request is received.
- *Reassocdeadlinetimerexp* – This variable contains the reassociation deadline timer value.
- *ReassocdeadlinetimerexpEvt* – This variable is set to TRUE when the reassociation deadline timer expires.
- *Request* – This variable is the value of the Request bit in the Key Information field in the EAPOL-Key frame.
- *Resv-flow* – This variable is set to TRUE when an indication of an FT Resource Request Protocol is received.
- *R0-TimeoutEvt* – This variable is set to TRUE when the timeout for R0KH authentication expires (e.g., when the EAP authentication session timeout expires).
- *TimeoutCtr* – This variable contains the number of successive timeouts waiting for protocol responses.

- *TimeoutEvt* – This variable is set to TRUE when a timeout for receiving EAPOL-Key response expires.

12.9.3.4 R1KH state machine procedures

The following list summarizes the procedure used by the R1KH state machine:

- **Calc-FT-PTK()** – This procedure calculates the PTK.

12.9.4 S0KH state machine

12.9.4.1 General

There is one S0KH state machine within the Supplicant, defined in Figure 12-16, which incorporates the FT initial association and key management.

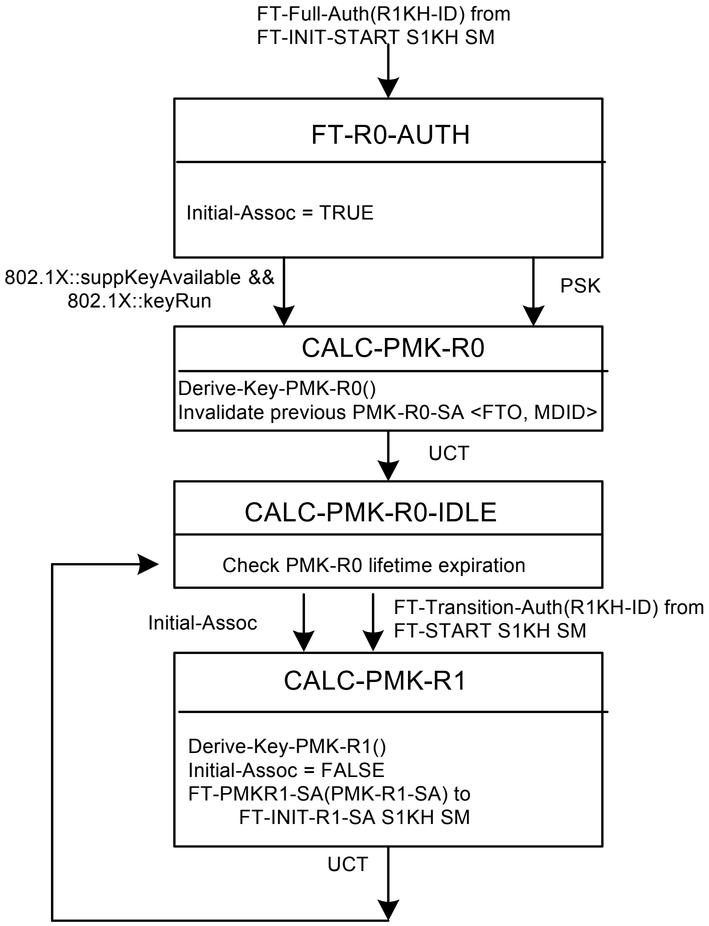


Figure 12-16—S0KH state machine

12.9.4.2 S0KH state machine states

The following list summarizes the states of the S0KH state machine:

- **CALC-PMK-R0**: This state is entered after the key is received, either from the EAP authentication or from the PSK.
- **CALC-PMK-R0-IDLE**: This state is entered after the PMK-R0 has been calculated and either continues with initial association or waits for requests from an S1KH for a PMK-R1.
- **CALC-PMK-R1**: For FT initial association, this state is entered as an unconditional transfer. For FT protocols, this state is entered through the event from the S1KH state machine. In this state, the PMK-R1 is sent to the S1KH.
- **FT-R0-AUTH**: This state is entered when the FT-Full-Auth event occurs during initial association in the S1KH state machine. The S1KH state machine sends this event when it determines that a new PMK-R0 is needed.

12.9.4.3 S0KH state machine variables

The following list summarizes the variables used by the S0KH state machine:

- *Initial-Assoc* – This variable is used to indicate whether the current authentication is the initial association, in order to trigger the initial derivation of PMK-R1.
- *PSK* – This variable is set to TRUE when authentication is performed by use of a preshared key.

12.9.4.4 S0KH state machine procedures

The following list summarizes the procedures used by the S0KH state machine:

- **Derive-Key-PMK-R0()** – This procedure derives the PMK-R0 and PMKR0Name and creates PMK-R0 security association.
- **Derive-Key-PMK-R1 (R1KH-ID)** – This procedure derives the PMK-R1 and PMKR1Name from PMK-R0 for the indicated R1KH and creates PMK-R1 security association.

12.9.5 S1KH state machine

12.9.5.1 General

The S1KH state machine includes functions for fast BSS transitions, including initial association. The S1KH state machine and other portions of the SME are defined in Figure 12-17 and Figure 12-18 and consist of a set of states that handle FT initial association, PTK handshake and session establishment, resource requests, cleanup, and teardown. This state machine interacts with the S0KH state machine to generate a fresh key hierarchy.

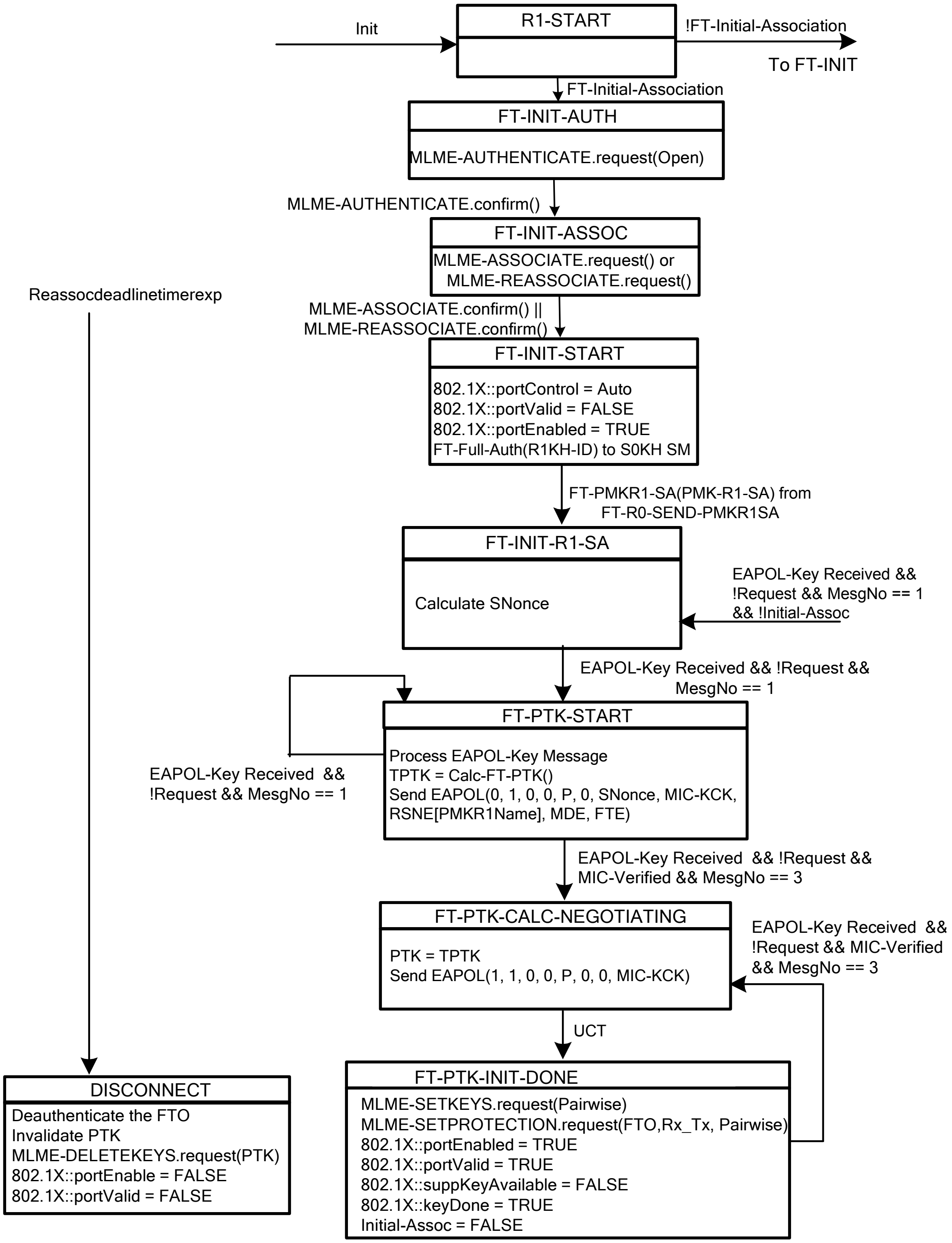

Init
R1-START
!FT-Initial-Association
To FT-INIT
FT-Initial-Association
FT-INIT-AUTH
MLME-AUTHENTICATE.request(Open)
MLME-AUTHENTICATE.confirm()
FT-INIT-ASSOC
MLME-ASSOCIATE.request() or
MLME-REASSOCIATE.request()
Reassocdeadlinetimerexp
MLME-ASSOCIATE.confirm() ||
MLME-REASSOCIATE.confirm()
FT-INIT-START
802.1X::portControl = Auto
802.1X::portValid = FALSE
802.1X::portEnabled = TRUE
FT-Full-Auth(R1KH-ID) to S0KH SM
FT-PMKR1-SA(PMK-R1-SA) from
FT-R0-SEND-PMKR1SA
FT-INIT-R1-SA
Calculate SNonce
EAPOL-Key Received &&
!Request && MesgNo == 1
&& !Initial-Assoc
EAPOL-Key Received && !Request &&
MesgNo == 1
FT-PTK-START
Process EAPOL-Key Message
TPTK = Calc-FT-PTK()
Send EAPOL(0, 1, 0, 0, P, 0, SNonce, MIC-KCK,
RSNE[PMKR1Name], MDE, FTE)
EAPOL-Key Received &&
!Request && MesgNo == 1
EAPOL-Key Received && !Request &&
MIC-Verified && MesgNo == 3
FT-PTK-CALC-NEGOTIATING
PTK = TPTK
Send EAPOL(1, 1, 0, 0, P, 0, 0, MIC-KCK)
EAPOL-Key Received &&
!Request && MIC-Verified
&& MesgNo == 3
UCT
DISCONNECT
Deauthenticate the FTO
Invalidate PTK
MLME-DELETEKEYS.request(PTK)
802.1X::portEnable = FALSE
802.1X::portValid = FALSE
FT-PTK-INIT-DONE
MLME-SETKEYS.request(Pairwise)
MLME-SETPROTECTION.request(FTO,Rx_Tx, Pairwise)
802.1X::portEnabled = TRUE
802.1X::portValid = TRUE
802.1X::suppKeyAvailable = FALSE
802.1X::keyDone = TRUE
Initial-Assoc = FALSE

Figure 12-17—S1KH state machine, including portions of the SME (part 1)

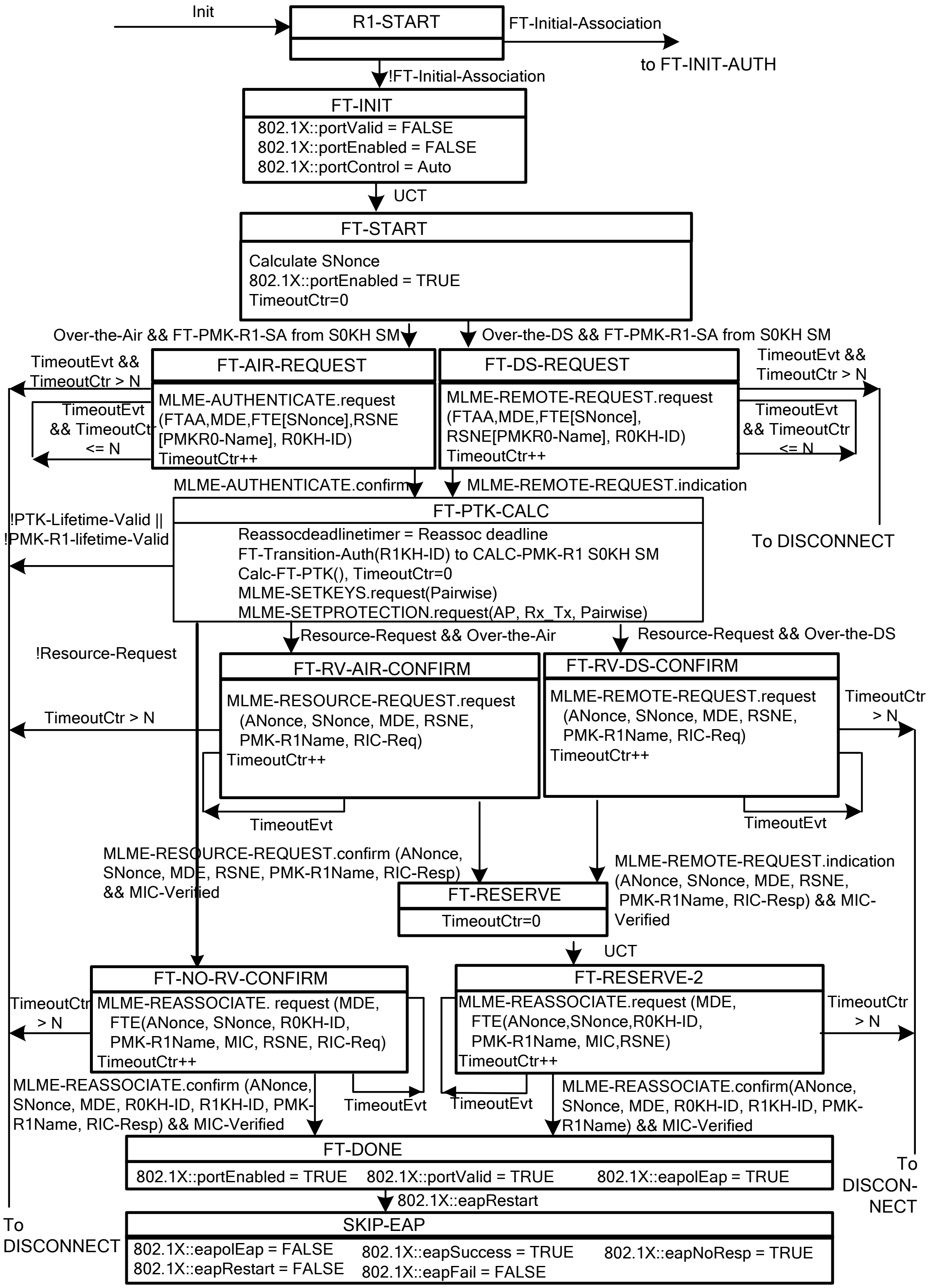

Init
R1-START
FT-Initial-Association
to FT-INIT-AUTH
!FT-Initial-Association
FT-INIT
802.1X::portValid = FALSE
802.1X::portEnabled = FALSE
802.1X::portControl = Auto
UCT
FT-START
Calculate SNonce
802.1X::portEnabled = TRUE
TimeoutCtr=0
Over-the-Air && FT-PMK-R1-SA from S0KH SM
Over-the-DS && FT-PMK-R1-SA from S0KH SM
TimeoutEvt &&
TimeoutCtr > N
FT-AIR-REQUEST
MLME-AUTHENTICATE.request
(FTAA,MDE,FTE[SNonce],RSNE
[PMKR0-Name], R0KH-ID)
TimeoutCtr++
TimeoutEvt
&& TimeoutCtr
<= N
FT-DS-REQUEST
MLME-REMOTE-REQUEST.request
(FTAA,MDE,FTE[SNonce],
RSNE[PMKR0-Name], R0KH-ID)
TimeoutCtr++
TimeoutEvt &&
TimeoutCtr > N
TimeoutEvt
&& TimeoutCtr
<= N
MLME-AUTHENTICATE.confirm
MLME-REMOTE-REQUEST.indication
To DISCONNECT
!PTK-Lifetime-Valid ||
!PMK-R1-lifetime-Valid
FT-PTK-CALC
Reassocdeadlinetimer = Reassoc deadline
FT-Transition-Auth(R1KH-ID) to CALC-PMK-R1 S0KH SM
Calc-FT-PTK(), TimeoutCtr=0
MLME-SETKEYS.request(Pairwise)
MLME-SETPROTECTION.request(AP, Rx_Tx, Pairwise)
!Resource-Request
Resource-Request && Over-the-Air
Resource-Request && Over-the-DS
FT-RV-AIR-CONFIRM
MLME-RESOURCE-REQUEST.request
(ANonce, SNonce, MDE, RSNE,
PMK-R1Name, RIC-Req)
TimeoutCtr++
TimeoutCtr > N
TimeoutEvt
FT-RV-DS-CONFIRM
MLME-REMOTE-REQUEST.request
(ANonce, SNonce, MDE, RSNE,
PMK-R1Name, RIC-Req)
TimeoutCtr++
TimeoutCtr
> N
TimeoutEvt
MLME-RESOURCE-REQUEST.confirm (ANonce,
SNonce, MDE, RSNE, PMK-R1Name, RIC-Resp)
&& MIC-Verified
MLME-REMOTE-REQUEST.indication
(ANonce, SNonce, MDE, RSNE,
PMK-R1Name, RIC-Resp) && MIC-
Verified
FT-RESERVE
TimeoutCtr=0
UCT
FT-NO-RV-CONFIRM
MLME-REASSOCIATE. request (MDE,
FTE(ANonce, SNonce, R0KH-ID,
PMK-R1Name, MIC, RSNE, RIC-Req)
TimeoutCtr++
TimeoutCtr
> N
FT-RESERVE-2
MLME-REASSOCIATE.request (MDE,
FTE(ANonce,SNonce,R0KH-ID,
PMK-R1Name, MIC,RSNE)
TimeoutCtr++
TimeoutCtr
> N
MLME-REASSOCIATE.confirm (ANonce,
SNonce, MDE, R0KH-ID, R1KH-ID, PMK-
R1Name, RIC-Resp) && MIC-Verified
TimeoutEvt
TimeoutEvt
MLME-REASSOCIATE.confirm(ANonce,
SNonce, MDE, R0KH-ID, R1KH-ID, PMK-
R1Name) && MIC-Verified
FT-DONE
802.1X::portEnabled = TRUE
802.1X::portValid = TRUE
802.1X::eapolEap = TRUE
To
DISCON-
NECT
802.1X::eapRestart
To
DISCONNECT
SKIP-EAP
802.1X::eapolEap = FALSE
802.1X::eapRestart = FALSE
802.1X::eapSuccess = TRUE
802.1X::eapFail = FALSE
802.1X::eapNoResp = TRUE

Figure 12-18—S1KH state machine, including portions of the SME (part 2)

12.9.5.2 S1KH state machine states

The following list summarizes the states of the S1KH state machine:

- **DISCONNECT**: This state is entered when the current session expires.
- **FT-AIR-REQUEST**: This state is entered when it is determined that an over-the-air FT method is to be executed. This state sends the FT Authentication Request frame over the air.
- **FT-DONE**: This state is entered when a Reassociation Response frame is received.
- **FT-DS-REQUEST**: This state is entered when it is determined that an over-the-DS FT method is to be executed. This state sends the FT Authentication Request frame over the DS.
- **FT-INIT**: This state is entered when an FT method is initiated.
- **FT-INIT-ASSOC:** This state is entered when authentication for initial mobility domain association has been completed.
- **FT-INIT-AUTH**: This state is entered when an FT initial association event is initiated.
- **FT-INIT-START:** This state is entered when association for initial mobility domain association has been completed.
- **FT-INIT-R1-SA**: This state is entered on receiving the PMK-R1-SA from the S0KH and when the Authenticator is starting PTK rekeying by sending out EAPOL-Key Message 1.
- **FT-NO-RV-CONFIRM**: This state is entered when performing FT Protocol (i.e., not the FT Resource Request Protocol). This state is entered for both over-the-air and over-the-DS processing. This state sends the Reassociation Request frame.
- **FT-PTK-CALC**: This state is entered when the over-the-air FT Authentication Response frame is received if an over-the-air FT Authentication Request frame was sent or when over-the-DS FT Response frame is received if an over-the-DS FT Request frame was sent. The PTK is calculated and installed in the MAC.
- **FT-PTK-INIT-DONE**: This state is entered after sending the fourth EAPOL-Key message. This state establishes the PTK keys into the MAC.
- **FT-PTK-CALC-NEGOTIATING**: This state is entered when a valid, third EAPOL-Key message is received. This state sends the fourth EAPOL-Key message.
- **FT-PTK-START**: This state is entered to derive a new PTK when the PMK-R1-SA is present and when the EAPOL-Key 4-Way Handshake Message 1 is received. This state sends the EAPOL-Key 4-Way Handshake Message 2.
- **FT-RESERVE**: This state is entered when the over-the-air FT Authentication Ack frame is received if an over-the-air FT Authentication Confirm frame was sent or when over-the-DS FT Ack frame is received if an over-the-DS FT Confirm frame was sent.
- **FT-RESERVE-2**: The Reassociation Request frame is sent in this state after completion of FT resource request.
- **FT-RV-AIR-CONFIRM**: This state is entered for over-the-air FT Resource Request Protocol processing. The FT Authentication Confirm frame containing the FT resource request is sent.
- **FT-RV-DS-CONFIRM**: This state is entered for over-the-DS FT Resource Request Protocol processing. The FT Authentication Confirm frame containing the FT resource request is sent.
- **FT-START**: This state is entered when all FT parameters are validated and FT needs to be initiated.
- **R1-START**: This is the start of the S1KH state machine.
- **SKIP-EAP**: This state is entered after successful completion of the FT Protocol. In this state, the EAPOL state machine is triggered to open the IEEE 802.1X port.

12.9.5.3 S1KH state machine variables

The following list summarizes the variables used by the S1KH state machine:

- *EAPOLKeyReceived* – This variable is set to TRUE when an EAPOL-Key message is received.
- *FT-Initial-Association* – This variable is set to TRUE when the S1KH is performing an initial association.
- *Init* – This variable is set to TRUE to initialize the S1KH state machine. In addition, this variable is used to restart the state machine when transitioning to a new AP.
- *MesgNo* – In conjunction with *EAPOLKeyReceived*, this variable indicates which message in the 4-Way Handshake has been received.
- *MIC-Verified* – This variable is set to TRUE when the message authentication integrity check is valid.
- *N* – This variable contains the limit of timeout events before considering the transition a failure.
- *Over-the-Air* – This variable is set to TRUE when the FT Protocol is to be exchanged over the air. Note that both *Over-the-Air* and *Over-the-DS* cannot be equal to TRUE at the same time.
- *Over-the-DS* – This variable is set to TRUE when the FT Protocol is to be exchanged over the DS. Note that both *Over-the-Air* or *Over-the-DS* cannot be equal to TRUE at the same time.
- *PMK-R1-Lifetime-Valid* – This variable is set to TRUE when the PMK-R1 lifetime is valid.
- *PTK-Lifetime-Valid* – This variable is set to TRUE when the PTK lifetime is valid.
- *Reassocdeadlinetimer* – This variable contains the reassociation deadline timer value.
- *ReassocdeadlinetimerExp* – This variable is set to TRUE when the reassociation deadline timer expires.
- *Resource-request* – This variable is set to TRUE when the FT Resource Request Protocol is to be executed.
- *TimeoutCtr* – This variable contains the number of successive timeouts waiting for protocol responses.
- *TimeoutEvt* – This variable is set to TRUE when a timeout event occurs.
- *TPTK* – This variable contains the newly calculated PTK, which is installed after receipt of Message 3 of the 4-Way Handshake.

12.9.5.4 S1KH state machine procedures

The following list summarizes the procedure used by the S1KH state machine:

- **Calc-FT-PTK()** – This procedure calculates the PTK.

12.10 Remote request broker (RRB) communication

12.10.1 Overview

The RRB mechanism allows the FTO to communicate with a target AP through the FTO's existing association (with the current AP). The FTO transmits an FT Action frame (including the address of the FTO and the BSSID of the target AP) to the current AP. The current AP includes the contents of the FT Action frame (Request or Confirm) inside a Remote Request frame and transmits it to the target AP over the DS. The target AP processes the remote request and responds to the FTO by sending an FT Action frame (Response or Acknowledgment) through the current AP.

The SME of the FTO initiates an exchange with a target AP by issuing an MLME-REMOTE-REQUEST.request primitive with parameters including the contents of the FT Action frame to be sent. The

MAC of the FTO transmits this Action frame. When the MAC of the current AP receives an FT Action frame, it passes it to the RRB by use of an MLME-REMOTE-REQUEST.indication primitive, with parameters including the contents of the received Action frame.

When the RRB of the current AP has received a response from the target AP, it uses the MLME-REMOTE-REQUEST.request primitive to send the response, as an FT Action frame, to the requesting FTO. The MAC of the current AP transmits this Action frame. When the MAC of the FTO receives an FT Action frame, the MAC passes the Action frame to the SME by use of an MLME-REMOTE-REQUEST.indication primitive, with parameters including the contents of the received Action frame.

12.10.2 Remote request broker (RRB)

The RRB resides in the SME on the APs and acts as a forwarding agent (at the current AP) and termination point (at the target AP) for protocol messages over the DS.

The RRB allows APs that are part of the same mobility domain to exchange information over the DS. APs that advertise the same MDID shall be reachable over the DS and support the over-the-DS communication.

As a termination point, when the RRB at the target AP receives a request frame from the current AP, it interacts with the MAC and other parts of the SME to process the request and respond with a Remote Response frame, through the RRB on the current AP, back to the requesting FTO.

As a forwarding agent, when the RRB at the current AP receives a request from an FTO directed to another AP in the same mobility domain, the current AP forwards the request to that target AP. The RRB on the current AP converts Action frames into Remote Request frames and converts Remote Response frames into Action frames.

The target AP and the current AP need to reside in the same mobility domain to successfully exchange Remote Request frames. The RRB on the current AP shall transmit Remote Request frames to the target AP based on the BSSID of the target AP (supplied in the FT Action frames) using the same procedures as preauthentication, as described in 11.5.9.2.

The message flow for a resource request over the DS is given in Figure 12-19. The FTO indicates the destination target AP BSSID as part of the FT Action frame. The RRB on the current AP encapsulates the FT Action frame and supplies the current AP BSSID in the Remote Request frame.

12.10.3 Remote Request/Response frame definition

This subclause defines a mechanism to transport the remote request and remote response between the current AP and the target AP. Any other mechanism may be used.

The Remote Request frame is transmitted over the DS from the current AP to the target AP. The Payload for the Remote Request/Response frame is given in Table 12-2. Remote Request/Response frames shall use an Ethertype of 89-0d, as specified in Annex H. The Remote Request/Response frame contains version, type, and length fields, along with the AP Address.

The FT Packet Type field shall be set to 0 for remote request and to 1 for remote response.

The FT Action Length field shall be set to an unsigned number representing the length in octets of the FT Action Frame field, following the bit ordering conventions of 8.2.2.

The AP Address field shall be set to the BSSID of the current AP. The target AP shall use this address as the destination address when sending the Remote Response frame as a response to the Remote Request.

The FT Action Frame field shall be set to the contents of the FT Action frame, from the Category field to the end of the Action frame body.

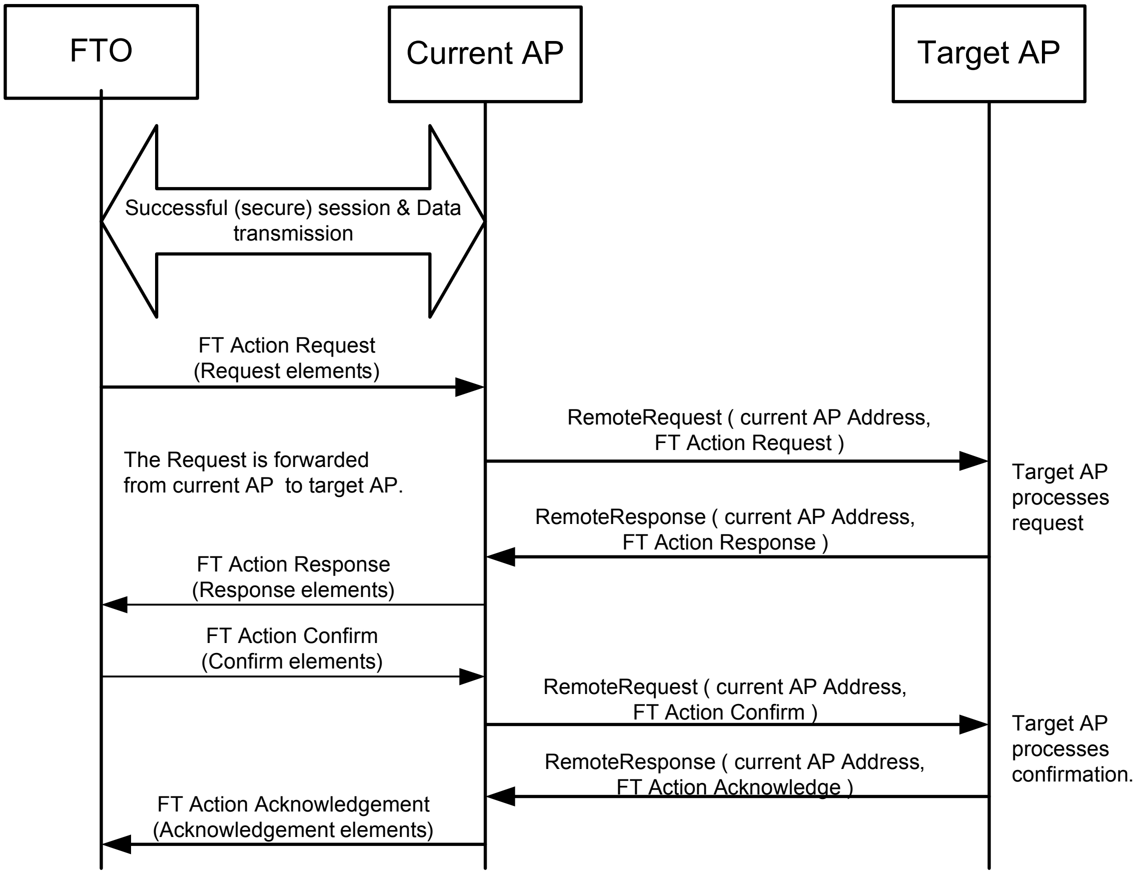

FTO
Current AP
Target AP
Successful (secure) session & Data transmission
FT Action Request (Request elements)
RemoteRequest (current AP Address, FT Action Request)
The Request is forwarded from current AP to target AP.
Target AP processes request
RemoteResponse (current AP Address, FT Action Response)
FT Action Response (Response elements)
FT Action Confirm (Confirm elements)
RemoteRequest (current AP Address, FT Action Confirm)
Target AP processes confirmation.
RemoteResponse (current AP Address, FT Action Acknowledge)
FT Action Acknowledgement (Acknowledgement elements)

Figure 12-19—Sample message flow for over-the-DS resource request

Table 12-2—Remote Request/Response Payload format

Size	Information
1	FT Packet Type
2	FT Action Length
6	AP Address
Variable	FT Action Frame

12.11 Resource request procedures

12.11.1 General

When using the resource request procedure, the FTO has the option to request a resource allocation at the target AP. To request resources, the FTO creates a resource information container (RIC) and inserts it in an appropriate request message to the target AP. The request message is sent to the target AP either directly (over the air), or via the current AP (over the DS), according to the FT procedures described in 12.5 and 12.6. In an RSNA, resource requests and responses are exchanged only after the establishment of the PTK and are protected by MICs.

The RIC contains a complete list of resources requested by the FTO. An AP that receives a resource request from an FTO shall discard any previous resource request from that FTO. In an RSN, this resource request shall first be authenticated by the AP through checking of the MIC before the AP discards any previous resource request.

If an FTO is performing a fast BSS transition according to the FT Protocol, described in 12.5, it shall generate a RIC and process the RIC-Response according to the procedures of 12.11.3.1, performing the exchange in the Reassociation Request/Response frames.

If an FTO is performing a fast BSS transition according to the FT Resource Request Protocol, described in 12.6, it shall generate a RIC and process the RIC-Response according to the procedures of 12.11.3.1, performing the exchange in the Authentication Confirm/Authentication Ack frames (over the air) or FT Confirm/FT Ack frames (over the DS).

12.11.2 Resource information container (RIC)

The RIC refers to a collection of elements that are used to express a resource request or response.

When used in making a request, a RIC has one or more Resource Requests, as shown in Figure 12-20.

Resource Request	Resource Request	Resource Request

Figure 12-20—RIC-Request format

Each Resource Request consists of an RDE followed by one or more alternative Resource Descriptors. An example of a Resource Request is shown in Figure 12-21.

RDE	Resource Descriptor

Figure 12-21—Resource Request format

Each Resource Descriptor consists of one or more elements. The possible Resource Descriptors that may appear in a RIC, and the elements that they contain, are given in Table 12-3.

Table 12-3—Resource types and resource descriptor definitions

Resource type	Resource Descriptor definition	Notes
802.11 QoS	In a request: TSPEC (see 8.4.2.32), followed by zero or more TCLAS (see 8.4.2.33), followed by zero or one TCLAS Processing (see 8.4.2.35), followed by zero or one Expedited Bandwidth Request elements (see 8.4.2.96). In a response: a TSPEC element (see 8.4.2.32), followed by zero or one Schedule elements (see 8.4.2.36), followed by zero or more Delay elements (see 8.4.2.34), followed by other optional elements as specified in 10.4.	May be sent by a QoS STA that is an FTO to a QoS AP. Definition of TSPEC elements shall be as given in 10.4. Definition of TCLAS, TCLAS Processing, Expedited Bandwidth Request, and Schedule elements, and the rules for including them in requests and responses, shall be as given in 10.4. Resource request procedures shall be as given in 10.4.

Table 12-3—Resource types and resource descriptor definitions *(continued)*

Resource type	Resource Descriptor definition	Notes
Block Ack Parameters	In a request: RIC Descriptor (see 8.4.2.53), containing a Resource Type field identifying Block Ack. In a response: RIC Descriptor (see 8.4.2.53), containing a Resource Type field identifying Block Ack.	Resource request procedures shall be as given in 10.5.
Vendor Specific	RDE is followed by any vendor-specific elements required to specify this resource.	

If there are multiple Resource Descriptors, then they are treated as choices by the target AP. The AP attempts to allocate whatever is specified in the first Resource Descriptor; if this fails, the AP attempts to allocate whatever is specified in the next Resource Descriptor instead, and so on until a successful allocation or the AP reaches the end of the Resource Descriptor list. Thus, an OR relationship exists between Resource Descriptors that follow an RDE, with the Resource Descriptors appearing in order of preference.

An example of a Resource Request consisting of two alternative Resource Descriptors is shown in Figure 12-22.

RDE	Resource Descriptor	Resource Descriptor

Figure 12-22—Resource Request example #1

For example, when the resource being requested is QoS for downstream traffic, a TSPEC element may be followed by one or more TCLAS elements and, when multiple TCLAS elements are present, a TCLAS Processing element and an Expedited Bandwidth Request (EBR) element. Such an example Resource Request with two alternative TSPECs, the second of which has an EBR, is shown in Figure 12-23.

RDE	TSPEC	TCLAS	TCLAS	TCLAS Processing	TSPEC	TCLAS	TCLAS	TCLAS Processing	EBR

Figure 12-23—Resource Request example #2

An example of a RIC with two resource requests, each with a single TSPEC, is given in Figure 12-24.

RDE	TSPEC	RDE	TSPEC

Figure 12-24—RIC-Request example #1

An example of a RIC with one resource request, with a choice of two TSPECs, is given in Figure 12-25. This indicates that the target AP can select one of the two TSPECs.

RDE	TSPEC	TSPEC

Figure 12-25—RIC-Request example #2

An example of a RIC with a RIC Descriptor is given in Figure 12-26. The target AP can acknowledge if the resource specified in the RIC Descriptor is available.

RDE	RIC Descriptor (BlockAck)

Figure 12-26—RIC-Request example #3

When sent by an AP in response to a RIC-Request, the RIC-Response consists of a list of one or more Resource Responses including one response for each of the Resource Requests that was contained in the RIC-Request. The basic format of a RIC-Response is shown in Figure 12-27.

Resource Response	Resource Response	Resource Response

Figure 12-27—RIC-Response format

Each Resource Response consists of an RDE with the RDE identifier matching the RDE identifier in the request, in the same order as the RDEs appeared in the request. The RDE is followed by zero or one Resource Descriptors. If the request was not successful (as indicated in the RDE status), then the AP may include a suggestion that could have been successful. If the resource request was successful, then the particular Resource Descriptor (of the alternatives given by the FTO) is included in the response, as modified by the AP during the processing of the resource request. For example, when the resource being requested is QoS for upstream traffic, the TSPEC element may be followed by a Schedule element.

An example of a RIC-Response with two QoS resource responses, each with a single TSPEC and Schedule element, is given in Figure 12-28.

RDE	TSPEC	Schedule	RDE	TSPEC	Schedule

Figure 12-28—Example QoS RIC-Response

12.11.3 Creation and handling of a resource request

12.11.3.1 FTO procedures

The resource request enables an FTO to request resources based on specified Resource Descriptors (e.g., TSPECs) before or at the time the FTO associates with the target AP. In using TSPECs for requesting QoS resources, the TSPECs in the request need not belong to only active TSs; the FTO can send TSPECs for any TS that it intends to use after the transition and request the same resources that would be requested by a later ADDTS exchange. For each resource, the FTO may provide the AP with a choice of Resource Descriptors in order of preference, any one of which meets the needs of the application.

The FTO shall construct the RIC with a number of Resource Requests, each delineated by an RDE.

The FTO shall indicate the resources required at the target AP. For QoS resources, each TS shall be requested by a separate RDE and associated TSPEC(s). The RDE Identifier field in the RDE shall be an arbitrary value chosen by the FTO that uniquely identifies the RDE within the RIC. The Status Code field shall be set to 0, and the Resource Count field shall be set to the number of alternative Resource Descriptors that follow.

Following each RDE, the FTO shall include one or more Resource Descriptors that define the resources required for this TS. When multiple TSPECs follow an RDE as part of a single QoS resource request, a logical "OR" relationship exists between them, and at most one of these TSPECs shall be accepted by the AP. The FTO shall order the Resource Descriptors in decreasing order of preference.

In generating the RDE for QoS resources for a TS, the procedures of 10.4 shall be followed for the generation of TSPECs and inclusion of TCLAS, TCLAS Processing, and Expedited Bandwidth Request elements. If the TS is a downstream flow, then the RDE may also include one or more TCLAS element(s) (defined in 8.4.2.33) and a TCLAS Processing element (defined in 8.4.2.35) if multiple TCLAS elements are included, and an optional Expedited Bandwidth Request (EBR) element, defined in 8.4.2.96. If present, the TCLAS shall appear after the corresponding TSPEC. If present, an EBR element shall appear after the corresponding TSPEC, TCLAS, and TCLAS Processing elements of the TSPEC.

A resource request is considered successful if the status code 0 is returned in each RDE.

If the frame containing the response to the resource request contains a status code other than 0, the FTO considers that the request has failed and that no resources are being held at the target AP.

The response from the target AP contains a RIC-Response, with the RDEs in the response indicating which resources were considered by the target AP and the setting of the status code indicating which Resource Descriptors were accepted by the AP.

The RDE Identifier field in the RDE enables the FTO to match the response with the RDE in the request. The value of the Status Code field is interpreted as follows:

- Status code = 0 indicates that the request has been accepted. The RDE may be followed by the Resource Descriptor that was accepted.
- Status code = nonzero (one of the values from 8.4.1.9) indicates that the resources could not be accepted. The RDE may be followed by a suggested Resource Descriptor that could have been accepted.

A response to a successful resource request (other than in a Reassociation Request frame) may contain a reassociation deadline. If the FTO does not initiate a Reassociation Request frame with the target AP within the reassociation deadline (if appropriate), then the AP releases resources held for that FTO.

12.11.3.2 AP procedures

When a RIC appears in a request message, the AP shall check its ability to allocate one resource for each RDE in the RIC in the order appearing in the RIC. In a Reassociation Request frame, the QoS Capability element shall be processed prior to the QoS resource requests in the RIC.

The behavior of the AP shall be identical to that described in the flowchart in Figure 12-29.

As shown in Figure 12-29, the Resource Descriptors are examined by the AP in the order presented, and the first that could have been allocated is accepted. Thus the preference ordering by the FTO is honored.

The target AP's SME examines the resource requests in the RIC. For requests that require processing by the MAC sublayer, the SME generates an MLME-RESOURCE-REQUEST-LOCAL.request primitive. The MAC shall respond with MLME-RESOURCE-REQUEST-LOCAL.confirm primitive that indicates whether the MAC has accepted the resource request. The SME may also send these resource requests to an external entity such as a backend QoS module for its consideration; these procedures are beyond the scope of this standard. The acceptance of a TSPEC by the target AP results in the resource allocation for a TS at the target AP.

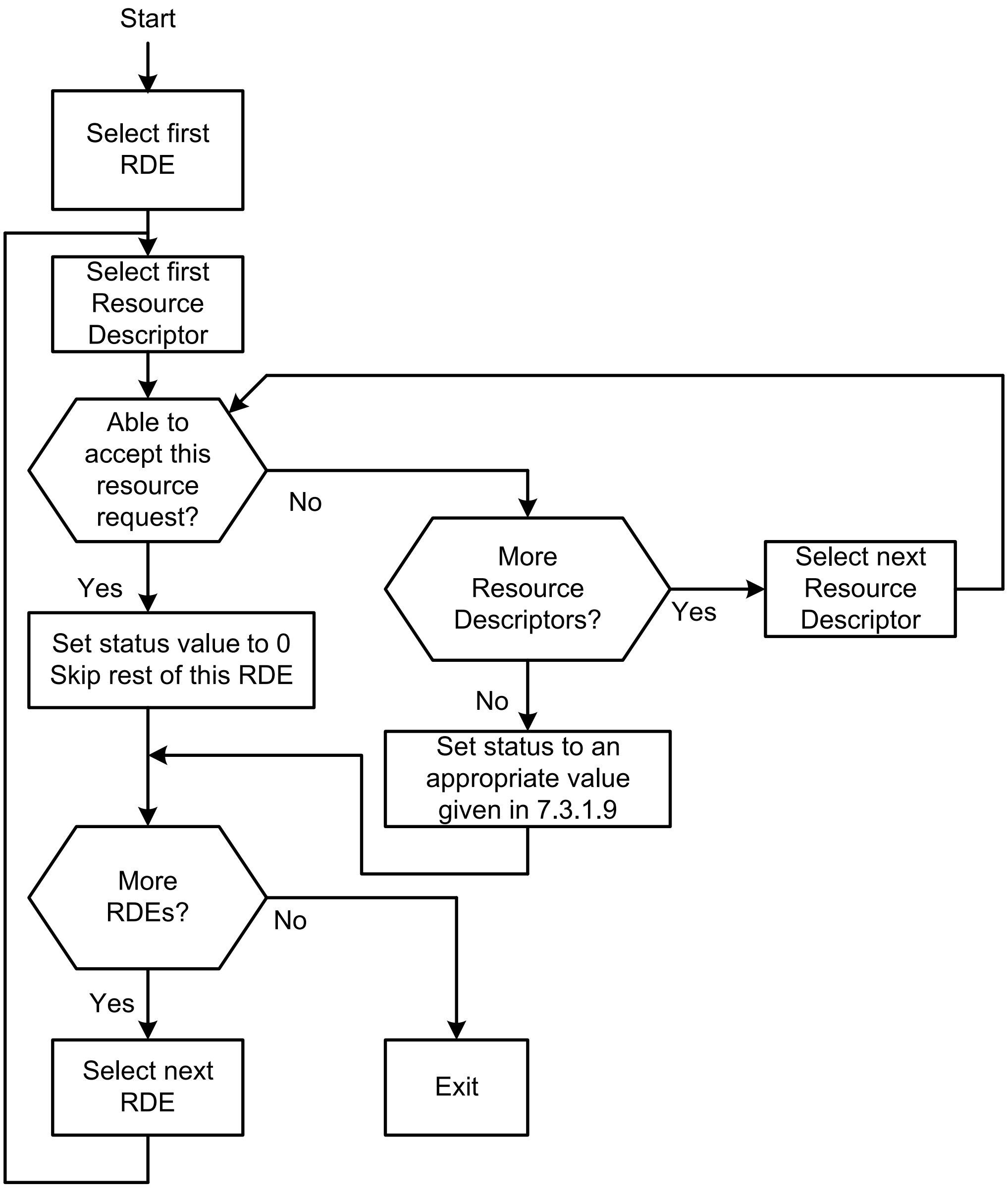

Start
Select first RDE
Select first Resource Descriptor
Able to accept this resource request?
No
Yes
More Resource Descriptors?
Yes
Select next Resource Descriptor
Set status value to 0 Skip rest of this RDE
No
Set status to an appropriate value given in 7.3.1.9
More RDEs?
No
Yes
Select next RDE
Exit

Figure 12-29—Overview of RIC processing at an AP

In response to a RIC-Request, the AP shall construct a RIC-Response. The RIC-Response shall contain one RDE for each RDE in the RIC-Request. The RDEs shall be in the same order as in the request, and the RDE Identifier field in each RDE shall be the value of the RDE Identifier field in the corresponding RDE in the request. The Status Code field in the RDE shall be set according to the result of the allocation request as follows:

- Status code = 0 indicates that the resource request has been accepted. The RDE shall also be followed by the Resource Descriptor that was accepted.
- Status code = nonzero indicates that the resources could not be accepted. The Status Code field contains a value from 8.4.1.9 indicating the reason for the failure. In this case, the AP may include a single Resource Descriptor following the RDE indicating a suggested resource that could have been

accepted. The Resource Count field shall be set to 0 or 1 depending whether the suggested Resource Descriptor is attached. A nonzero status code in an RDE shall not cause a nonzero status code in the frame containing the RIC.

If the resource request included QoS resources and is successful, then the procedures for handling of TSPEC, TCLAS, TCLAS Processing and Expedited Bandwidth Request elements shall be as specified in 10.4, and the AP shall place the TSs into the accepted state. The RIC-Response shall contain the updated accepted TSPEC. Each RDE may also include a Schedule element (as defined in 8.4.2.36) after the accepted TSPEC. Upon reassociation, AP shall move all of the TSs from the accepted state into the active state.

If the FTO does not invoke a reassociation within the reassociation deadline, then the TSs that had been accepted shall become inactive, and the resources shall be released. At the point that the FTO reassociates with the target AP (within the reassociation deadline, if appropriate), the TSs are put into the active state. This may be immediate if the RIC-Request was part of a Reassociation Request frame.

13. MLME mesh procedures

13.1 Mesh STA dependencies

When dot11MeshActivated is true, the STA is a mesh STA.

When dot11MeshActivated is true, following MIB attributes shall be set to true.

- dot11QosOptionImplemented
- dot11ExtendedChannelSwitchActivated
- dot11SpectrumManagementRequired

When dot11MeshActivated is true, following MIB attributes shall be set to false.

- dot11OCBActivated
- dot11FastBSSTransitionActivated

A mesh STA does not support functionalities that depend on AP or are only available in an infrastructure BSS, such as HCCA, traffic specifications (TSPECs), traffic stream (TS) management, admission control, automatic power save delivery (APSD), direct-link setup (DLS), or tunneled direct-link setup (TDLS).

An HT mesh STA does not support PSMP, STBC, or PCO.

13.2 Mesh discovery

13.2.1 General

A mesh STA shall perform either active scanning or passive scanning to discover an operating mesh BSS using the MLME-SCAN.request primitive (see 6.3.3). A mesh profile, a set of parameters identifying the mesh BSS configuration, is also obtained through the scanning process, and it is used to determine the scanning mesh STA's active mesh profile. Based on the result of the scan, the mesh STA may establish a new mesh BSS or become a member of the existing mesh BSS, using the START primitive (see 6.3.11). The MLME-START.request primitive triggers beaconing that facilitates the discovery of the mesh STA by the neighbor mesh STAs. A mesh STA that becomes a member of a mesh BSS should establish a mesh peering with one or more neighbor mesh STAs that are in the same mesh BSS.

13.2.2 Mesh identifier

The Mesh ID is an identifier of an MBSS. The Mesh ID may be installed in mesh capable devices by a variety of means that are beyond the scope of this standard. For example, the Mesh ID might be set by the user, e.g., "Mike's Mesh." A mesh STA shall include the Mesh ID element (see 8.4.2.101) in its Beacon and Probe Response frames, in order to advertise its identity. The mesh STA shall also include the Mesh ID element in its Mesh Peering Open frames, Mesh Peering Confirm frames, and Mesh Peering Close frames.

The mesh STA shall set the SSID element (see 8.4.2.2) in Beacon, Probe Request, and Probe Response frames to the wildcard SSID.

NOTE—The wildcard SSID is used to notify nonmesh STAs that the mesh STA is neither a part of an infrastructure BSS nor an IBSS, so that the nonmesh STAs do not try to join the mesh BSS.

13.2.3 Mesh profile

A mesh profile is a set of parameters that specifies the attributes of a mesh BSS. A mesh profile consists of the following:

a) A Mesh ID—specified by dot11MeshID
b) A path selection protocol identifier—specified by dot11MeshActivePathSelectionProtocol
c) A path selection metric identifier—specified by dot11MeshActivePathSelectionMetric
d) A congestion control mode identifier—specified by dot11MeshActiveCongestionControlMode
e) A synchronization method identifier—specified by dot11MeshActiveSynchronizationMethod
f) An authentication protocol identifier—specified by dot11MeshActiveAuthenticationProtocol

In a mesh BSS all mesh STAs use the same mesh profile. Two mesh profiles are considered to be the same if all of their parameters match.

Before establishing a mesh BSS or becoming a member of a mesh BSS, a mesh STA shall configure one mesh profile. The mesh STA shall not change its mesh profile unless it leaves the mesh BSS of which it is a member. When the mesh STA leaves the mesh BSS of which it is a member, it should explicitly close all of its active mesh peerings using Mesh Peering Close frames (see 13.3.8) and shall discard all session information obtained while the mesh profile was active, such as local forwarding information, security associations (and related keys), etc. The MLME receives the mesh STA's mesh profile from the SME upon receipt of the MLME-START.request primitive.

The mesh profile is signalled by means of the Mesh ID element and the Mesh Configuration element. The mesh profile is included in the Beacon and Probe Response frames, so that the mesh profile can be obtained by its neighbor mesh STAs through scanning. Mesh Peering Open and Mesh Peering Confirm frames also contain a mesh profile.

13.2.4 Mesh STA configuration

The mesh STA configuration consists of the mesh profile (see 13.2.3), the Supported Rates element, the Extended Supported Rates element, and the HT Operations element (if present).

Mesh STA configurations are identical if the following conditions hold:

— The mesh profiles are identical
— The BSSBasicRateSet parameters are identical
— For HT mesh STAs, the BSSBasicMCSSet parameters are identical

13.2.5 Supplemental information for the mesh discovery

A mesh STA shall signal whether it is able to establish additional mesh peerings. A mesh STA signals its ability to establish additional mesh peerings by setting the Accepting Additional Mesh Peerings subfield in the Mesh Capability field in the Mesh Configuration element to 1 (see 8.4.2.100.8).

The mesh STA sets the Accepting Additional Mesh Peerings subfield in the Mesh Capability field in the Mesh Configuration element to 0 when it is not able to accept new mesh peerings. This parameter is dynamically controlled by the SME and given to the MLME by dot11MeshAcceptingAdditionalPeerings.

NOTE—This control is driven by internal policies. When the Accepting Additional Mesh Peering subfield is 1, the mesh STA is assumed to have sufficient internal resources to accommodate more mesh peerings. The internal policy is outside the scope of this standard. For instance, a mesh STA might be configured to be able to maintain only two mesh peerings.

A mesh STA shall announce its topological information through the Mesh Formation Info field in the Mesh Configuration element. The contents of the Mesh Formation Info field shall be coded to reflect the current configuration.

13.2.6 Scanning mesh BSSs

A mesh STA shall perform active scanning or passive scanning, depending on the value of the ScanMode parameter of the MLME-SCAN.request primitive (see 10.1.4), to discover neighbor mesh STAs. Upon receipt of an MLME-SCAN.request primitive with the Mesh ID parameter set to the wildcard Mesh ID, the STA shall passively scan for any Beacon frames, or actively transmit Probe Request frames containing the wildcard Mesh ID, as appropriate, depending on the value of ScanMode. Upon completion of scanning, an MLME-SCAN.confirm primitive is issued by the MLME indicating all of the discovery information received. Further, mesh STAs shall conform to the passive scan procedure as described in 10.1.4.2 and the active scan procedure as described in 10.1.4.3.

13.2.7 Candidate peer mesh STA

When a mesh STA discovers a neighbor mesh STA through the scanning process and the discovered mesh STA is considered a candidate peer mesh STA, it may become a member of the mesh BSS of which the discovered mesh STA is a member and establish a mesh peering with the neighbor mesh STA.

The discovered neighbor mesh STA shall be considered a candidate peer mesh STA if and only if all of the following conditions are met:

a) The mesh STA uses the same mesh profile as the received Beacon or Probe Response frame indicates for the neighbor mesh STA.

   NOTE—If the scanning mesh STA has not become a member of any MBSS yet, it might simply activate the same mesh profile as the discovered neighbor mesh STA's profile to fulfill this condition.

b) The Accepting Additional Mesh Peerings subfield in the Mesh Capability field in the received Beacon or Probe Response frame equals 1.
c) The mesh STA supports the data rates indicated by the BSSBasicRateSet of the received Beacon or Probe Response frame.
d) If both the scanning mesh STA and the discovered neighbor STA are HT STAs, the mesh STA uses the same BSSBasicMCSSet as the received Beacon or Probe Response frame indicates for the neighbor mesh STA.
e) If the scanning mesh STA has dot11MeshSecurityActivated set to true and the dot11MeshActiveAuthenticationProtocol is ieee8021x (2), either the scanning mesh STA has an active connection to an AS or the discovered mesh STA has the Connected to AS subfield in the Mesh Formation field in the Mesh Configuration element equal to 1 in the received Beacon or Probe Response frame.

13.2.8 Establishing or becoming a member of a mesh BSS

The Mesh Formation Info field in the Mesh Configuration element is available to assist scanning mesh STAs in choosing the mesh BSS of which to become a member. The details of the usage of this information are beyond the scope of this standard.

NOTE—Selection of the mesh BSS of which the scanning mesh STA becomes a new member is outside the scope of this standard. That is, the mesh STA might freely select the mesh BSS of a candidate peer mesh STA of which it becomes a new member.

After the determination of the active mesh profile, the mesh STA may establish a new mesh BSS or become a new member to an existing mesh BSS.

When dot11MBCAActivated is true, the mesh STA shall perform the TBTT selection procedure described in 13.13.4.3 using TimeStamp, Local Time, Beacon Period, and Beacon Timing in the BSSDescription parameter given by the MLME-SCAN.confirm primitive, before starting its beaconing.

When dot11MCCAActivated is true, the mesh STA shall choose a DTIM interval with a duration of $2^n \times 100$ TU with n a non-negative integer less than or equal to 17.

NOTE—It is allowed that a different value for the DTIM interval is used for mesh STAs that use MCCA in an MBSS that is centrally controlled and the central authority provides a coordination of the DTIM interval of mesh STAs that use MCCA in the MBSS.

A mesh STA shall include a Country element in its Beacon frames if either dot11MultiDomainCapabilityActivated, dot11SpectrumManagementRequired, or dot11RadioMeasurementActivated is true. See 8.3.3.2 for the description of a properly formed Beacon frame.

The mesh STA establishes a new mesh BSS by activating a mesh profile that is different from any mesh profile discovered during the scanning of mesh BSSs (see 13.2.6).

The mesh STA becomes a new member of an existing mesh BSS by activating the same mesh profile as received from a candidate peer mesh STA of this mesh BSS (see 13.2.6 and 13.2.7).

In either case, the mesh STA shall start beaconing using the START primitive. Upon receipt of the MLME-START.request primitive, the mesh STA shall initialize and start its TSF timer as specified by its active synchronization method as described in 13.13.2, and begin transmitting Beacon frames as described in 13.13.3.

If the mesh STA has become a new member of an existing mesh BSS, it should establish a mesh peering with one or more candidate peer mesh STAs of this mesh BSS (see 13.2.9) in order to form the MBSS.

If the mesh STA has become a new member of an existing mesh BSS, it shall adopt the BSSBasicRateSet parameter from a candidate peer mesh STA of this mesh BSS.

After establishing or becoming a member of an MBSS, the mesh STA may continue the discovery procedure described in 13.2.6 to discover other candidate peer mesh STAs.

13.2.9 Establishing mesh peerings

Mesh peerings shall be established only with candidate mesh STAs that are members of the same MBSS.

A mesh peering is established between the mesh STA and the candidate peer mesh STA after the successful completion of the mesh peering management (MPM) protocol (see 13.3) or of the authenticated mesh peering exchange (AMPE) (see 13.5). When establishing a secure mesh peering, mesh STAs authenticate each other and create a mesh PMKSA before processing the AMPE (see 13.3.3).

A candidate peer mesh STA becomes a peer mesh STA when a mesh peering is established between the two mesh STAs.

When dot11MeshActiveSynchronizationMethod is neighborOffsetSynchronization (1), a mesh STA may establish mesh peerings with up to dot11MeshNbrOffsetMaxNeighbor mesh STAs (see 13.13.2.2.1).

13.3 Mesh peering management (MPM)

13.3.1 General

The mesh peering management (MPM) protocol is used to establish, maintain, and close mesh peerings between mesh STAs when dot11MeshSecurityActivated is false. When dot11MeshSecurityActivated is true, the peers establish an authenticated mesh peering using the authenticated mesh peering exchange (AMPE) protocol. The AMPE protocol requires an existing mesh PMKSA. If a mesh PMKSA with the candidate peer mesh STA exists, the AMPE shall use that mesh PMKSA. If no mesh PMKSA exists, the peers shall first authenticate to establish a mesh PMKSA; see 13.5.

Figure 13-1 shows the logical flow of protocol interactions in the peering management framework.

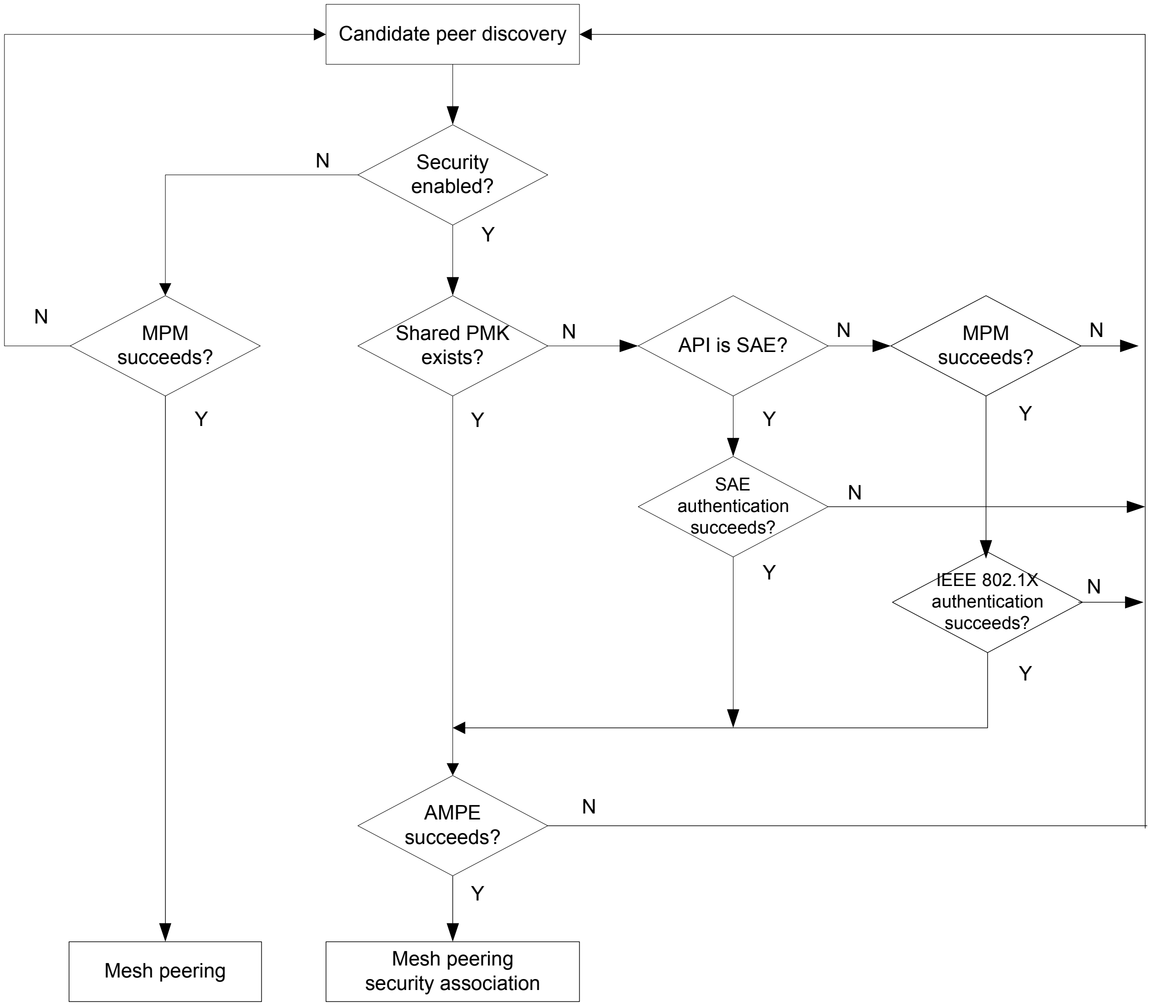


Figure 13-1—Logical flowchart of protocol interaction in the mesh peering management framework

The MPM protocol uses Mesh Peering Open frames, Mesh Peering Confirm frames, and Mesh Peering Close frames to establish, manage, and tear down a mesh peering.

The protocol succeeds in establishing a mesh peering when the following requirements are satisfied: 1) both mesh STAs have sent and received (and correctly processed) a Mesh Peering Open frame for this mesh peering; 2) both mesh STAs have sent and received (and correctly processed) a corresponding Mesh Peering Confirm frame for this mesh peering.

A mesh STA that receives and accepts a Mesh Peering Open frame (see 13.3.6.2) shall assign a unique AID among its neighbor peer mesh STAs to the transmitter of the frame. The AID is used in the encoding of the TIM element in the Beacon frame (see 8.4.2.7). AID 0 (zero) is reserved to indicate the presence of buffered group addressed MSDUs and MMPDUs (see 13.14.4).

13.3.2 State variable management

A mesh STA keeps an enumerated state variable (see 10.3.1) for each neighbor STA with which direct communication via the WM is needed. This state variable expresses the relationship between the local STA and a neighbor STA that varies depending on the active authentication protocol. It takes on the values shown in Table 13-1.

Table 13-1—State variables for mesh STAs

State	**Active authentication**		
	None	**SAE**	**IEEE 802.1X**
State 1	Initial start state, mesh peering not established	Initial start state, unauthenticated, mesh peering not established	Initial start state, unauthenticated, mesh peering not established
State 2	N/A	Authenticated, mesh peering not established	N/A
State 3	Mesh peering established	Authenticated, mesh peering established	Unauthenticated, mesh peering established (Pending IEEE 802.1X authentication)
State 4	N/A	N/A	Authenticated, mesh peering established

The state transitions in accordance with the protocol interaction shown in Figure 13-1.

The current state existing between the neighbor STAs determines the IEEE 802.11 frame types that may be exchanged between that pair of STAs (see Clause 8). The allowed frame types are grouped into classes and the classes correspond to the STA state. The allowed frame types and the frame classes in each state are defined in 10.3.3.

Mesh STAs shall not transmit frames other than the ones used for candidate peer mesh STA discovery, MPM, and SAE to a neighboring mesh STA until a mesh peering has been established with the mesh STA.

13.3.3 Mesh authentication

See 11.5.1.3.4.

13.3.4 Mesh peering instance controller

13.3.4.1 Overview

A mesh STA uses a mesh peering instance controller to manage all mesh peering instances.

The mesh peering instance controller performs the following functions:

- Create and destroy MPM finite state machines and AMPE finite state machines
- Manage instance identifiers for each mesh peering instance
- Manage mesh TKSAs for each mesh peering instance when dot11MeshSecurityActivated is true
- Preprocess the incoming Mesh Peering Management frames and pass the frames to the finite state machine with matching instance identifier
- Pass internal commands to the finite state machine with matching instance identifier

A mesh peering instance is identified by a mesh peering instance identifier. The mesh peering instance identifier is the set of localLinkID, localMAC, and peerMAC.

A mesh peering instance consists of its identifier (the localLinkID, localMAC, peerMAC), a peerLinkID (an integer generated by the peer mesh STA or candidate peer mesh STA), and the configuration and capability negotiated and agreed upon by exchanging Mesh Peering Open frames (see 8.5.16.2) and Mesh Peering Confirm frames (see 8.5.16.3). If dot11MeshSecurityActivated is true, the mesh peering instance also contains a PMKID identifying the shared PMKSA, a localNonce chosen by the mesh STA and a peerNonce chosen by the peer mesh STA or candidate peer mesh STA.

The localMAC is the MAC address of the mesh STA that is managing this mesh peering instance. The peerMAC is the MAC address of the peer mesh STA or the candidate peer mesh STA. The localLinkID is an integer generated by the mesh STA. The localLinkID shall be unique among all existing link identifiers used by the mesh STA for its MPM finite state machines. The mesh STA selects the localLinkID to provide high assurance that the same number has not been used to identify a recent MPM finite state machine. The peerLinkID is the localLinkID of the peer mesh STA or candidate peer mesh STA and is supplied in the Mesh Peering Management element (see 8.4.2.104) of the Mesh Peering Open and Mesh Peering Confirm frames.

A mesh peering instance is controlled by an MPM finite state machine (see Table 13-2) or an AMPE finite state machine (see Table 13-3).

13.3.4.2 Creating a new mesh peering instance

The mesh peering instance controller creates a new mesh peering instance after either of the following two events:

- The receipt of a Mesh Peering Open frame from a candidate peer mesh STA according to the rules of 13.3.5
- The receipt of an MLME-MESHPEERINGMANAGEMENT.request primitive with a Mesh Peering Open frame

A unique localLinkID shall be generated for the mesh peering instance. If the mesh peering instance is established by AMPE, a random local nonce shall also be generated.

A mesh STA may create multiple mesh peering instances to establish a peering with the same candidate peer mesh STA.

13.3.4.3 Deleting mesh peering instances

The mesh peering instance controller deletes a mesh peering instance after either:

- Expiry of a holding timer (see 13.4.4).
- The acceptance of a peer's response to an existing request to close the peering (see 13.4.3).
- Indication from the SME that the peer mesh STA, or candidate peer mesh STA, has deauthenticated.

When the deletion occurs, the mesh TKSA that is bound to the mesh peering shall be deleted.

13.3.5 Mesh peering instance selection

The content of a Mesh Peering Management frame received from a candidate peer mesh STA, and the set of mesh peering instances in the mesh peering instance controller determine whether

- A new mesh peering instance is created (see 13.3.4.2); or,
- An existing mesh peering instance is updated

If dot11MeshSecurityActivated is true and the mesh STA shares a PMK with the candidate peer mesh STA but the Mesh Peering Protocol Identifier field in the Mesh Peering Management element of the frame indicates "mesh peering management protocol," the frame shall be silently discarded.

If dot11MeshSecurityActivated is true and the mesh STA shares a PMK with the candidate peer mesh STA but either the Mesh Peering element or the MIC element are not present in the frame, the frame shall be silently discarded.

If dot11MeshSecurityActivated is false but the Mesh Peering Protocol Identifier field in the Mesh Peering Management element of the received frame indicates "authenticated mesh peering exchange," the frame shall be silently discarded.

If dot11MeshSecurityActivated is false but either the Mesh Peering element or the MIC element is present in the frame, the frame shall be silently discarded.

If the frame contains a group address in TA or RA, it shall be silently discarded.

If the incoming Mesh Peering Management frame is for AMPE and the Chosen PMK from the received frame contains a PMKID that does not identify a valid mesh PMKSA, the frame shall be silently discarded.

If the Mesh Peering Management frame has not been silently discarded, the mesh peering instance controller attempts to locate a matching mesh peering instance identifier. A match is determined by comparing the contents of the Mesh Peering Management frame with each peering instance. A match is found if all the following conditions are true:

- The transmitter's MAC address (Address 2) is the same as the peerMAC of the mesh peering instance
- The receiver's MAC address (Address 1) is the same as the localMAC of the mesh peering instance
- The value of the Peer Link ID field is the same as the localLinkID of the mesh peering instance

If the incoming frame is a Mesh Peering Open frame and no matching peering instance was found, a new mesh peering instance is created (and a new Mesh TSKA if dot11MeshSecurityActivated is true). See 13.3.4.2.

If the incoming frame is a Mesh Peering Confirm or Mesh Peering Close frame and no matching mesh peering instance is found, it shall be silently discarded.

If the incoming Mesh Peering Management frame is for AMPE and has not been discarded it shall be further processed as follows:

- If the Peer Nonce field is present in the received frame, and the localNonce in the mesh peering instance is different than the Peer Nonce field of the received frame, the frame shall be dropped.
- If the peerNonce in the mesh peering instance exists and is different than the Local Nonce field of the received frame, the frame shall be dropped.

13.3.6 Mesh peering open

13.3.6.1 Generating Mesh Peering Open frames

A Mesh Peering Open frame is generated as a result of a sendOpen() action (see 13.4.3).

The contents of the frame are described in 8.5.16.2.2.

13.3.6.2 Mesh Peering Open frame processing

The mesh STA checks that the Mesh ID element and Mesh Configuration element of the Mesh Peering Open frame is identical to its own mesh STA configuration as specified in 13.2.3 and 13.2.4. If a mismatch is found the frame shall be rejected with a reason code of MESH-CONFIGURATION-POLICY-VIOLATION and the mesh peering establishment attempt shall be terminated.

When the mesh STA has established a mesh PMKSA with the candidate peer mesh STA, the mesh peering instance controller shall silently discard the Mesh Peering Open frame in the following two conditions:

- The Mesh Peering Open frame supports MPM protocol and the negotiated active authentication is SAE, or
- The Mesh Peering Open frame supports AMPE but the PMKID in the Chosen PMK field in the Authenticated Mesh Peering Exchange element does not identify a mesh PMKSA.

If the Mesh Peering Open frame is not discarded, the mesh peering instance controller actively rejects or accepts the mesh peering open request (see 13.4). If dot11MeshAcceptingAdditionalPeerings is set to zero the Mesh Peering Open request shall be rejected with reason code MESH-MAX-PEERS.

If the peerLinkID in the mesh peering instance has not been set, the Local Link ID field of the Mesh Peering Open request shall be copied into the peerLinkID in the mesh peering instance. If the incoming Mesh Peering Open frame is for AMPE and the peerNonce in the mesh peering instance has not been set, the Local Nonce field in the incoming Mesh Peering Open frame shall be copied into the peerNonce in the mesh peering instance.

The mesh peering open request may be rejected due to an internal reason with a reason code of MESH-PEERING-CANCELED.

If the Mesh Peering Open request is rejected, the REQ_RJCT event shall be passed with the specified reason code to the protocol finite state machine to actively reject the mesh peering open request.

NOTE—Example internal reasons to reject new mesh peering request could be the mesh STA has reached its capacity to set up more mesh peering, the mesh STA is configured to reject mesh peering request from another specific peer mesh STA.

13.3.7 Mesh peering confirm

13.3.7.1 Generating Mesh Peering Confirm frames

A Mesh Peering Confirm frame is generated as a result of a sendConfirm() action (see 13.4.3).

The contents of the frame are described in 8.5.16.3.2.

13.3.7.2 Mesh Peering Confirm frame processing

The mesh STA shall check that the Mesh ID element and Mesh Configuration element of the Mesh Peering Confirm frame match its own mesh STA configuration as specified in 13.2.3 and 13.2.4. If a mismatch is found, the frame shall be rejected with the reason code of MESH-INCONSISTENT-PARAMETERS.

Otherwise, the mesh STA accepts the Mesh Peering Confirm frame and performs the actions described in 13.4.

If the peerLinkID in the mesh peering instance has not been set, the Local Link ID field of the Mesh Peering Confirm request shall be copied into the peerLinkID in the mesh peering instance. If the incoming Mesh Peering Confirm frame is for AMPE and the peerNonce in the mesh peering instance has not been set, the Local Nonce field in the incoming Mesh Peering Confirm frame shall be copied into the peerNonce in the mesh peering instance.

13.3.8 Mesh peering close

13.3.8.1 Generating Mesh Peering Close frames

A Mesh Peering Close frame is generated as a result of a sendClose() action (see 13.4.3).

The contents of the frame are described in 8.5.16.4.2.

When the Mesh Peering Close is generated as a result of a CNCL event, the reason code is MESH-PEERING-CANCELLED. When the Mesh Peering Close is generated as a result of a CLS_ACPT event, the reason code is MESH-CLOSE-RCVD.

13.3.8.2 Mesh Peering Close frame processing

The mesh STA shall reject the Mesh Peering Close frame if the value in the Mesh ID element is not the same as the mesh STA's mesh profile. Otherwise, the mesh STA accepts the Mesh Peering Close frame and performs the actions described in 13.4.

13.4 Mesh peering management finite state machine (MPM FSM)

13.4.1 General

Each mesh peering instance, including its states and resource, are managed by a mesh peering management finite state machine (MPM FSM). The MPM FSM uses MLME primitives to control the mesh STA to send and receive Mesh Peering Management frames.

13.4.2 States

The MPM FSM uses the following six states:

- IDLE—IDLE state is a terminal state. In the IDLE state, the MPM FSM is ready to start a new mesh peering instance by either passively listening for an incoming Mesh Peering Open frame or actively initiating a mesh peering instance.
- OPN_SNT—In the OPN_SNT state, the finite state machine has sent a Mesh Peering Open frame and is waiting for a Mesh Peering Open frame and Mesh Peering Confirm frame from the candidate peer mesh STA.
- CNF_RCVD—In the CNF_RCVD state, the finite state machine has received a Mesh Peering Confirm frame, but has not received a Mesh Peering Open frame. The mesh STA has not sent the corresponding Mesh Peering Confirm frame yet.
- OPN_RCVD—In the OPN_RCVD state, the finite state machine has received only the Mesh Peering Open frame but not the Mesh Peering Confirm. The mesh STA has also sent a Mesh Peering Confirm frame upon receiving a Mesh Peering Open frame.
- ESTAB—In the ESTAB state, the finite state machine has received both the Mesh Peering Open and Mesh Peering Confirm frames. The mesh STA has also sent both the Mesh Peering Open frame and Mesh Peering Confirm frame. The mesh peering is established and configured for exchanging frames with the peer mesh STA in the ESTAB state.
- HOLDING—In the HOLDING state, the finite state machine is closing the mesh peering instance with the peer mesh STA or the candidate peer mesh STA.

13.4.3 Events and actions

The finite state machine uses three types of events: 1) events for state machine transitions; 2) external events generated by frame processing; and 3) events associated with internal timers.

The events for state machine transitions are as follows:

- CNCL(localLinkID, peerMAC, ReasonCode)—Used to instruct the mesh peering instance to cancel the mesh peering with the peer mesh STA. localLinkID identifies the MPM FSM for the corresponding mesh peering instance. peerMAC is the MAC address of the peer mesh entity. ReasonCode is used to inform the reason to cancel the mesh peering instance. See 13.3.8.2.
- ACTOPN(peerMAC, localLinkID)—The SME uses this event to create a new mesh peering instance to actively initiate the mesh peering establishment with the candidate peer mesh STA whose MAC address is peerMAC. localLinkID identifies the MPM FSM.

The events generated by frame processing are as follows:

- OPN_ACPT—PeeringOpen_Accept(peerMAC, peerLinkID) event indicates that a Mesh Peering Open frame meeting the correctness criteria of 13.3.6 has been received from peerMAC for the mesh peering instance identified by peerLinkID.
- OPN_RJCT—PeeringOpen_Reject(peerMAC, peerLinkID, Configuration, reasonCode) event indicates that a Mesh Peering Open frame from peerMAC for the mesh peering instance identified by peerLinkID is rejected due to incomplete or erroneous configuration, as indicated by the Configuration, with reasonCode being the specific reason for rejection of the Mesh Peering Open frame. See 13.3.6.2.
- CNF_ACPT—PeeringConfirm_Accept(peerMAC, localLinkID, peerLinkID) event indicates that a Mesh Peering Confirm frame meeting the correctness criteria of 13.3.7 has been received from peerMAC for the mesh peering instance identified by localLinkID and peerLinkID.
- CNF_RJCT—PeeringConfirm_Reject(peerMAC, localLinkID, peerLinkID, reasonCode) event indicates that a Mesh Peering Confirm frame from peerMAC for the mesh peering instance identified by localLinkID and peerLinkID is rejected due to incomplete or erroneous configuration, and reasonCode is the specific reason for rejection of the Confirm frame. See 13.3.7.2.

- CLS_ACPT—PeeringClose_Accept(peerMAC, localLinkID, peerLinkID, reasonCode) event indicates that a Mesh Peering Close frame meeting the correctness criteria of 13.3.8 has been received from peerMAC for the mesh peering instance identified by localLinkID and peerLinkID. The reasonCode specifies the reason that caused the generation of the Mesh Peering Close frame. See 13.3.8.2.
- REQ_RJCT—PeeringRequest_Reject(peerMAC, peerLinkID, reasonCode) event indicates a special incidence that the mesh STA rejects the incoming Mesh Peering Open frame requesting to set up a new mesh peering for some specified reason. The incoming request is identified by the peerMAC, peerLinkID is the peerLinkID received from the Mesh Peering Open frame, and reasonCode is the specific reason for rejection of the Mesh Peering Open frame. See 13.3.6.2.

The finite state machine may take an action triggered by an event. It uses two types of actions: sending a Mesh Peering Management frame and handling a timer.

Actions related to sending a Mesh Peering Management frame are as follows:

- sndOPN—sendOpen(peerMAC, localLinkID, Configuration) is the action that the mesh STA takes to send a Mesh Peering Open frame to the candidate peer mesh STA, whose MAC address is peerMAC. The MLME-MESHPEERINGMANAGEMENT.request primitive shall be invoked to send the frame to the peer mesh entity.
- sndCNF—sendConfirm(peerMAC, localLinkID, peerLinkID, Configuration) is the action that the mesh STA takes to send a Mesh Peering Confirm frame to the candidate peer mesh STA, whose MAC address is peerMAC. The MLME-MESHPEERINGMANAGEMENT.request primitive shall be invoked to send the frame to the peer mesh entity.
- sndCLS—sendClose(peerMAC, localLinkID, peerLinkID, reasonCode) is the action that the mesh STA takes to send a Mesh Peering Close frame to the peer mesh STA or candidate peer mesh STA, whose MAC address is peerMAC. The MLME-MESHPEERINGMANAGEMENT.request primitive shall be invoked to send the frame to the peer mesh entity.

13.4.4 Timers

The following three timers are used by the finite state machine:

a) The retryTimer triggers a resend of the Mesh Peering Open frame when a Mesh Peering Confirm frame was not received as a response. The retryTimer is set to the dot11MeshRetryTimeout.

b) The confirmTimer signals that a link establishment attempt should be aborted because a Mesh Peering Confirm frame responding to a Mesh Peering Open frame was never received. The confirmTimer is set to the value of dot11MeshConfirmTimeout.

c) The holdingTimer signals that its mesh peering instance may be completely closed and facilitates graceful shutdown. The holdingTimer is set to the value of dot11MeshHoldingTimeout.

The events associated with internal timers are indicated in the state machine as acronyms that indicate timer expiry. With each timer event there is an associated action.

- TOR1—This event indicates that the retryTimer has expired and dot11MeshMaxRetries has not been reached. The Mesh Peering Open frame shall be resent, an action indicated in the state machine by setR.
- TOR2—This event indicates that the retryTimer has expired and dot11MeshMaxRetries has been reached. The mesh peering instance shall be closed when TOR2 occurs.
- TOC—This event indicates that the confirmTimer has expired. When TOC event occurs, the mesh peering instance shall be closed, an action indicated in the state machine as setC.

— TOH—This event indicates that the holdingTimer has expired. When TOH occurs, the mesh peering instance shall be closed and the finite state machine shall transition to IDLE state, an action indicated in the state machine as setH.

13.4.5 State transitions

Table 13-2 and Figure 13-2 summarize the state transitions for the MPM protocol.

In Table 13-2, each row represents state transitions from the state to all other states. A blank entry indicates an impossible transition.

Table 13-2—MPM finite state machine

		To State					
		IDLE	OPN_SNT	CNF_RCVD	OPN_RCVD	ESTAB	HOLDING
From State	IDLE	REQ_RJCT / sndCLS	ACTOPN / (sndOPN, setR)		OPN_ACPT / (sndOPN, sndCNF, setR)		
	OPN_SNT		TOR1 / (sndOPN, setR)	CNF_ACPT / (clR, setC)	OPN_ACPT / (sndCNF)		CLS_ACPT, OPN_RJCT, CNF_RJCT, TOR2, CNCL / (sndCLS, clR, setH)
	CNF_RCVD					OPN_ACPT / (clC, sndCNF)	CLS_ACPT, OPN_RJCT, CNF_RJCT, CNCL / (sndCLS, clC, setH) TOC / (sndCLS, setH)
	OPN_RCVD				OPN_ACPT / sndCNF TOR1 / (sndOPN, setR)	CNF_ACPT / clR	CLS_ACPT, OPN_RJCT, CNF_RJCT, TOR2, CNCL / (sndCLS, clR, setH)
	ESTAB					OPN_ACPT / sndCNF	CLS_ACPT, OPN_RJCT, CNF_RJCT, CNCL / (sndCLS, setH)
	HOLDING	TOH / —, CLS_ACPT / clH					OPN_ACPT, CNF_ACPT, OPN_RJCT, CNF_RJCT / sndCLS

In Figure 13-2, each arrow represents a state transition.

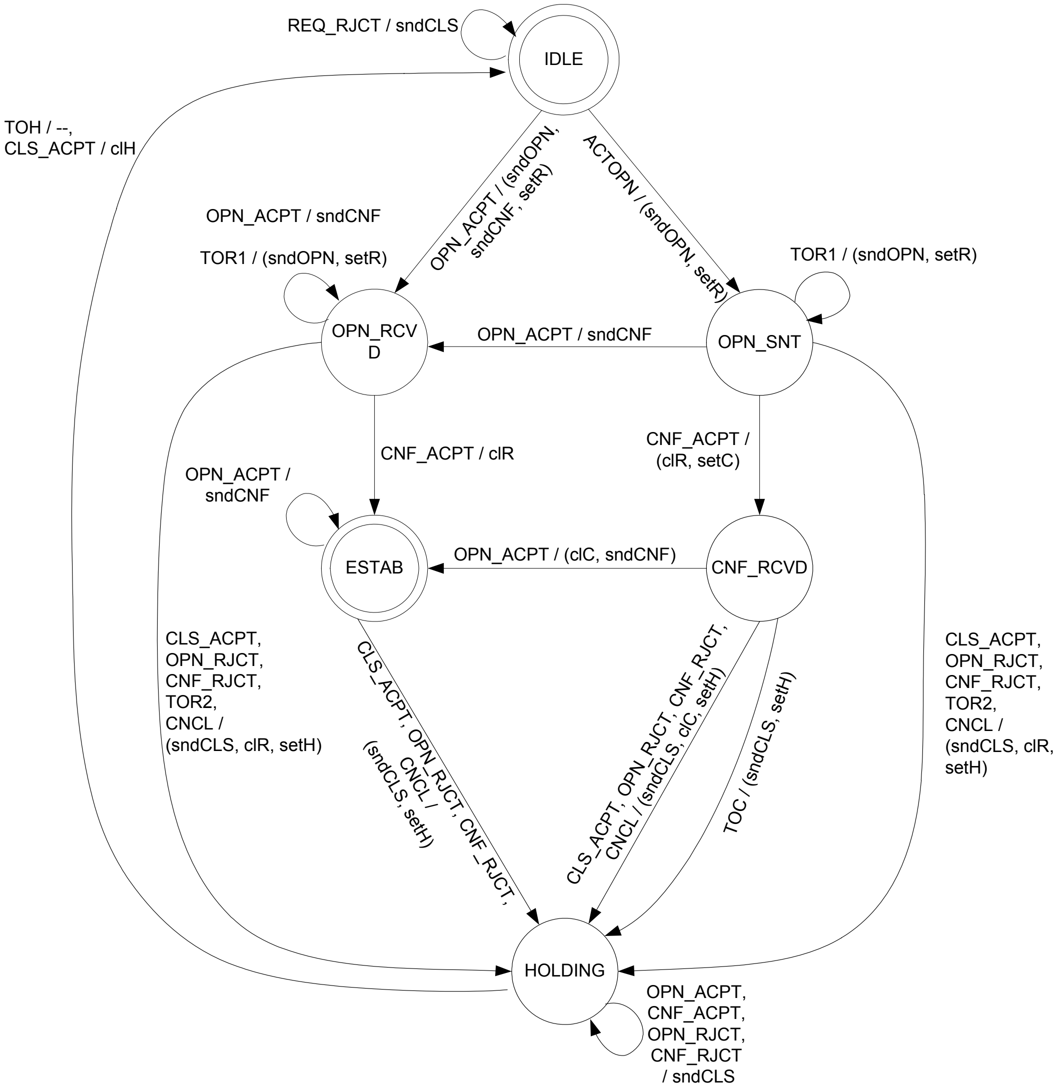

REQ_RJCT / sndCLS
IDLE
TOH / --,
CLS_ACPT / clH
OPN_ACPT / (sndOPN, sndCNF, setR)
ACTOPN / (sndOPN, setR)
OPN_ACPT / sndCNF
TOR1 / (sndOPN, setR)
TOR1 / (sndOPN, setR)
OPN_RCVD
OPN_ACPT / sndCNF
OPN_SNT
CNF_ACPT / clR
CNF_ACPT / (clR, setC)
OPN_ACPT / sndCNF
ESTAB
OPN_ACPT / (clC, sndCNF)
CNF_RCVD
CLS_ACPT, OPN_RJCT, CNF_RJCT, TOR2, CNCL / (sndCLS, clR, setH)
CLS_ACPT, OPN_RJCT, CNF_RJCT, CNCL / (sndCLS, setH)
CLS_ACPT, OPN_RJCT, CNF_RJCT, CNCL / (sndCLS, clC, setH)
TOC / (sndCLS, setH)
CLS_ACPT, OPN_RJCT, CNF_RJCT, TOR2, CNCL / (sndCLS, clR, setH)
HOLDING
OPN_ACPT, CNF_ACPT, OPN_RJCT, CNF_RJCT / sndCLS

Figure 13-2—Finite state machine of the MPM protocol

The event/action representation is defined as the following. "E/A" string represents that the action A is taken given that the event E occurs. "E1, E2/A" string represents that the action A is taken given that the event E1 or event E2 occurs. "E/(A1, A2)" string represents that the action A1 and A2 are taken at a time when event E occurs.

Note that Table 13-2 and Figure 13-2 are used for illustration purposes. The protocol behavior is in the following subclauses.

13.4.6 IDLE state

IDLE is a quiescent state the finite state machine enters prior to establishing a new mesh peering.

When ACTOPN event occurs, the mesh STA shall set the retryCounter to zero, and perform a sndOPN action. The retryTimer shall be set and the finite state machine shall transition to OPN_SNT state.

When an OPN_ACPT event occurs, the mesh STA shall perform a sndOPN action and sndCNF action, and set the retryTimer. The finite state machine shall transition to OPN_RCVD state.

When an REQ_RJCT event occurs, a Mesh Peering Close frame shall be sent to reject the mesh peering open request. The reason code in the Mesh Peering Close frame shall be set to the reason code in REQ_RJCT event. The finite state machine shall stay in the IDLE state.

All other events shall be ignored in this state.

13.4.7 OPN_SNT state

In the OPN_SNT state, the mesh STA waits for a Mesh Peering Confirm frame. In this state, the retryTimer is set.

When a CNCL event occurs, the mesh STA shall clear the retryTimer, perform a sndCLS using the reason code specified by the CNCL event, and set the holdingTimer. The finite state machine shall transition to HOLDING state.

When a CLS_ACPT event occurs, the mesh STA shall clear the retryTimer, perform a sndCLS using the reason code specified by the CLS_ACPT event, and set the holdingTimer. The finite state machine shall transition to HOLDING state.

When an OPN_ACPT event occurs, the mesh STA shall send perform a sndCNF action. The finite state machine shall transition to OPN_RCVD state.

NOTE—The retryTimer is still in effect after the state transition.

When an OPN_RJCT event occurs, the mesh STA shall clear the retryTimer, perform a sndCLS using the reason code specified by the OPN_RJCT event, and set the holdingTimer. The finite state machine shall transition to HOLDING state.

When a CNF_ACPT event occurs, the mesh STA shall clear the retryTimer and shall set the confirmTimer and the finite state machine shall transition to CNF_RCVD state.

When a CNF_RJCT event occurs, the mesh STA shall clear the retryTimer, perform a sndCLS action using the reason code specified by the CNF_RJCT event, and set the holdingTimer. The finite state machine shall transition to HOLDING state.

When a TOR1 event occurs, the Mesh STA shall perform a sndOPN action and the retryCounter shall be incremented. The retryTimer shall be and the finite state machine shall stay in the OPN_SNT state.

When a TOR2 event occurs, the mesh STA shall perform a sndCLS action using the reason code MESH-MAX-RETRIES. The holdingTimer shall be set, and the finite state machine shall transition to HOLDING state.

All other events shall be ignored in this state.

13.4.8 CNF_RCVD state

In the CNF_RCVD state, the mesh STA has received a Mesh Peering Confirm frame and is waiting for a Mesh Peering Open frame.

When a CNCL event occurs, the mesh STA shall clear the confirmTimer, perform a sndCLS action using the reason code MESH-PEERING-CANCELLED, and set the holdingTimer. The finite state machine shall transition to HOLDING state.

When a CLS_ACPT event occurs, the mesh STA shall clear the confirmTimer, perform a sndCLS using the reason code MESH-CLOSE-RCVD, and set the holdingTimer. The finite state machine shall transition to HOLDING state.

When an OPN_ACPT event occurs, the mesh STA shall clear the confirmTimer and shall perform a sndCNF action. The finite state machine shall transition to ESTAB state.

When an OPN_RJCT event occurs, the mesh STA shall clear the confirmTimer, perform a sndCLS action using the reason code as specified by the OPN_RJCT event, and set the holdingTimer. The finite state machine shall transition to HOLDING state.

When a CNF_RJCT event occurs, the mesh STA shall clear the confirmTimer, perform a sndCLS action using the reason code as specified by the CNF_RJCT event, and set the holdingTimer The finite state machine shall transition to HOLDING state.

When TOC event occurs, the mesh STA shall perform a sndCLS action using the reason code MESH-CONFIRM-TIMEOUT and set the holdingTimer. The finite state machine shall transition to HOLDING state.

All other events shall be ignored in this state.

13.4.9 OPN_RCVD state

In the OPN_RCVD state, the mesh STA has received a Mesh Peering Open frame and sent a Mesh Peering Open frame and the corresponding Mesh Peering Confirm frame. An incoming Mesh Peering Confirm is expected.

When a CNCL event occurs, the mesh STA shall clear the retryTimer, perform a sndCLS action using the reason code MESH-PEERING-CANCELLED, and set the holdingTimer. The finite state machine shall transition to HOLDING state.

When a CLS_ACPT event occurs, the mesh STA shall clear the retryTimer, perform a sndCLS action, and set the holdingTimer. The finite state machine shall transition to HOLDING state.

When an OPN_ACPT event occurs, the mesh STA shall perform a sndCNF action. The finite state machine shall stay in the OPN_RCVD state.

When an OPN_RJCT event occurs, the mesh STA shall clear the retryTimer, perform a sndCLS action using the reason code as specified by the OPN_RJCT event, and set the holdingTimer. The finite state machine shall transition to HOLDING state.

When a CNF_ACPT event occurs, the retryTimer shall be cleared. The finite state machine shall transition to ESTAB state.

When a CNF_RJCT event occurs, the mesh STA shall clear the retryTimer, perform a sndCLS action using the reason code as specified by the CNF_RJCT event, and set the holdingTimer. The finite state machine shall transition to HOLDING state.

When a TOR1 event occurs, the Mesh STA shall perform a sndOPN action, increment the retryCounter, and set the retryTimer. The finite state machine shall stay in the OPN_RCVD state.

When a TOR2 event occurs, the mesh STA shall perform a sndCLS action using the reason code MESH-MAX-RETRIES. The holdingTimer shall be set, and the finite state machine shall transition to HOLDING state.

All other events shall be ignored in this state.

13.4.10 ESTAB state

In the ESTAB state, mesh peering has been successfully established with the peer mesh STA.

When a CNCL event occurs, the mesh STA shall perform a sndCLS action using the reason code MESH-PEERING-CANCELLED, and set the holdingTimer. The finite state machine shall transition to HOLDING state.

When a CLS_ACPT event occurs, the mesh STA shall perform a sndCLS action using the reason code MESH-CLOSE-RCVD, and set the holdingTimer. The finite state machine shall transition to HOLDING state.

When an OPN_ACPT event occurs, the mesh STA shall respond by performing a sndCNF action. The finite state machine shall stay in the ESTAB state.

All other events shall be ignored in this state.

13.4.11 HOLDING state

In HOLDING state, the mesh STA is closing the mesh peering. The holdingTimer has been set according to the value of dot11MeshHoldingTimeOut.

When a CLS_ACPT event occurs, the holdingTimer shall be cleared. The finite state machine shall transition to IDLE state.

When any of the following four events occurs—OPN ACPT, CNF_ACPT, OPN_RJCT, CNF_RJCT—the mesh STA shall send a Mesh Peering Close frame. The finite state machine shall stay in the HOLDING state.

When a TOH event occurs, the finite state machine shall transition to IDLE state.

All other events are ignored in this state.

13.5 Authenticated mesh peering exchange (AMPE)

13.5.1 Overview

The authenticated mesh peering exchange (AMPE) establishes an authenticated mesh peering between the mesh STAs, under the assumption that mesh PMKSA has already been established before the initiation of the protocol. An authenticated mesh peering includes a mesh peering, corresponding mesh TKSA, and the two mesh STAs mesh GTKSAs.

The AMPE uses Mesh Peering Management frames. Parameters are exchanged via the RSNE, the Authenticated Mesh Peering Exchange element, and the MIC element.

The major functions provided by AMPE are security capabilities selection, key confirmation, and key management.

- The security capabilities selection function (specified in 13.5.2) is performed by agreeing on the security parameters used for the protocol instance.
- Key confirmation using the shared Mesh PMK is performed by verifying that the protection on the Mesh Peering Management frames is correct.

— Key management (specified in 13.5.7) is performed by the derivation of the temporal key in the mesh TKSA and the exchange of each mesh STA's MGTK.

During the AMPE handshake, the mesh STAs generate nonces and transmit them via Mesh Peering Management frames. The mesh STA shall generate a random value for its localNonce, as specified in 11.6.5. The candidate peer mesh STA is expected to generate a random value for the peerNonce, which the mesh STA receives from the candidate peer mesh STA in Confirm and Close Action frames.

Mesh Peering Management frames used in the AMPE are protected using the deterministic authenticated encryption mode of AES-SIV (IETF RFC 5297).

13.5.2 Security capabilities selection

13.5.2.1 Instance Pairwise Cipher Suite selection

Pairwise cipher suite selectors WEP-40, WEP-104, and TKIP shall not be used as the pairwise cipher suite when dot11MeshSecurityActivated is enabled.

If the pairwise cipher suite has not been selected, mesh STAs shall attempt to reach the agreement on the pairwise cipher suite using the following procedure in four steps:

a) The mesh STA shall announce the list of pairwise cipher suites it supports using an ordered list in the RSNE in the Mesh Peering Open frame. The first value in the list is the mesh STA's most preferred cipher suite, and the last value the least preferred.
b) If the mesh STA receives a Mesh Peering Open frame from the candidate peer mesh STA, the mesh STA shall make its decision on the selected pairwise cipher suite based on the intersection of its own ordered list and the received ordered list.
    1) If the intersection is empty, the pairwise cipher suite selection fails and the mesh STA generates the failure reason code MESH-INVALID-SECURITY-CAPABILITY and then takes the corresponding actions specified in 13.5.6.
    2) If the intersection contains more than one value, the selected cipher suite shall be the entry in the intersection list most preferred by the mesh STA that has the largest MAC address in the lexicographic ordering.
c) If the mesh STA receives a Mesh Peering Confirm frame from the candidate peer mesh STA before receiving a Mesh Peering Open frame, the mesh STA shall verify that it supports the pairwise cipher suite chosen by the candidate peer mesh STA. Otherwise, the selection fails and the mesh STA shall generate the failure reason code MESH-INVALID-SECURITY-CAPABILITY.
   Furthermore, upon receiving a Mesh Peering Open frame, the mesh STA shall verify that the accepted selected pairwise cipher suite matches the pairwise cipher suite chosen in step b). If they do not match, the selection fails and the mesh STA shall generate the failure reason code MESH-INVALID-SECURITY-CAPABILITY. Otherwise, the pairwise cipher suite selection succeeds, and the mesh STA shall proceed to step d).
d) If the mesh STA is generating a Mesh Peering Confirm frame, it shall set the Selected Pairwise Cipher Suite to the selected pairwise cipher suite upon successful pairwise cipher suite selection.

13.5.2.2 Group cipher suite selection

Group cipher suite selectors WEP-40, WEP-104, and TKIP shall not be used as the group cipher suite when dot11MeshSecurityActivated is true.

The mesh STA shall not use a different group cipher suite than the one used by the peer mesh STA or candidate peer mesh STA in the same MBSS.

A mesh STA shall announce in a Mesh Peering Open action frame the group cipher suite it uses for broadcast protection. When it receives a Mesh Peering Open frame from a candidate peer, it shall verify that it supports the candidate's announced group cipher suite. In addition, if the mesh STA receives a Mesh Peering Confirm frame, it shall verify that it supports the group cipher suite listed in that frame. If either selection fails, the mesh STA shall issue the appropriate reply frame with the MESH-INVALID-SECURITY-CAPABILITY reason code.

13.5.3 Construction and processing AES-SIV-protected Mesh Peering Management frames

AES-SIV performs deterministic authenticated encryption and takes additional data that is authenticated but not encrypted (AAD). When encrypting and authenticating, AES-SIV takes a key, plaintext data to protect, and multiple distinct components of AAD, to produce a synthetic initialization vector and a cipher text. When verifying encrypted and authenticated data AES-SIV takes a key, a synthetic initialization vector, cipher text data to decrypt and verify, and AAD, to produce either plaintext or the symbol "FAIL," indicating failure to decrypt and verify. Note that the AAD used in the encryption process shall be identical to the AAD used in the decryption process and the synthetic initialization vector produced by the encryption process shall be used in the decryption process.

When the mesh STA constructs a Mesh Peering Management frame, it shall follow the following procedure:

- The input key shall be the AEK
- The input plaintext shall be the Authenticated Mesh Peering element (see 8.5.16.2, 8.5.16.3, 8.5.16.4)
- The input AAD shall be three distinct components consisting of
  1) The localMAC
  2) The peerMAC
  3) The contents of the Mesh Peering Management frame from the category (inclusive) to the MIC element (exclusive)
- The output synthetic initialization vector shall be copied into the MIC field of the MIC element in the Mesh Peering Management frame
- The output cipher text shall become the remainder of the Mesh Peering Management frame after the MIC element

When the mesh STA verifies a Mesh Peering Management frame, it shall follow the following procedure:

- The input key shall be the AEK
- The input synthetic initialization vector shall be the MIC field of the MIC element in the Mesh Peering Management frame
- The input cipher text shall be the part of the Mesh Peering Management frame following the MIC element
- The input AAD shall be three distinct components consisting of
  1) The peerMAC
  2) The localMAC
  3) The contents of the Mesh Peering Management frame from the category (inclusive) to the MIC element (exclusive)
- If AES-SIV returns the symbol "FAIL" processing of the frame shall be deemed a failure with a behavior dependent on the type of Mesh Peering Management frame
- If AES-SIV returns plaintext it shall be treated as the components of the Mesh Peering Management frame and processed accordingly

13.5.4 Distribution of group transient keys in an MBSS

The MGTK shall be a random or pseudorandom number. The mesh STA shall distribute the MGTK to the peer mesh STA using the Mesh Peering Open frame during the AMPE. Upon successful completion of AMPE, each mesh STA shall establish states for the peer mesh STA's mesh GTKSA. The GTKData subfield in the Authenticated Mesh Peering Exchange element shall contain the MGTK concatenated by the Key RSC and the GTKExpirationTime (as indicated in 8.4.2.120).

When dot11RSNAProtectedManagementFramesActivated is true, a mesh STA shall distribute the IGTK to the peer mesh STA using the Mesh Peering Open frame during the AMPE. Upon successful completion of AMPE, each mesh STA shall establish an IGTKSA (see 11.5.1.1.9) with the mesh peer.

13.5.5 Mesh Peering Management frames for AMPE

13.5.5.1 General

The AMPE is inclusive of the mesh peering management (MPM) protocol. Mesh Peering Management frames for AMPE have additional processing and construction requirements on top of those for Mesh Peering Management frames.

The Mesh Peering Management frames shall be generated with additional information using the RSNE and the Authenticated Mesh Peering Exchange element to support AMPE.

13.5.5.2 Mesh peering open for AMPE

13.5.5.2.1 Generating Mesh Peering Open frames for AMPE

In addition to contents for establishing a mesh peering as specified in 13.3.6.1, the Mesh Peering Open frame, when used for the AMPE, shall contain the following:

- In the Mesh Peering Management element, the Mesh Peering Protocol Identifier shall be set to 1 "authenticated mesh peering exchange protocol."
- In the Mesh Peering Management element, the Chosen PMK field shall be set to PMKID that identifies the mesh PMKSA the mesh STA established with the candidate peer mesh STA.
- The RSNE shall be identical to the RSNE in the STA's Beacon and Probe Response frames.
- In the Authenticated Mesh Peering Exchange element:
  - The Selected Pairwise Cipher Suite field shall be set to the first cipher suite selector in the Pairwise Cipher Suite List field in RSNE.
  - The Local Nonce field shall be set to the localNonce value generated by the mesh STA for identifying the current mesh peering instance.
  - The Peer Nonce field shall be set to 0.
  - The GTKdata field shall be present and shall contain the data for the mesh STA's MGTK. The components of the GTKdata are specified in 13.5.4.

The Mesh Peering Open frame shall be protected using AES-SIV as specified in 13.5.3.

13.5.5.2.2 Processing Mesh Peering Open frames for AMPE

On receiving a Mesh Peering Open frame, the mesh STA shall verify the received frame. If AES-SIV returns the symbol "FAIL" the OPN_RJCT event shall be invoked to the corresponding AMPE finite state machine and the reason code "MESH-INVALID-GTK" is generated. Otherwise, processing continues.

The received frame shall be rejected if the security capability selection fails (see 13.5.2). The OPN_RJCT event shall be invoked to the corresponding AMPE finite state machine.

The peer mesh STA's MGTK extracted from the Mesh Peering Open frame shall be added to the Receive MGTK SA in which the peer's MAC address equals the MGTK Source mesh STA MAC address.

If all operations succeed, the mesh STA shall proceed to process the Mesh Peering Open frame on basic parameters as specified in 13.3.6.2.

13.5.5.3 Mesh peering confirm for AMPE

13.5.5.3.1 Generating Mesh Peering Confirm frames for AMPE

In addition to contents for establishing a mesh peering as specified in 13.3.7.1, the Mesh Peering Confirm frame, when used with the AMPE, shall contain the following:

- In the Mesh Peering Management element, the Mesh Peering Protocol Identifier shall be set to 1 "authenticated mesh peering exchange protocol."
- The RSNE shall be the same as sent in the Mesh Peering Open frame.
- In the Authenticated Mesh Peering Exchange element:
  - The Selected Pairwise Cipher Suite field shall be set to the cipher suite selector that indicates the successfully selected pairwise cipher suite (specified in 13.5.2.1).
  - The Peer Nonce field shall be set to the nonce value chosen by the peer mesh STA as received in the Local Nonce field in the Mesh Peering Open frame from the candidate peer mesh STA.
  - The GTKdata field shall not be present.
  - The rest of fields are set to the same values sent in the Mesh Peering Open frame.

The Mesh Peering Confirm frame shall be protected using AES-SIV as specified in 13.5.3.

13.5.5.3.2 Processing Mesh Peering Confirm frames for AMPE

On receiving a Mesh Peering Confirm frame, the mesh STA shall verify the received frame. The received frame shall be discarded if AES-SIV returns the symbol "FAIL."

If AES-SIV returns plaintext, the following operations shall be performed in order:

a) The Selected Pairwise Cipher Suite is checked. If the security capability selection has been done and the received value from Chosen Pairwise Cipher Suite field is not the same as the agreed pairwise cipher suite, the mesh STA shall reject the received frame and the CNF_RJCT event is invoked to the corresponding AMPE finite state machine with the failure reason code MESH-INVALID-SECURITY-CAPABILITY.

b) The Group Cipher Suite is checked. If the received group cipher suite is not supported by the mesh STA, the mesh STA shall reject the received Mesh Peering Confirm frame and the CNF_RJCT event is invoked to the corresponding AMPE finite state machine with the failure reason code MESH-INVALID-SECURITY-CAPABILITY.

If none of the cases is true, the mesh STA shall proceed to process the Mesh Peering Confirm Action frame on basic parameters as specified in 13.3.7.2.

13.5.5.4 Mesh peering close for AMPE

13.5.5.4.1 Generating Mesh Peering Close frames for AMPE

In addition to contents for closing a mesh peering as specified in 13.3.8.1, the Mesh Peering Close frame, when used for the AMPE, shall contain the following:

- In the Mesh Peering Management element, the Mesh Peering Protocol Identifier shall be set to 1 "authenticated mesh peering exchange protocol."
- In the Mesh Peering Management element, the Chosen PMK field shall be set to the same value as sent in the Mesh Peering Open frame.
- In the Authenticated Mesh Peering Exchange element:
  - The Selected Pairwise Cipher Suite field shall be set to the same value as sent in the Mesh Peering Open frame.

    NOTE—If the reason for sending the Mesh Peering Close is the pairwise cipher suite selection failure, the information in this field is used to inform the candidate peer mesh STA what was announced by the mesh STA for the mesh peering instance.

  - The Local Nonce field shall be set to the same value as sent in the Mesh Peering Open frame.
  - The Peer Nonce field shall be set to the same value as received in the Local Nonce field of the Authenticated Mesh Peering Exchange element of the incoming Mesh Peering Management frame from the candidate peer mesh STA.

The Mesh Peering Close frame shall be protected using AES-SIV as specified in 13.5.3.

13.5.5.4.2 Processing Mesh Peering Close frames for AMPE

On receiving a Mesh Peering Close frame, the mesh STA shall verify the received frame. The received frame shall be discarded if AES-SIV returns the symbol "FAIL."

If AES-SIV returns plaintext, the mesh STA shall proceed to process the Mesh Peering Close frame on basic parameters as specified in 13.3.8.2.

13.5.6 AMPE finite state machine

13.5.6.1 Overview

The finite state machine for AMPE supports all the states, events, and actions defined for the finite state machine for the MPM protocol. In addition, new events, actions, and state transitions are added to specify the security functions for AMPE.

When a finite state machine is generated and activated for an AMPE instance, the localNonce shall be generated and used together with a new localLinkID to identify the instance.

13.5.6.2 Additional events and actions to MPM FSM

All events for rejecting or ignoring received Action frames shall report the corresponding reason code related to AMPE functions as described in 13.5.5.

In addition, there is one new event as follows:

- TOR3—This event indicates that the retryTimer has expired, the dot11MeshMaxRetries has been reached, the AMPE is enabled, but the mesh STA failed to confirm the selection of the shared mesh PMKSA. When this event triggers, the protocol instance shall be closed, but no Mesh Peering Close frame shall be sent.

The actions of sending Mesh Peering Management frames are updated as the following:

- sndOPN—Generate a Mesh Peering Open frame for the current AMPE protocol instance (as specified in 13.5.5.2.1) and send it to the candidate peer mesh STA.

- sndCNF—Generate a Mesh Peering Confirm frame for the current AMPE protocol instance (as specified in 13.5.5.3.1) and send it to the candidate peer mesh STA.
- sndClose—Generate a Mesh Peering Close frame for the current AMPE protocol instance (as specified in 13.5.5.4.1) and send it to the candidate peer mesh STA.

13.5.6.3 State transitions

All state transitions specified in MPM FSM shall be used for AMPE finite state machine.

In OPN_SNT state, the following are additional state transitions and actions:

> When TOR3 event occurs, the retryTimer shall be cleared and the holdingTimer shall be set. The finite state machine shall transition to HOLDING state.

In OPN_RCVD state, the following are the additional actions:

> When CNF_ACPT event occurs, in addition to the actions for MPM protocol, the mesh STA shall signal the completion of key management by utilizing the MLME-SETKEYS.request primitive to configure the agreed-upon mesh temporal pairwise key into the IEEE 802.11 MAC and by calling the MLME-SETPROTECTION.request primitive to enable its use.

In CNF_RCVD state, the following are the additional actions:

> When OPN_ACPT event occurs, in addition to the actions for MPM protocol, the mesh STA shall signal the completion of key management by utilizing the MLME-SETKEYS.request primitive to configure the agreed-upon mesh temporal pairwise key into the IEEE 802.11 MAC and received MGTK and by calling the MLME-SETPROTECTION.request primitive to enable the usage.

Table 13-3 and Figure 13-3 specify the state transitions of the finite state machine for AMPE.

Table 13-3—AMPE finite state machine

		To State					
		IDLE	OPN_SNT	CNF_RCVD	OPN_RCVD	ESTAB	HOLDING
From State	IDLE	REQ_RJCT / sndCLS	ACTOPN / (sndOPN, setR)		OPN_ACPT / (sndOPN, sndCNF, setR)		
	OPN_SNT		TOR1 / (sndOPN, setR)	CNF_ACPT / (clR, setC)	OPN_ACPT / (sndCNF)		CLS_ACPT, OPN_RJCT, CNF_RJCT, TOR2, CNCL / (sndCLS, clR, setH) TOR3 / (clR, setH)
	CNF_RCVD					OPN_ACPT / (clC, sndCNF)	CLS_ACPT, OPN_RJCT, CNF_RJCT, CNCL / (sndCLS, clC, setH) TOC / (sndCLS, setH)
	OPN_RCVD				TOR1 / (sndOPN, setR) OPN_ACPT / sndCNF	CNF_ACPT / clR	CLS_ACPT, OPN_RJCT, CNF_RJCT,TOR2, CNCL / (sndCLS, clR, setH)
	ESTAB					OPN_ACPT / sndCNF	CLS_ACPT, OPN_RJCT, CNF_RJCT, CNCL / (sndCLS, setH)
	HOLDING	TOH / —, CLS_ACPT / clH-					OPN_ACPT, CNF_ACPT, OPN_RJCT, CNF_RJCT / sndCLS

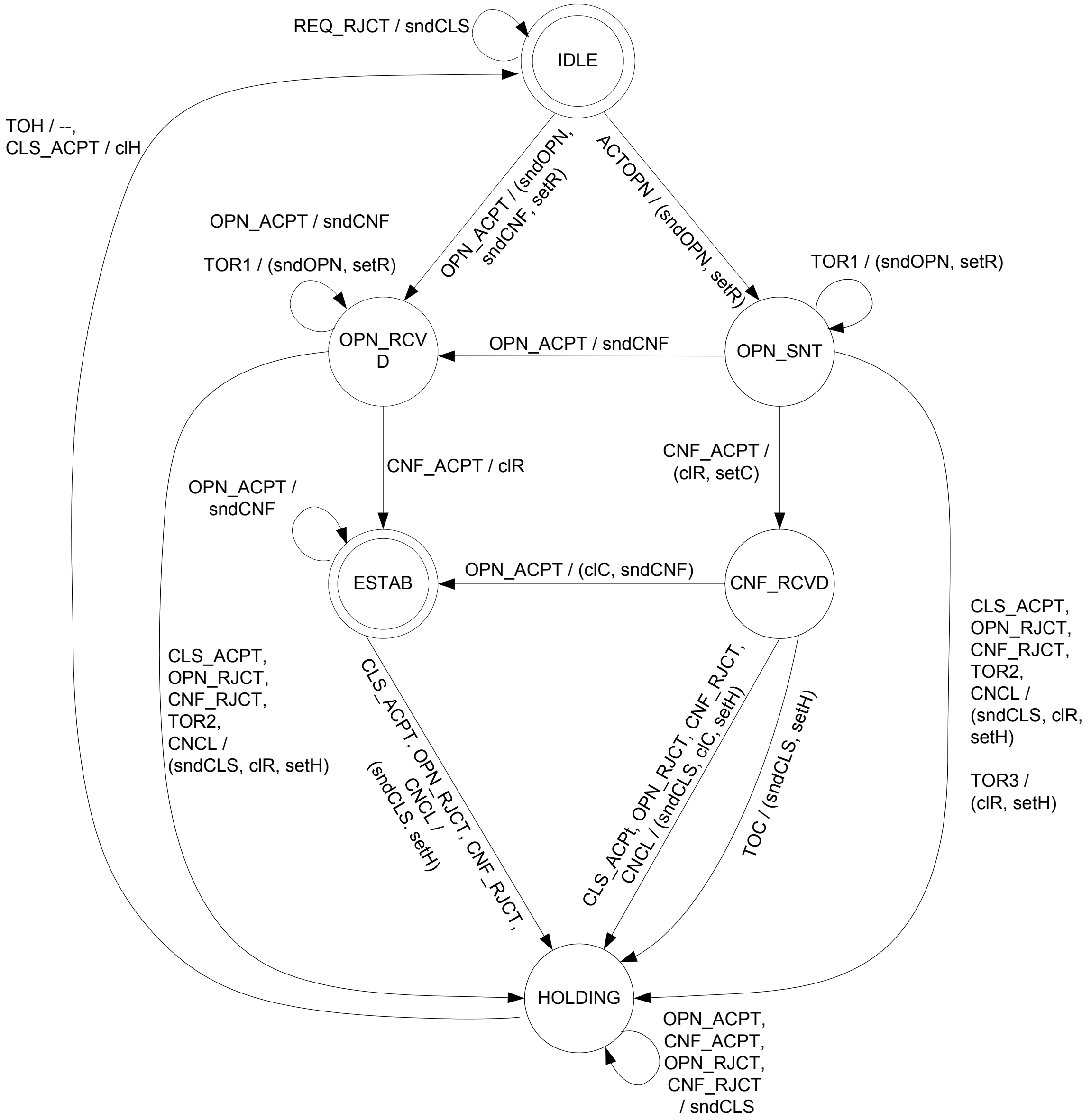

REQ_RJCT / sndCLS
IDLE
TOH / --,
CLS_ACPT / clH
OPN_ACPT / (sndOPN, sndCNF, setR)
ACTOPN / (sndOPN, setR)
OPN_ACPT / sndCNF
TOR1 / (sndOPN, setR)
TOR1 / (sndOPN, setR)
OPN_RCV D
OPN_ACPT / sndCNF
OPN_SNT
CNF_ACPT / clR
CNF_ACPT / (clR, setC)
OPN_ACPT / sndCNF
ESTAB
OPN_ACPT / (clC, sndCNF)
CNF_RCVD
CLS_ACPT, OPN_RJCT, CNF_RJCT, TOR2, CNCL / (sndCLS, clR, setH)
CLS_ACPT, OPN_RJCT, CNF_RJCT, CNCL / (sndCLS, setH)
CLS_ACPt, OPN_RJCT, CNF_RJCT, CNCL / (sndCLS, clC, setH)
TOC / (sndCLS, setH)
CLS_ACPT, OPN_RJCT, CNF_RJCT, TOR2, CNCL / (sndCLS, clR, setH)
TOR3 / (clR, setH)
HOLDING
OPN_ACPT, CNF_ACPT, OPN_RJCT, CNF_RJCT / sndCLS

Figure 13-3—Finite state machine of the AMPE protocol

13.5.7 Keys and key derivation algorithm for the authenticated mesh peering exchange (AMPE)

To execute the AMPE and mesh group key handshake with a candidate peer mesh STA, the mesh STA shall derive an authenticated encryption key (AEK) and a mesh temporal key (MTK) using the PMK it shares with the candidate peer mesh STA.

The AEK is derived statically from the shared PMK. The MTK is derived from the shared PMK and dynamic information provided by the mesh STA and candidate peer mesh STA.

The AEK is mutually derived by the local mesh STA and the peer mesh STA once a new PMK has been selected. The AEK shall be derived from the PMK by

AEK ← KDF-256(PMK, “AEK Derivation”, Selected AKM Suite ||
min(localMAC, peerMAC) || max(localMAC, peerMAC)).

The temporal key (MTK) shall be derived from the PMK by

MTK ← KDF-X(PMK, “Temporal Key Derivation”, min(localNonce, peerNonce) ||
max(localNonce, peerNonce) || min(localLinkID, peerLinkID) ||
max(localLinkID, peerLinkID) || Selected AKM Suite ||
min(localMAC, peerMAC) || max(localMAC, peerMAC)).

CCMP uses X = 128. The “min” and “max” operations for IEEE 802 addresses are with the address converted to a positive integer, treating the first transmitted octet as the most significant octet of the integer as specified in 11.6.1.3. The min and max operations for nonces are with the nonces treated as positive integers converted as specified in 8.2.2.

The MTK is used to protect communications between two peer mesh STAs. The local mesh STA and peer mesh STA derive an MTK per peering instance and may rekey the MTK using AMPE.

13.6 Mesh group key handshake

13.6.1 General

The mesh group key handshake may be used by either mesh STA, after a secure mesh peering has been established, to update the MGTK that it uses to protect group addressed MPDUs that it transmits to its peer mesh STAs.

The mesh STA may update its MGTK when a mesh peering is terminated.

To update the MGTK, the mesh STA shall execute the mesh group key handshake with each of its current peer mesh STAs. The “MGTK source” is the mesh STA that is sending the MGTK to a peer mesh STA using this protocol. A “MGTK recipient” is a mesh STA receiving the MGTK being sent by the MGTK Source.

The mesh group key handshake exchange shall include the following two messages:

— Message 1: Mesh Group Key Inform frame
— Message 2: Mesh Group Key Acknowledge frame

Mesh Group Key Inform frame and Mesh Group Key Acknowledge frame are conventionally referred to as “mesh group key handshake frames.”

The mesh STA shall do an AMPE handshake before a mesh group key handshake if both are required to be done.

NOTE—It is impossible that the MGTK source initiates the mesh group key handshake before the AMPE completes successfully.

13.6.2 Protection on mesh group key handshake frames

Mesh group key handshake frames used in mesh group key handshake are protected using the deterministic authenticated encryption mode of AES-SIV (RFC 5297) when dot11MeshSecurityActivated is true.

When constructing protection on mesh group handshake frames, the following procedure shall be used:

- The key shall be the AEK from the current active security association with the peer mesh STA that receives the mesh group key handshake frame.
- The input plaintext shall be the AMPE Authenticated Mesh Peering element (see 8.5.16.5 and 8.5.16.6).
- The plaintext shall be the Authenticated Mesh Peering Exchange element.
- AAD shall be three distinct components as follows:
  1) The localMAC
  2) The peerMAC
  3) The contents of the mesh group key handshake frame from the category (inclusive) to the MIC element (exclusive)
- The synthetic initialization vector produced by AES-SIV shall be copied into the MIC field of the MIC element in the frame.
- The produced cipher text shall become the remainder of the mesh group key handshake frame after the MIC element.

When verifying the protection on the mesh group handshake frames, the following procedure shall be used:

- The key shall be the AEK from the current active security association with the peer mesh STA that receives the mesh group key handshake frame.
- AAD shall be three distinct components as follows:
  1) The peerMAC
  2) The localMAC
  3) The contents of the mesh group key handshake frame from the category (inclusive) to the MIC element (exclusive)
- The synthetic initialization vector shall be the MIC field of the MIC element in the frame.
- The cipher text shall be the content after the MIC element in the frame.
- If AES-SIV validation function takes above input.
  - If the function returns the special symbol “FAIL,” the frame shall be discarded.
  - If the plaintext is returned successfully, the produced plaintext shall be treated as the contents after the MIC element in the frame.

13.6.3 Mesh Group Key Inform frame construction and processing

Mesh Group Key Inform frame shall be constructed as follows:

- The Authenticated Mesh Peering Exchange element shall be set as the following:
  - The Selected Pairwise Cipher Suite field shall be left blank.
  - The Local Nonce field shall be set to the same value as sent in the Mesh Peering Open frame that established the mesh peering instance.
  - The Peer Nonce field shall be set to the same value as received in the Local Nonce field of the Authenticated Mesh Peering Exchange element of the incoming Mesh Peering Open frame that established the peering instance.
  - The Key Replay Counter field shall be set to the mesh STA’s local replay counter value, incremented by 1, for the mesh peering. After setting this field, the local replay counter shall also be incremented by 1.
  - The GTKdata field shall be present and shall contain the data for the MGTK from MGTK source. The components of the GTKdata are specified in 13.5.4.
- The MIC element shall be set according to the protection mechanism in 13.6.2.

The construction of AES-SIV protection on Mesh Group Key Inform frame shall use the construction procedure as in 13.6.2.

The MGTK source sends the Mesh Group Key Inform frame to the MGTK recipient.

On reception of Mesh Group Key Inform frame, the MGTK recipient shall use the verification procedure in 13.6.2 to validate the AES-SIV construction.

- If the validation recovers the plaintext successfully, the MGTK recipient shall proceed with the following procedure:
  - Verify that values in the Local Nonce field and the Peer Nonce field in the Authenticated Mesh Peering Exchange element are the same as in the current valid mesh TKSA that the MGTK recipient established with the sender of the Mesh Group Key Inform frame. If there is any mismatch, the received Mesh Group Key Inform frame shall be discarded and no further action shall be taken.
  - Verify that the Key Replay Counter has not yet been seen before, i.e., its value is strictly larger than that in any other mesh Group Key Inform frame received thus far during this security association. If this verification fails, the received Mesh Group Key Inform frame shall be discarded and no further action shall be taken.
  - Use the MLME-SETKEYS.request primitive to configure the temporal MGTK into its IEEE 802.11 MAC.
  - Respond by constructing and sending mesh group key handshake acknowledge to the MGTK source and incrementing the replay counter.

  NOTE—The MGTK source increments and uses a new Key Replay Counter field value on every Mesh Group Key Inform frame, even retries, because the Mesh Group Key Acknowledge responding to an earlier Mesh Group Key Inform frame might have been lost. If the MGTK source did not increment the replay counter, the MGTK receiver discards the retry, and no responding Mesh Group Key Acknowledge frame will ever arrive.

- If the AES-SIV validation returns a special symbol “FAIL,” the Mesh Group Key Inform frame shall be discarded. No further action shall be taken.

13.6.4 Mesh Group Key Acknowledge frame construction and processing

Mesh Group Key Acknowledge frame shall be constructed as follows:

- The Authenticated Mesh Peering Exchange element shall be set as follows:
  - The Selected Pairwise Cipher Suite field shall be left blank.
  - The Local Nonce field shall be set to the same value as sent in the Mesh Peering Open frame that established the mesh peering instance.
  - The Peer Nonce field shall be set to the same value as received in the Local Nonce field of the Authenticated Mesh Peering Exchange element of the incoming Mesh Peering Open frame that established the peering instance.
  - The Key Replay Counter shall be set to the same value as received in the Mesh Group Key Inform frame.
  - The GTKdata field shall be blank.
- The MIC element shall be set according to the protection mechanism in 13.6.2.

The construction of AES-SIV protection on Mesh Group Key Acknowledge frame shall use the construction procedure as in 13.6.2.

The MGTK recipient sends the Mesh Group Key Acknowledge frame to the MGTK source.

On reception of Mesh Group Key Acknowledge frame, the MGTK source shall use the verification procedure in 13.6.2 to validate the AES-SIV construction.

- If the validation recovers the plaintext successfully, the MGTK source shall set the content of the Authenticated Mesh Peering Exchange element using the recovered plaintext and proceed with the following procedure:
  - Verify that values in the Local Nonce field and the Peer Nonce field in the Authenticated Mesh Peering Exchange element are the same as in the current valid mesh TKSA that the MGTK source established with the sender of the Mesh Group Key Acknowledge frame. If there is any mismatch, the received Mesh Group Key Acknowledge frame shall be discarded and no further action shall be taken.
  - Verify that the Key Replay Counter value matches the one that it has used for the mesh group key handshake. If this verification fails, the received Mesh Group Key Acknowledge frame shall be discarded and the MGTK source may invoke a retry to send a new Mesh Group Key Inform frame with a new Key Replay Counter value.
- If the validation returns a special symbol "FAIL," the Mesh Group Key Acknowledge frame shall be discarded and the MGTK source may invoke a retry to send a new Mesh Group Key Inform frame with a new Key Replay Counter value.

13.6.5 Mesh group key implementation considerations

If the MGTK source does not receive a Mesh Group Key Acknowledge frame to its Mesh Group Key Inform frames, it shall attempt dot11MeshConfigGroupUpdateCount additional transmissions of the Mesh Group Key Inform frame. The retransmit timeout value shall be 100 ms for the first timeout, half the listen interval for the second timeout, and the listen interval for subsequent timeouts. If there is no listen interval or the listen interval is zero, then 100 ms shall be used for all timeout values. If it still has not received a response after this, then the MGTK source shall tear down the mesh peering and mesh TKSA with this MGTK recipient, by generating a CNCL event for the peering instance, and pass the event to the mesh peering instance controller.

13.7 Mesh security

During the AMPE, the peers negotiate, and agree upon, a pairwise ciphersuite and a group cipher suite. They also establish a Mesh TKSA and Mesh GTKSA to be used with the pairwise cipher suite and group cipher suite, respectively.

When dot11MeshSecurityActivated is true, all Mesh Data frames and individually addressed management frames (excluding Authentication frames and self-protected management frames) shall be protected by the Mesh TKSA, and all group addressed data frames and group addressed management frames that are indicated as "Group Addressed Privacy" in Table 8-38 shall be protected by the Mesh GTKSA.

13.8 Mesh path selection and metric framework

13.8.1 General

The term *mesh path selection* is used to describe selection of multi-hop paths between mesh STAs at the link layer. Mesh path selection creates forwarding information that is utilized for MSDU/MMPDU forwarding as described in 9.32.

13.8.2 Extensible path selection framework

This standard allows for alternative and flexible implementations of path selection protocols and metrics.

A mesh STA may include multiple protocol implementations (that is, the default protocol, vendor-specific protocols, etc.) as well as multiple metric implementations, but only one path selection protocol and only one path selection metric shall be used by a mesh STA at a time.

As described in 13.2.3 and 13.2.7, mesh STAs use the Mesh Configuration element (8.4.2.100) to announce the active path selection protocol and active path selection metric of the MBSS. This allows a neighbor mesh STA to identify if it should become a member of the MBSS and how it should establish mesh peerings with its members. This standard does not force an existing MBSS that is using a protocol other than the default protocol to switch to the default protocol when a new mesh STA requests mesh peering establishment. While it is possible, in principle, to implement such behavior, an algorithm to coordinate such reconfiguration is beyond the scope of this standard.

Path selection protocol and path selection metric are identified by a unique identifier as defined in 8.4.2.100.2 and 8.4.2.100.3, respectively. Also, each path selection protocol and each path selection metric specifies the following:

- Data type of metric values
- Length of the metric field
- Operator for aggregation of link metrics to a path metric; the symbol $\oplus$ is used to identify an arbitrary operator for aggregation
- Comparison operator for determining a better or worse path; how this is performed depends on the actual comparison operator
- Initial value of the path metric (path selection metric only)

The standard defines a default mandatory path selection protocol (HWMP, 13.10) and a default mandatory path selection metric (airtime link metric, 13.9). Both shall be implemented on all mesh STAs to ensure interoperability.

13.8.3 Link metric reporting

A mesh STA may submit a link metric report to or request a link metric report from its neighbor peer mesh STA by transmitting a Mesh Link Metric Report frame. A mesh STA receiving a Mesh Link Metric Report element with the Request subfield of the Flags field equal to 1 shall reply with a Mesh Link Metric Report frame containing the link metric value for the corresponding link.

Upon reception of a Mesh Link Metric Report frame, the mesh STA may update its local link metric information using the link metric information received. The procedure to update the local link metric information with the link metric information received from a neighbor peer mesh STA is outside the scope of the standard.

13.9 Airtime link metric

This subclause defines a default link metric that may be used by a path selection protocol to identify an efficient radio-aware path. The extensibility framework allows this metric to be overridden by any path selection metric as specified in the mesh profile.

Airtime reflects the amount of channel resources consumed by transmitting the frame over a particular link. This measure is approximate and designed for ease of implementation and interoperability.

The airtime for each link is calculated as follows:

$$c_a = \left[O + \frac{B_t}{r}\right]\frac{1}{1 - e_f}$$

where

O and B_t are constants listed in Table 13-4

input parameter *r* is the data rate (in Mb/s)
input parameter e_f is the frame error rate for the test frame size B_t
rate *r* represents the data rate at which the mesh STA would transmit a frame of standard size B_t based on current conditions, and its estimation is dependent on local implementation of rate adaptation
frame error rate e_f is the probability that when a frame of standard size B_t is transmitted at the current transmission bit rate *r*, the frame is corrupted due to transmission error; its estimation is a local implementation choice. Frame failures due to exceeding Mesh TTL should not be included in this estimate as they are not correlated with link performance.

The airtime link metric shall be encoded as an unsigned integer in units of 0.01 TU.

Table 13-4—Airtime cost constants

Parameter	Recommended value	Description
O	Varies depending on PHY	Channel access overhead, which includes frame headers, training sequences, access protocol frames, etc.
B_t	8192	Number of bits in test frame

Table 13-5 gives the parameters of the airtime link metric for the extensible path selection framework.

Table 13-5—Parameters of the airtime link metric for extensible path selection framework

Parameter	Notes
Path Selection Metric ID	See Table 8-178 in 8.4.2.100.3
Data type	Unsigned integer, 0 ≤ metric value< 4 294 967 296
Length of metric field	4 octets
Operator for metric aggregation	addition (+)
Comparison operator	*less than, equal to, greater than* as used with integers — metric *a* is *better than* metric *b* iff $a < b$ — metric *a* is *equal to* metric *b* iff $a = b$ — metric *a* is *worse than* metric *b* iff $a > b$
Initial value of path metric	0

An example of the airtime link metric is shown in W.5.

13.10 Hybrid wireless mesh protocol (HWMP)

13.10.1 General

The hybrid wireless mesh protocol (HWMP) is a mesh path selection protocol that combines the flexibility of on-demand path selection with proactive topology tree extensions. The combination of reactive and proactive elements of HWMP enables efficient path selection in a wide variety of mesh networks (with or without access to the infrastructure).

HWMP uses a common set of protocol elements, generation and processing rules inspired by Ad Hoc On-Demand Distance Vector (AODV) protocol (IETF RFC 3561 [B34]) adapted for MAC address-based path selection and link metric awareness. HWMP is completely specified herein and does not require reference to AODV specifications or descriptions.

HWMP supports two modes of operation depending on the configuration. These modes provide different levels of functionality as follows:

— On-demand mode: The functionality of this mode is always available, independent of whether a root mesh STA is configured in the MBSS or not. It allows mesh STAs to communicate using peer-to-peer paths.

— Proactive tree building mode: In this mode, additional proactive tree building functionality is added to the on-demand mode. This can be performed by configuring a mesh STA as root mesh STA using either the proactive PREQ or RANN mechanism.The proactive PREQ mechanism creates paths from the mesh STAs to the root, using only group-addressed communication. The RANN mechanism creates paths between the root and each mesh STA using acknowledged communication.

These modes are not exclusive. On-demand and proactive modes are used concurrently, because the proactive modes are extensions of the on-demand mode.

NOTE—One example of concurrent usage of on-demand and proactive mode is for two mesh STAs that are part of the same mesh BSS (or STAs that are proxied by mesh STAs in the same MBSS) to begin communicating using the proactively built tree but subsequently to perform an on-demand discovery for a direct path. This type of concurrent usage of the proactive and on-demand modes allows communication to begin immediately (by forwarding all traffic to the root, which knows all mesh STAs and addresses proxied by mesh STAs in the MBSS) while an on-demand discovery finds a shorter path between two mesh STAs (or STAs that are proxied by mesh STAs in the same MBSS).

All HWMP modes of operation utilize common processing rules and primitives. HWMP elements are the PREQ (path request), PREP (path reply), PERR (path error), and RANN (root announcement). The metric cost of the links determines which paths HWMP builds. In order to propagate the metric information between mesh STAs, a Metric field is used in the PREQ, PREP, and RANN elements.

Path selection in HWMP uses a sequence number mechanism to ensure that mesh STAs can distinguish current path information from stale path information at all times in order to maintain loop-free connectivity. Each mesh STA maintains its own HWMP SN, which is propagated to other mesh STAs in the HWMP elements. Rules for maintaining HWMP SNs are given in 13.10.8.3.

13.10.2 Terminology

This subclause describes terminology for HWMP, especially for the process of path discovery. Terms such as Path Originator or Path Target designate very specific entities within the path discovery process. They stay with the same assigned entity for the whole path discovery process and other procedures related to this path discovery. Figure 13-4 illustrates an example utilizing this terminology.

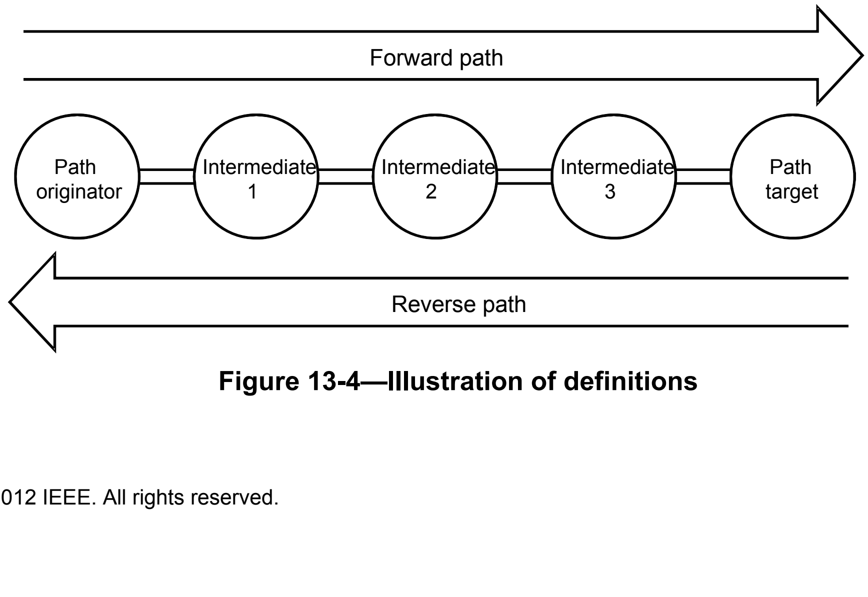


Figure 13-4—Illustration of definitions

NOTE—Both the path target and path originator are a path destination for the forward path and the reverse path respectively.

The following terms are used within the context of a single PREQ/PREP pair, a so-called HWMP path discovery:

- **path originator:** The path originator is the mesh STA that triggers the path discovery.
- **path originator address:** The MAC address of the path originator.
- **path target:** The path target is the entity to which the path originator attempts to establish a path.

  NOTE—When an originator mesh STA initially attempts to establish a path to a target, it does not know whether the target is a mesh STA in the mesh BSS or not. Only when the Originator receives a PREP does it learn if the target is a mesh STA in the mesh BSS or not. If the target is in the mesh BSS, it is referred to as a target mesh STA. If the target is outside the mesh BSS, the term target proxy mesh gate refers to the mesh gate proxying for the target.

- **path target address:** The MAC address of the path target.
- **intermediate mesh STA:** The intermediate mesh STA is the mesh STA that participates in path selection and is neither path originator nor path target.
- **intermediate mesh STA address:** The MAC address of the intermediate mesh STA.
- **forward path:** The forward path is the mesh path to the path target, set up at the path originator and intermediate mesh STAs.
- **reverse path:** The reverse path is the mesh path to the path originator, set up at the path target and intermediate mesh STAs.
- **HWMP sequence number (HWMP SN):** Each mesh HWMP path selection element contains an HWMP SN that allows recipients to distinguish newer from stale information. An HWMP SN is specific to a mesh STA. See also 13.10.8.3.
- **forwarding information:** The forwarding information maintained by an originator mesh STA, an intermediate mesh STA, or a target mesh STA that allows the mesh STA to perform its path selection and forwarding functions.

  The terminology used when discussing forwarding information is relative to the mesh STA (reference mesh STA, given mesh STA or local mesh STA) and a particular mesh destination of the path. The following terms are specific to a given instance of the forwarding information:
  - **destination mesh STA:** The end station (mesh STA) of a (forward or reverse) path.
  - **destination mesh STA address:** The MAC address of the destination mesh STA.
  - **destination HWMP SN:** The HWMP SN of the destination mesh STA.
  - **next-hop mesh STA:** The next-hop mesh STA is the next peer mesh STA on the mesh path to the destination mesh STA.
  - **next-hop mesh STA address:** The MAC address of the next-hop mesh STA.
  - **precursor mesh STA:** A precursor mesh STA is a neighbor peer mesh STA on the mesh path that identifies a given mesh STA as the next-hop mesh STA to the destination mesh STA.
  - **precursor mesh STA address:** The MAC address of the precursor mesh STA.
  - **lifetime:** The time during which forwarding information remains active (see 13.10.8.4)
- **unreachable destination:** A destination mesh STA is considered unreachable by a source mesh STA or an intermediate mesh STA if the link to the next hop of the mesh path to this destination mesh STA, as derived from its forwarding information, is no longer usable.
- **element time to live (Element TTL)**: An integer number that is used to limit the number of hops an HWMP element may be processed and propagated. Note that this Element TTL is different from the Mesh TTL in the Mesh Control field (see 8.2.4.7.3).
- **root mesh STA:** A root mesh STA is configured to originate pro-active PREQs or RANNs. It is the root of a path selection tree.

Table 13-6 and Table 13-7 shows the roles of the various mesh STAs in the forward path and reverse path generated as a result of the full PREQ and PREP processing as shown in Figure 13-4. Each row in the table contains the roles of a forward/reverse path from the reference mesh STA's perspective.

Table 13-6—Precursor and next hop examples (forward path)

Forward path (to Path Target)			
Reference mesh STA	**Precursor mesh STA**	**Next-hop mesh STA**	**Destination mesh STA**
Path Originator	N/A	Intermediate 1	Path Target
Intermediate 2	Intermediate 1	Intermediate 3	Path Target
Path Target	Intermediate 3	N/A	Path Target

Table 13-7—Precursor and next hop examples (reverse path)

Reverse path (to Path Originator)			
Reference mesh STA	**Precursor mesh STA**	**Next-hop mesh STA**	**Destination mesh STA**
Path Originator	Intermediate 1	N/A	Path Originator
Intermediate 2	Intermediate 3	Intermediate 1	Path Originator
Path Target	N/A	Intermediate 3	Path Originator

13.10.3 On-demand path selection mode

If a source mesh STA needs to find a path to a destination mesh STA using the on-demand path selection mode, it broadcasts a PREQ with the path target specified in the list of targets and the metric field initialized to the initial value of the active path selection metric.

When a mesh STA receives a new PREQ, it creates or updates its path information to the originator mesh STA and propagates the PREQ to its neighbor peer mesh STAs if the PREQ contains a greater HWMP SN, or the HWMP SN is the same as the current path and the PREQ offers a better metric than the current path. Each mesh STA may receive multiple copies of the same PREQ that originated at the originator mesh STA, each PREQ traversing a unique path.

Whenever a mesh STA propagates a PREQ, the metric field in the PREQ is updated to reflect the cumulative metric of the path to the originator mesh STA. After creating or updating a path to the originator mesh STA, the target mesh STA sends an individually addressed PREP back to the originator mesh STA.

If the mesh STA that received a PREQ is the target mesh STA, it sends an individually addressed PREP back to the originator mesh STA after creating or updating a path to the originator mesh STA.

The PREQ provides the TO (Target Only) subfield that allows path selection to take advantage of existing paths to the target mesh STA by allowing an intermediate mesh STA to return a PREP to the originator mesh STA. If the TO (Target Only) subfield is 1, only the target mesh STA responds with a PREP. The effect of setting the TO (Target Only) subfield to 0 is the quick establishment of a path using the PREP generated by an intermediate mesh STA, allowing the forwarding of MSDUs with a low path selection delay. In order to select (or validate) the best path during the path selection procedure, the intermediate mesh STA that responded with a PREP propagates the PREQ with the TO (Target Only) subfield set to 1. This prevents all other intermediate mesh STAs on the way to the target from sending a PREP.

Intermediate mesh STAs create a path to the target mesh STA on receiving the PREP, and also forward the PREP toward the originator. When the originator receives the PREP, it creates a path to the target mesh STA. If the target mesh STA receives further PREQs with a better metric, then the target updates its path to the originator with the new path and also sends a new PREP to the originator along the updated path. A bidirectional, best metric end-to-end path is established between the originator and target mesh STA.

13.10.4 Proactive tree building mode

13.10.4.1 General

There are two mechanisms for proactively disseminating path selection information for reaching the root mesh STA. The first method uses a *proactive* PREQ element and is intended to create paths between all mesh STAs and the root mesh STA in the network proactively. The second method uses a RANN element and is intended to distribute path information for reaching the root mesh STA but there is no forwarding information created.

A mesh STA configured as root mesh STA sends either proactive PREQ or RANN elements periodically.

13.10.4.2 Proactive PREQ mechanism

The PREQ tree building process begins with a proactive PREQ element sent by the root mesh STA, with the Target Address set to all ones and the TO subfield set to 1. The PREQ contains the path metric (set to the initial value of the active path selection metric by the root mesh STA) and an HWMP SN. The proactive PREQ is sent periodically by the root mesh STA, with increasing HWMP SNs.

A mesh STA receiving a proactive PREQ creates or updates its forwarding information to the root mesh STA, updates the metric and hop count of the PREQ, records the metric and hop count to the root mesh STA, and then transmits the updated PREQ. Information about the presence of and distance to available root mesh STA(s) is disseminated to all mesh STAs in the network.

Each mesh STA may receive multiple copies of a proactive PREQ, each traversing a unique path from the root mesh STA to the mesh STA. A mesh STA updates its current path to the root mesh STA if and only if the PREQ contains a greater HWMP SN, or the HWMP SN is the same as the current path and the PREQ offers a better metric than the current path to the root mesh STA. The processing of the proactive PREQ is the same as the processing of the PREQ in the on-demand mode described in 13.10.3.

If the proactive PREQ is sent with the Proactive PREP subfield set to 0, the recipient mesh STA may send a proactive PREP. A proactive PREP is necessary, for example, if the mesh STA has data to send to the root mesh STA, thus requiring the establishment of a forward path from the root mesh STA. During the time the forward path is required, the recipient mesh STA shall send a proactive PREP even if the Proactive PREP subfield is set to 0. Guidance on controlling the generation of proactive PREQs in such a case is given in W.6.

If the PREQ is sent with a Proactive PREP subfield set to 1, the recipient mesh STA shall send a proactive PREP. The proactive PREP establishes the path from the root mesh STA to the mesh STA.

13.10.4.3 Proactive RANN mechanism

The root mesh STA periodically propagates a RANN element into the network. The information contained in the RANN is used to disseminate path metrics to the root mesh STA, but reception of a RANN does not establish a path.

Upon reception of a RANN, each mesh STA that has to create or refresh a path to the root mesh STA sends an individually addressed PREQ to the root mesh STA via the mesh STA from which it received the RANN.

The root mesh STA sends a PREP in response to each PREQ. The individually addressed PREQ creates the reverse path from the root mesh STA to the originator mesh STA, while the PREP creates the forward path from the mesh STA to the root mesh STA.

13.10.5 Collocated STAs

HWMP terminology strictly refers to a STA whose address is used for destination mapping (i.e., the originator address, the intermediate mesh STA address, or the target address). HWMP terminology does not make any assumption about the address used for communication over the WM (i.e., the Transmitter Address and the Receiver Address). For example, there is no requirement that the Path Originator or the Path Target of an HWMP path are the same as the address used for transmitting the first and receiving the last (respectively) HWMP Mesh Path Selection frame containing a PREQ.

The corollary to this is that the first hop of a PREQ may have a transmitter address that is not the same as the Originator address in the PREQ element and that the first hop may have a transmitter address that is not the same as the source address. In order to determine whether a transmission is a first hop or not, mesh STAs should not compare the source and transmitter addresses. Instead, this determination can be made by looking at the hop count field of the PREQ element (which is not used as an acceptance criterion).

13.10.6 Parameters for extensible path selection framework

Table 13-8 gives the parameters of HWMP for the extensible path selection framework (see 13.8.2).

Table 13-8—Parameters of HWMP for extensible path selection framework

Parameter	Notes
Path Selection Protocol ID	See Table 8-177 in 8.4.2.100.2
Data type of metric field	As defined by active path selection metric
Length of metric field	4 octets
Operator for metric aggregation	As defined by active path selection metric
Comparison operator	As defined by active path selection metric
Initial value of path metric	As defined by active path selection metric

13.10.7 Addressing of HWMP Mesh Path Selection frame

All HWMP elements are sent in an HWMP Mesh Path Selection frame (see 8.5.17.3). The RANN element may also be sent in a Beacon frame. “Cases” refer to the different conditions that trigger the transmission of an HWMP Mesh Path Selection frame. PREQ cases are specified in 13.10.9.3. PREP cases are specified in 13.10.10.3. PERR cases are specified in 13.10.11.3. RANN cases are specified in 13.10.12.3.

Note that the PREQ Addressing Mode subfield in the Flags field identifies the propagation mode of the PREQ; an Addressing Mode subfield of 0 indicates that the PREQ is group addressed, an Addressing Mode subfield of 1 indicates that the PREQ is individually addressed.

The addresses of the HWMP Mesh Path Selection frame shall be as follows:

- PREQ group addressed—Addressing Mode subfield = 0 [Case A: Path Discovery (Original transmission), Case B: Path Maintenance (Original transmission), Case C: Proactive PREQ (Original transmission), and Case E: PREQ Propagation]:
  - Address 1: Group address
  - Address 2: Address of the mesh STA sending the PREQ
  - Address 3: Same as Address 2
- PREQ individually addressed—Addressing Mode subfield = 1 [Case D: Root Path Confirmation (Original transmission)]:
  - Address 1: Address 2 of the frame containing the RANN element that triggered the PREQ
  - Address 2: Address of the mesh STA sending the PREQ
  - Address 3: Same as Address 2
- PREQ individually addressed—Addressing Mode subfield = 1 [Case E: PREQ Propagation]:
  - Address 1: Next-hop MAC address to the mesh STA identified as the Target MAC address in the PREQ element
  - Address 2: Address of the mesh STA sending the PREQ
  - Address 3: Same as Address 2
- PREP Case A: Original transmission, Case C: Intermediate reply, Case D: Proactive PREP in Proactive PREQ mode:
  - Address 1: Address of the next hop to the Originator Mesh STA Address in the PREQ that triggered the PREP
  - Address 2: Address of the mesh STA sending the PREP
  - Address 3: Same as Address 2
- PREP Case B: PREP Propagation:
  - Address 1: Address of the next hop to the Originator Mesh STA Address in the PREP that triggered the PREP
  - Address 2: Address of the mesh STA sending the PREP
  - Address 3: Same as Address 2
- PERR individually addressed [Case A: Original transmission—next hop is unusable]:
  - Address 1: Address of each one of the precursors for which the active forwarding information has been invalidated [see Case A, 13.10.11.3]
  - Address 2: Address of the mesh STA sending the PERR
  - Address 3: Same as Address 2
- PERR individually addressed [Case B: Original transmission—missing forwarding information]:
  - Address 1: Address of the transmitter of the frame that triggered the PERR (see Case B, 13.10.11.3)
  - Address 2: Address of the mesh STA sending the PERR
  - Address 3: Same as Address 2
- PERR individually addressed [Case C: Original transmission (proxy information is unusable)]:
  - Address 1: Address of each one of the neighbor peer mesh STAs
  - Address 2: Address of the mesh STA sending the PERR
  - Address 3: Same as Address 2
- PERR individually addressed [Case D: PERR propagation]:
  - Address 1: Address of each one of the precursors for which the active forwarding information has been invalidated (see 13.10.11.4.3)
  - Address 2: Address of the mesh STA sending the PERR
  - Address 3: Same as Address 2
- PERR group addressed [all cases]:
  - Address 1: Group address
  - Address 2: Address of the mesh STA sending the PERR

  - Address 3: Same as Address 2
- RANN all cases:
  - Address 1: Group address
  - Address 2: Address of the mesh STA sending the RANN
  - Address 3: Same as Address 2

Multiple HWMP elements may be sent in the same HWMP Mesh Path Selection frame if they share the same intended Address 1.

13.10.8 General rules for processing HWMP elements

13.10.8.1 General

This subclause describes the rules for the processing of the following components of the HWMP elements:

- HWMP SN
- Element TTL
- Metric

13.10.8.2 HWMP propagation

The term *propagate* is used to describe the means by which elements are not transmitted "as is" across the network but are processed and modified along the way. Many HWMP elements are intended to be processed and propagated across an MBSS by mesh STAs. Each propagation is subject to certain rules or limitations as explained in the following subclauses. Certain parameters in the HWMP elements are updated during the propagation. See 13.10.9, 13.10.10, 13.10.11, and 13.10.12.

The originator of an HWMP element sets the initial value of the Element TTL. The mesh STA that receives the HWMP element shall propagate it if the received value of Element TTL is greater than 1. Before propagating the HWMP element, the mesh STA decrements the Element TTL value.

In general, the propagation of an HWMP element is not subject to a delay. Exception exists for the RANN element as described in 13.10.12.

13.10.8.3 HWMP sequence numbering

HWMP uses sequence numbers to prevent the creation of path loops and to distinguish stale and fresh path information. Each mesh STA keeps its own HWMP SN that it increments, uses in HWMP elements, and processes according to the HWMP rules. HWMP SNs for other mesh STAs are maintained in the forwarding information (see 13.10.8.4).

An HWMP SN is included in the PREQ, PREP, PERR, and RANN elements. The HWMP SN in the forwarding information is updated whenever a mesh STA receives new (i.e., not stale) information about the HWMP SN from a PREQ, PREP, or PERR that may be received relative to that originator mesh STA, target mesh STA, or destination mesh STA.

HWMP depends on each mesh STA in the network to own and maintain its HWMP SN to guarantee the loop-freedom of all paths towards that mesh STA. A mesh STA increments its own HWMP SN in the following two circumstances:

- If it is an originator mesh STA, it shall increment its own HWMP SN immediately before it starts a path discovery. This prevents conflicts with previously established reverse paths towards the originator mesh STA. However, it might be advantageous not to increment the HWMP SN too frequently. An optional mechanism for achieving this is described in 13.10.8.6.

- If it is a target mesh STA, it shall update its own HWMP SN to maximum (current HWMP SN, target HWMP SN in the PREQ) + 1 immediately before it generates a PREP in response to a PREQ. The target HWMP SN of the PREQ is relevant when a link was broken along the path and the stored sequence number was increased at an intermediate mesh STA.

HWMP SNs are processed as follows:

a) HWMP SNs are incremented monotonically as unsigned integers.
b) Comparing HWMP SNs is done using a circular modulo 2^{32} comparison.

In general, when a mesh STA receives an element with an HWMP SN that is less than the HWMP SN in the corresponding forwarding information, it discards the received element. If they are the same, the outcome (element processed or not) depends on the type of the element and some additional conditions. These cases are noted in the applicable element descriptions.

The only circumstance in which a mesh STA may change the HWMP SN of another mesh STA in the forwarding information independently of the reception of an HWMP element originated by this mesh STA is in response to a broken or no longer usable link to the next hop towards that destination mesh STA. The mesh STA determines which destinations use a particular next hop by consulting its forwarding information. In this case, for each destination that uses the next hop, the mesh STA increments the HWMP SN in the forwarding information and marks the path as invalid (see also 13.10.11). Whenever any forwarding information containing an HWMP SN greater than the recorded HWMP SN for an affected destination is received by a mesh STA that has marked that recorded forwarding information as invalid, the mesh STA shall update its forwarding information according to the information contained in the update.

13.10.8.4 Forwarding information

In addition to the parameters contained in the basic forwarding information as described in 9.32.2, the forwarding information to a destination defined by HWMP also contains at least the destination HWMP SN, the path metric, and the number of hops.

PREQ elements and PREP elements create or update the forwarding information of the mesh STAs that process these elements as follows:

- The mesh STA may create or update its forwarding information to the transmitter of the element if the path metric improves.
- The mesh STA shall create or update its forwarding information to the originator mesh STA, if it received a PREQ, and one of the following conditions is met:
  - The Originator HWMP SN > HWMP SN in the forwarding information for this originator mesh STA, or
  - The Originator HWMP SN = HWMP SN in the forwarding information for this originator mesh STA AND the updated path metric is better than the path metric in the forwarding information.
- The mesh STA shall create or update its forwarding information to the target mesh STA, if it received a PREP, and one of the following conditions is met:
  - The Target HWMP SN > HWMP SN in the forwarding information for this target mesh STA, or
  - The Target HWMP SN = HWMP SN in the forwarding information for this target mesh STA AND the updated path metric is better than the path metric in the forwarding information.

Table 13-9 defines the values to be stored in the different fields of the forwarding information after a PREQ or PREP has been received.

Table 13-9—Data for creation and update of forwarding information due to PREQ and PREP

Field of forwarding information	Received PREQ		Received PREP	
	Forwarding information for transmitter of PREQ	Forwarding information for originator mesh STA	Forwarding information for transmitter of PREP	Forwarding information for target mesh STA
HWMP SN	Invalid if created, no change if updated	PREQ field Originator HWMP Sequence Number	Invalid if created, no change if updated	PREP field Target HWMP Sequence Number
Next hop	Transmitter address of the management frame containing the PREQ element	Transmitter address of the management frame containing the PREQ element	Transmitter address of the management frame containing the PREP element	Transmitter address of the management frame containing the PREP element
Path metric	Accumulation of the initial value of the path metric with the metric of the link to the transmitter of the PREQ element	Accumulation of the value of PREQ field Metric with the metric of the link to the transmitter of the PREQ element	Accumulation of the initial value of the path metric with the metric of the link to the transmitter of the PREP element	Accumulation of the value of PREP field Metric with the metric of the link to the transmitter of the PREP element
Number of hops	1	Value of PREQ field Hop Count + 1	1	Value of PREP field Hop Count + 1
Precursor list	No change	No change except in case of an intermediate reply [see 13.10.9.4.3 step f)]	No change	See 13.10.10.4.3 step d)
Lifetime	The longer one of the lifetime of the stored forwarding information and the value of PREQ field Lifetime	The longer one of the lifetime of the stored forwarding information and the value of PREQ field Lifetime	The longer one of the lifetime of the stored forwarding information and the value of PREP field Lifetime	The longer one of the lifetime of the stored forwarding information and the value of PREP field Lifetime

Changes to the forwarding information in other situations, for instance, when processing a PERR element (see 13.10.11), are described in the corresponding clauses.

13.10.8.5 Repeated attempts at path discovery

Repeated attempts by a mesh STA at path discovery towards a single target shall be limited to dot11MeshHWMPmaxPREQretries. The minimum waiting time for the repeated attempt at path discovery to a single target is 2 × dot11MeshHWMPnetDiameterTraversalTime. For each attempt, the HWMP SN is incremented and a new Path Discovery ID is chosen.

13.10.8.6 Limiting the rate of HWMP SN increments

In order to improve path stability (and further reduce overhead), a mesh STA may use the same originator HWMP SN for a certain time interval. In this case, the originator HWMP SN shall be incremented only after at least dot11MeshHWMPnetDiameterTraversalTime has elapsed since the previous increment. This mechanism prevents mesh STAs from changing the path frequently to the originator mesh STA every time the originator mesh STA sends a burst of PREQs within a very short time. This element of the protocol allows an originator mesh STA to immediately initiate on-demand path discovery to a new target without affecting recently refreshed paths to the originator in other mesh STAs.

13.10.9 Path request (PREQ)

13.10.9.1 General

This subclause describes the function, generation, and processing of the PREQ element.

13.10.9.2 Function

The PREQ element, described in 8.4.2.115, is used for the following purposes:

— Discovering a path to one or more target mesh STAs
— Maintaining a path (optional)
— Building a proactive (reverse) path selection tree to the root mesh STA
— Confirming a path to a target mesh STA (optional)

13.10.9.3 Conditions for generating and sending a PREQ element

A mesh STA shall send a PREQ element in an HWMP Mesh Path Selection frame, as defined in 8.5.17.3, in the following cases:

Case A: Path Discovery (Original Transmission)

All of the following conditions apply:

— The mesh STA needs to establish an on-demand path to one or more targets for which there is no ongoing path discovery initiated by this mesh STA.
— The mesh STA has not sent a PREQ element for the target mesh STAs less than dot11MeshHWMPpreqMinInterval TUs ago. If this is the case, the transmission of the PREQ has to be postponed until this condition becomes true.
— The mesh STA has not made more than (dot11MeshHWMPmaxPREQretries – 1) repeated attempts at path discovery towards the target of the PREQ.

The content of a PREQ element in Case A shall be as shown in Table 13-10.

Table 13-10—Contents of a PREQ element in Case A

Field		Value
Element ID		Value given in Table 8-54 for the PREQ element
Length		26 + N × 11 (if Bit 6 (AE subfield) in the Flags field = 0) 32 + N × 11 (if Bit 6 (AE subfield) in the Flags field = 1)
Flags		Bit 0: 0 (gate announcement not applicable) Bit 1: 0 (group addressed) Bit 2: 0 (no proactive PREP applicable) Bit 3–5: Reserved Bit 6: (1 – if external address present, 0 – otherwise) Bit 7: Reserved
Hop Count		0
Element TTL		Maximum number of hops allowed for this element, e.g., dot11MeshHWMPnetDiameter.
Path Discovery ID		New unique Path Discovery ID, for instance, previous Path Discovery ID + 1
Originator Mesh STA Address		MAC address of the path originator
Originator HWMP Sequence Number		Previous Originator HWMP SN + 1. See 13.10.8.6
Originator External Address		Present only if Bit 6 in Flags field = 1. This value is set to the external address, which is the source address of the MSDU (from outside the mesh BSS) that triggered the path discovery at the originator.
Lifetime		The time for which mesh STAs receiving the PREQ consider the forwarding information to be valid, e.g., dot11MeshHWMPactivePathTimeout.
Metric		Initial value of active path selection metric
Target Count		N (N ≥ 1)
Per Target	Per Target Flags	Bit 0 (TO): dot11MeshHWMPtargetOnly Bit 1: Reserved Bit 2 (USN): 0 if forwarding information for Target Address with valid HWMP SN exists, 1 otherwise Bit 3–7: Reserved
	Target Address	MAC address of requested target
	Target HWMP Sequence Number	If Per Target Flags Bit 2 (USN) is 0, the latest HWMP SN stored by the originator mesh STA for the target mesh STA from the forwarding information (see 13.10.8.4). Otherwise, reserved.

Case B: Path Maintenance (Original Transmission) (optional)

All of the following conditions apply:

- The mesh STA has a path to a given target mesh STA that is not a root mesh STA.
- The last PREQ to this target was sent dot11MeshHWMPmaintenanceInterval TUs (or more) ago.

The content of a PREQ in Case B shall be as shown in Table 13-11.

Table 13-11—Contents of a PREQ element in Case B

Field		Value
Element ID		Value given in Table 8-54 for the PREQ element
Length		26 + N × 11
Flags		Bit 0: 0 (gate announcement not applicable) Bit 1: 0 (group addressed) Bit 2: 0 (no proactive PREP applicable) Bit 3–5: Reserved Bit 6: 0 (no address extension) Bit 7: Reserved
Hop Count		0
Element TTL		Maximum number of hops allowed for this element, e.g., dot11MeshHWMPnetDiameter
Path Discovery ID		New unique Path Discovery ID, for instance, previous Path Discovery ID + 1
Originator Mesh STA Address		MAC address of the originator of the PREQ
Originator HWMP Sequence Number		Originator HWMP SN + 1. See 13.10.8.6
Originator External Address		Field not present
Lifetime		The time for which mesh STAs receiving the PREQ consider the forwarding information to be valid, e.g., dot11MeshHWMPactivePathTimeout.
Metric		Initial value of active path selection metric
Target Count		N (N ≥ 1)
Per Target	Per Target Flags	Bit 0 (TO): 1 (target only) Bit 1: Reserved Bit 2 (USN): 0 Bit 3–7: Reserved
	Target Address	MAC Address of target mesh STA
	Target HWMP Sequence Number	The latest HWMP SN for this target known to the originator mesh STA.

Case C: Proactive PREQ (Original Transmission)

All of the following conditions apply:

- The root mesh STA is configured as root mesh STA using proactive PREQs ([dot11MeshHWMProotMode = proactivePREQnoPREP (2)] OR [dot11MeshHWMProotMode = proactivePREQwithPREP (3)]).
- The root mesh STA sent its previous proactive PREQ dot11MeshHWMProotInterval TUs ago.

The contents of a PREQ in Case C shall be as shown in Table 13-12.

Table 13-12—Contents of a PREQ element in Case C

Field		Value
Element ID		Value given in Table 8-54 for the PREQ element
Length		37
Flags		Bit 0: 1 if dot11MeshGateAnnouncements is true (gate announcement), 0 otherwise Bit 1: 0 (group addressed) Bit 2: 0 if dot11MeshHWMProotMode = proactivePREQnoPREP(2), 1 if dot11MeshHWMProotMode = proactivePREQwithPREP(3) (proactive PREP) Bit 3–5: Reserved Bit 6: 0 (no address extension) Bit 7: Reserved
Hop Count		0
Element TTL		Maximum number of hops allowed for this element, e.g., dot11MeshHWMPnetDiameter.
Path Discovery ID		New unique Path Discovery ID, for instance, previous Path Discovery ID + 1
Originator Mesh STA Address		MAC address of the root mesh STA
Originator HWMP Sequence Number		Originator HWMP SN + 1. See 13.10.8.6
Originator External Address		Field not present
Lifetime		dot11MeshHWMPactivePathToRootTimeout
Metric		Initial value of active path selection metric
Target Count		1
Per Target	Per Target Flags	Bit 0 (TO): 1 Bit 1: Reserved Bit 2 (USN): 1 Bit 3–7: Reserved
	Target Address	Broadcast address
	Target HWMP Sequence Number	0

Case D: Root Path Confirmation (Original Transmission)

One of the following conditions applies:

— The mesh STA has received a RANN and the metric (RANN metric ⊕ metric to the transmitter of the RANN) is better than the metric to the root in the current forwarding information.

— The mesh STA has a path to a root mesh STA and the last PREQ to the root mesh STA was sent dot11MeshHWMPconfirmationInterval TUs (or more) ago.

The content of a PREQ element in Case D shall be as shown in Table 13-13.

Table 13-13—Contents of a PREQ element in Case D

Field		Value
Element ID		Value given in Table 8-54 for the PREQ element
Length		As required
Flags		Bit 0: 0 (gate announcement not applicable) Bit 1: 1 (individually addressed) Bit 2: 0 (no proactive PREP applicable) Bit 3–5: Reserved Bit 6: 0 (no address extension) Bit 7: Reserved
Hop Count		0
Element TTL		Maximum number of hops allowed for this element, e.g., dot11MeshHWMPnetDiameter
Path Discovery ID		Not used
Originator Mesh STA Address		MAC address of the originator mesh STA
Originator HWMP Sequence Number		Originator HWMP SN + 1. See 13.10.8.6
Originator External Address		Field not present
Lifetime		The time for which mesh STAs receiving the PREQ consider the forwarding information to be valid, e.g., dot11MeshHWMPactivePathToRootTimeout.
Metric		Initial value of active path selection metric
Target Count		1
Per Target	Per Target Flags	Bit 0 (TO): 1 Bit 1: Reserved Bit 2 (USN): 0 Bit 3–7: Reserved
	Target Address	Root mesh STA MAC Address
	Target HWMP Sequence Number	The latest HWMP SN for this target known to the originator mesh STA

Case E: PREQ Propagation

Case E1 (target count = 1, no PREP generation as intermediate mesh STA):

All of the following conditions apply:

— The mesh STA has received and accepted a PREQ—see 13.10.9.4.2.
— dot11MeshForwarding is true.
— [The active forwarding information for the Originator Mesh STA was created or updated according to the rules defined in 13.10.8.4] OR [{the Originator HWMP SN of the accepted PREQ = HWMP SN in the forwarding information for this originator mesh STA} AND {the mesh STA has not previously received a PREQ with the same Originator Mesh STA Address and the same Path Discovery ID}].
— The Element TTL field is greater than 1—see 13.10.8.2.
— Target Count = 1.
— [The mesh STA is not the target of the PREQ)]
OR
[the target of the PREQ is the MAC broadcast address (all ones)].
— The mesh STA is not the proxy of the target address.
— [The TO (Target Only) subfield of the target in the PREQ is set (TO = 1)]
OR
[{the TO (Target Only) subfield of the target in the PREQ is not set (TO = 0)} AND {mesh STA has no active forwarding information for the requested target}].

The content of a PREQ element in Case E1 shall be as shown in Table 13-14.

Table 13-14—Contents of a PREQ element in Case E1

Field	Value
Element ID	Value given in Table 8-54 for the PREQ element
Length	As received
Flags	As received
Hop Count	As received + 1
Element TTL	As received – 1
Path Discovery ID	As received
Originator Mesh STA Address	As received
Originator HWMP Sequence Number	As received
Originator External Address	As received. This field is only present if Bit 6 of the Flags field (AE subfield) is 1.
Lifetime	As received
Metric	As received ⊕ own metric toward transmitter of received PREQ
Target Count	1

Table 13-14—Contents of a PREQ element in Case E1 *(continued)*

Field		Value
Per Target	Per Target Flags	As received
	Target MAC Address	As received
	Target HWMP Sequence Number	As received

Case E2 (target count = 1, PREP generation as intermediate mesh STA):

All of the following conditions apply:

- The mesh STA has received and accepted a PREQ—see 13.10.9.4.2.
- dot11MeshForwarding is true.
- [The active forwarding information for the Originator Mesh STA was created or updated according to the rules defined in 13.10.8.4]
  OR
  [{the Originator HWMP SN of the accepted PREQ = HWMP SN in the forwarding information for this originator mesh STA} AND {the mesh STA has not previously received a PREQ with the same Originator Mesh STA Address and the same Path Discovery ID}].
- The Element TTL field is greater than 1—see 13.10.8.2.
- Target Count = 1.
- The mesh STA is not the target of the PREQ.
- The mesh STA is not the proxy of the target address.
- The mesh STA has active forwarding information for the requested target.
- [The TO (Target Only) subfield of the target in the PREQ is not set (TO = 0)]
  AND
  [the mesh STA has active forwarding information for the requested target].

The contents of a PREQ element in Case E2 shall be as shown in Table 13-15.

Table 13-15—Contents of a PREQ element in Case E2

Field	Value
Element ID	Value given in Table 8-54 for the PREQ element
Length	As received
Flags	As received
Hop Count	As received + 1
Element TTL	As received – 1
Path Discovery ID	As received
Originator Mesh STA Address	As received
Originator HWMP Sequence Number	As received

Table 13-15—Contents of a PREQ element in Case E2 *(continued)*

Field		Value
Originator External Address		As received. This field is only present if Bit 6 of the Flags field (AE subfield) is 1.
Lifetime		As received
Metric		As received ⊕ own metric toward transmitter of received PREQ
Target Count		1
Per Target	Per Target Flags	Bit 0 (TO): 1 (target only because mesh STA sent a PREP) Bit 1: Reserved Bit 2 (USN): As received Bit 3–7: Reserved
	Target MAC Address	As received
	Target HWMP Sequence Number	As received

Case E3 (target count > 1):

All of the following conditions apply:

- — The mesh STA has received and accepted a PREQ—see 13.10.9.4.2.
- — dot11MeshForwarding is true.
- — [The active forwarding information for the Originator Mesh STA was created or updated according to the rules defined in 13.10.8.4 (Forwarding information)]
  OR
  [{the Originator HWMP SN of the accepted PREQ = HWMP SN in the forwarding information for this originator mesh STA} AND {the mesh STA has not previously received a PREQ with the same Originator Mesh STA Address and the same Path Discovery ID}].
- — The Element TTL field is greater than 1—see 13.10.8.2.
- — Target Count > 1.
- — There is at least one requested target that is neither the recipient MAC address nor an external MAC address proxied by the recipient.

The contents of a PREQ element in Case E3 shall be as shown in Table 13-16.

Table 13-16—Contents of a PREQ element in Case E3

Field	Value
Element ID	Value given in Table 8-54 for the PREQ element
Length	26 + N × 11
Flags	As received
Hop Count	As received + 1
Element TTL	As received – 1

Table 13-16—Contents of a PREQ element in Case E3 *(continued)*

Field		Value
Path Discovery ID		As received
Originator Mesh STA Address		As received
Originator HWMP Sequence Number		As received
Originator External Address		As received. This field is only present if Bit 6 of the Flags field (AE subfield) is set to 1.
Lifetime		As received
Metric		As received ⊕ own metric toward the transmitter of the received PREQ
Target Count		1 ≤ target count ≤ received target count received target count less the number of requested destinations, for which the processing mesh STA — is the target mesh STA or — is the target proxy mesh gate
Per Target #A	Per Target Flags #A	As received
	Target MAC Address #A	As received
	Target HWMP Sequence Number #A	As received
Per Target #B	Per Target Flags #B	Bit 0 (TO): 1 (target only because mesh STA sent PREP) Bit 1: As received Bit 2 (USN): As received Bit 3–7: As received
	Target MAC Address #B	As received
	Target HWMP Sequence Number #B	As received

For the per target fields (Per Target Flags, Target Address, Target HWMP Sequence Number) assume the following:

— Target #A: If target A would have been the only requested target, it would generate a PREQ for propagation according to case E1.

— Target #B: If target B would have been the only requested target, it would generate a PREQ for propagation according to case E2.

13.10.9.4 PREQ element processing

13.10.9.4.1 General

Received PREQ elements are subject to certain acceptance criteria. Processing and actions taken depend on the contents of the PREQ and the information available to the receiving mesh STA. See also 13.10.8.

13.10.9.4.2 Acceptance criteria

The PREQ element shall not be accepted (and shall not be processed as described in 13.10.9.4.3) if any of the following is true:

- (The target address of the PREQ is neither the recipient MAC address, broadcast address, nor an external MAC address proxied by the recipient) AND (dot11MeshForwarding is false)
- (Bit 1 (Addressing Mode subfield) of the Flags field in the PREQ element is equal to 1) AND (there is no valid forwarding information with the destination mesh STA address equal to the Target Address of the PREQ element)

Otherwise, the PREQ element is accepted. See also 13.10.8.

13.10.9.4.3 Effect of receipt

A mesh STA receiving a PREQ according to the acceptance criteria in 13.10.9.4.2 shall record the Path Discovery ID and the Originator Mesh STA Address. The receiving mesh STA shall create or update the active forwarding information it maintains for the originator mesh STA of the PREQ according to the rules defined in 13.10.8.4.

If the active forwarding information for the Originator Mesh STA was created or updated according to the rules defined in 13.10.8.4 or if the Target HWMP SN of the PREQ is the same as the HWMP SN in the forwarding information for the Target Mesh STA and there has been no record of the Originator Mesh STA and Path Discovery ID, the following applies:

a) If the mesh STA is the target of the PREQ or is the proxy of the target MAC address it shall initiate the transmission of a PREP to the originator mesh STA (13.10.10.3 Case A). If the PREQ carries an external address (indicated by the AE subfield in the Flags field), the mesh STA shall update its proxy information with the Originator External Address as external address, the PREQ Originator Mesh STA Address as the corresponding proxy, the HWMP Sequence Number as proxy information sequence number, and for the proxy lifetime the longer one of the value of the PREQ Lifetime field and the proxy lifetime if the proxy information already exists (see also 13.11.4.3).

b) If step a) was not applicable for the mesh STA and the AE subfield in the Flags field in the PREQ is 1, the mesh STA may update its proxy information with the Originator External Address as external address, the PREQ Originator Mesh STA Address as the corresponding proxy, the HWMP Sequence Number as proxy information sequence number, and for the proxy lifetime the longer one of the value of the PREQ Lifetime field and the proxy lifetime if the proxy information already exists (see also 13.11.4.3).

c) If the mesh STA has valid forwarding information to any of the requested targets and the TO (Target Only) subfield for such a target is not set (TO = 0), it initiates the transmission of a PREP for each of these targets (see 13.10.10.3 Case C).

d) If the mesh STA is initiating a PREP transmission on behalf of another target according to step c) (intermediate reply), it shall process all of the following:
   - Update the precursor list in its forwarding information for the target mesh STA with the next hop from the forwarding information of the originator mesh STA.
   - Update the lifetime for this precursor that is the longer one of the lifetime of the forwarding information of the target mesh STA.
   - Update the lifetime of the precursor list entry in case it already exists.
   - Update the precursor list in its forwarding information for the originator mesh STA with the next hop toward the target mesh STA.
   - Update the lifetime for this precursor that is the longer one of the lifetime of the forwarding information of the originator mesh STA.
   - Update the lifetime of the precursor list entry in case it already exists.

e) If the received PREQ is a proactive PREQ [target address is set to all ones, TO subfield is set (TO = 1)], the mesh STA generates a proactive PREP to the root mesh STA (see 13.10.10.3 Case D) depending on the setting of the Proactive PREP subfield. If the Proactive PREP subfield is 1, a proactive PREP is generated, if it is 0, a proactive PREP is generated only if a bidirectional path to the root mesh STA is required (see W.6).

f) If there are individually addressed targets in the PREQ that have not been processed in step a) or that have been processed in step c) or in step e), the receiving mesh STA shall propagate the PREQ as defined in 13.10.9.3 Case E.

13.10.10 Path reply (PREP)

13.10.10.1 General

Subclause 13.10.10 describes the function, generation, and processing of the PREP element.

13.10.10.2 Function

The PREP element is transmitted in individually addressed frames and is described in 8.4.2.116. The purpose of the PREP is as follows:

- To establish the forward path to a target mesh STA or target proxy mesh gate.
- To confirm the reverse path to the originator.

13.10.10.3 Conditions for generating and sending a PREP element

A mesh STA sends out a PREP element in an HWMP Mesh Path Selection frame, as defined in 8.5.17.3, in the following cases:

Case A: Path Discovery (Original Transmission)

A PREP is transmitted if the mesh STA has received and accepted a PREQ (see 13.10.9.4.2) fulfilling any one of the following conditions:

- The Target Address of the PREQ is the same as MAC address of the receiving mesh STA.
- The Target Address of the PREQ is an external address currently proxied by the mesh STA.

The content of the generated PREP in Case A shall be as shown in Table 13-17.

Table 13-17—Contents of a PREP element in Case A

Field	Value
Element ID	Value given in Table 8-54 for the PREP element
Length	As required
Flags	Bit 0–5: Reserved Bit 6 (AE): (1 = external address present, 0 = otherwise) Bit 7: Reserved
Hop Count	0
Element TTL	Maximum number of hops allowed for this element
Target Mesh STA Address	MAC address of the target mesh STA or target proxy mesh gate

Table 13-17—Contents of a PREP element in Case A *(continued)*

Field	Value
Target HWMP Sequence Number	HWMP SN of the target mesh STA or target proxy mesh gate after it has been updated according to 13.10.8.3
Target External Address	External target address on behalf of which the PREP is sent. Present only if Bit 6 (AE subfield) in the Flags field is 1
Lifetime	As per the PREQ that triggered the transmission of this PREP
Metric	Initial value of active path selection metric
Originator Mesh STA Address	MAC address of the originator mesh STA
Originator HWMP Sequence Number	HWMP SN of the originator mesh STA

Case B: PREP Propagation

A PREP is propagated if all of the following conditions apply:

- The mesh STA has received and accepted the PREP—see 13.10.10.4.2.
- The mesh STA is not the path originator.

The contents of a PREP element in Case B shall be as shown in Table 13-18.

Table 13-18—Contents of a PREP element in Case B

Field	Value
Element ID	Value given in Table 8-54 for the PREP element
Length	As received
Flags	As received
Hop Count	As received + 1
Element TTL	As received – 1
Target Mesh STA Address	As received
Target HWMP Sequence Number	As received
Target External Address	As received
Lifetime	As received
Metric	As received $\oplus$ own metric toward the transmitting mesh STA
Originator Mesh STA Address	As received
Originator HWMP Sequence Number	As received

Case C: Intermediate reply (Original Transmission)

A PREP is transmitted if the mesh STA has received a PREQ fulfilling all of the following conditions:

— The TO (Target Only) subfield in the corresponding Per Target Flags field in the PREQ is not set (TO = 0)

— The receiving mesh STA has active forwarding information with

a)A destination that is the same as the Target Address of the PREQ

b)An HWMP SN that is greater than or equal to the Target HWMP SN of the PREQ

c)A nonzero lifetime

The content of the generated PREP in Case C shall be as shown in Table 13-19.

Table 13-19—Contents of a PREP element in Case C

Field	Value
Element ID	Value given in Table 8-54 for the PREP element
Length	31
Flags	Bit 0–5: Reserved Bit 6 (AE): 0 Bit 7: Reserved
Hop Count	0
Element TTL	Maximum number of hops allowed for this element
Target Mesh STA Address	Target MAC address from the PREQ
Target HWMP Sequence Number	HWMP SN of the stored forwarding information of the Target of the PREQ
Target External Address	Not present
Lifetime	As per the PREQ that triggered the transmission of this PREP
Metric	Value of path metric taken from the active forwarding information for the target address of the PREQ
Originator Mesh STA Address	MAC address of the originator mesh STA
Originator HWMP Sequence Number	HWMP SN of the originator mesh STA

Case D: Proactive PREP in Proactive PREQ mode (Original Transmission)

One of the following conditions applies:

— The mesh STA has received a proactive PREQ with the Proactive PREP subfield set to 0 AND the mesh STA needs to establish or update a bidirectional path to the root mesh STA.

— The mesh STA has received a proactive PREQ with the Proactive PREP subfield set to 1.

Note that a proactive PREQ is a PREQ with a Target Address set to all ones and its TO subfield set (TO=1).

The content of the generated PREP in Case D shall be as shown in Table 13-20.

Table 13-20—Contents of a PREP element in Case D

Field	Value
Element ID	Value given in Table 8-54 for the PREP element
Length	31
Flags	Bit 0–5: Reserved Bit 6 (AE): 0 Bit 7: Reserved
Hop Count	0
Element TTL	Maximum number of hops allowed for this element
Target Mesh STA Address	MAC address of the mesh STA
Target HWMP Sequence Number	HWMP SN of the mesh STA
Target External Address	Not present
Lifetime	Lifetime of the PREQ that triggered the transmission of this PREP
Metric	Initial value of active path selection metric
Originator Mesh STA Address	MAC address of the root mesh STA (originator mesh STA of the PREQ)
Originator HWMP Sequence Number	HWMP SN of the root mesh STA (originator HWMP SN of the PREQ)

13.10.10.4 PREP element processing

13.10.10.4.1 General

Received PREP elements are subject to certain acceptance criteria. Processing and actions taken depend on the contents of the PREP and the information available to the receiving mesh STA.

13.10.10.4.2 Acceptance criteria

The PREP element shall not be accepted (and shall not be processed as described in 13.10.10.4.3) if any of the following is true:

— (The Originator Mesh STA Address of the PREP is neither the recipient MAC address nor an external MAC address proxied by the recipient) AND (dot11MeshForwarding is false).

Otherwise, the PREP element shall be accepted.

13.10.10.4.3 Effect of receipt

A mesh STA receiving a PREP according to the acceptance criteria in 13.10.10.4.2 shall create or update the active forwarding information it maintains for the target mesh STA of the PREP (according to the rules defined in 13.10.8.4). If the conditions for creating or updating the forwarding information have not been met in those rules, no further steps are applied to the PREP.

If the active forwarding information was created or updated according to the rules defined in 13.10.8.4, the following apply:

a) If the receiving mesh STA is not the final destination of the PREP (originator mesh STA) and the field Element TTL > 1, the PREP is propagated as defined in 13.10.10.3 Case B.
b) If the receiving mesh STA is the final destination of the PREP (originator mesh STA) and its AE subfield in the Flags field is 1, the mesh STA shall store the Target External Address, the Target Mesh STA Address, and the HWMP Sequence Number as proxy information sequence number in its proxy information. The proxy lifetime is the longer one of the value of the PREP Lifetime field and the proxy lifetime if the proxy information already exists (see also 13.11.4.3).
c) If the receiving mesh STA is not the final destination of the PREP (originator mesh STA) and its AE subfield in the Flags field is 1, the mesh STA may store the Target External Address, the Target Mesh STA Address, and the HWMP Sequence Number as proxy information sequence number in its proxy information. The proxy lifetime is the longer one of the value of the PREP Lifetime field and the proxy lifetime if the proxy information already exists (see also 13.11.4.3).
d) If the mesh STA propagates the PREP, the precursor list for the Target Mesh STA Address is updated by adding the next-hop mesh STA to which the PREP is propagated. In addition, at the mesh STA the precursor list for the originator mesh STA address is updated by adding the next-hop mesh STA towards the Target Address. The lifetimes of these entries in the precursor lists are the values of the lifetimes of the corresponding forwarding information.

13.10.11 Path error (PERR)

13.10.11.1 General

Subclause 13.10.11 describes the function, generation, and processing of the PERR element.

13.10.11.2 Function

The PERR element is used for announcing one or more unreachable destination(s). The announcement is sent to all traffic sources that have a known active path to the destination(s). The active forwarding information associated with the unreachable destination(s) should no longer be used for forwarding.

A PERR element may be either group addressed (if there are many precursors), individually addressed (if there is only one precursor), or individually addressed iteratively to all precursors (see 13.10.7, item “PERR individually addressed”). The PERR element is processed as a single element when iteratively individually addressed to several precursors. The PERR element contains the destinations that are unreachable.

A PERR element is propagated by mesh STAs receiving a PERR if certain conditions are met.

A mesh STA generating or receiving a PERR may decide to establish paths to unreachable destinations using any of the available HWMP mechanisms.

13.10.11.3 Conditions for generating and sending a PERR element

A mesh STA shall send out a PERR element in an HWMP Mesh Path Selection frame, as defined in 8.5.17.3, in the following cases:

Case A: Original transmission (next hop is unusable)

The mesh STA has not sent a PERR element less than dot11MeshHWMPperrMinInterval TUs ago, and the following condition applies:

— The mesh STA determines that the link to the next hop of an active path in its forwarding information is no longer usable.

NOTE—The detection might be triggered by the fact that a mesh STA is unable to forward an MSDU/MMPDU to a next-hop mesh STA.

The HWMP SN in the forwarding information of all unreachable destinations announced in this PERR is incremented by 1. The forwarding information for each unreachable destination announced in this PERR is invalidated.

The contents of a PERR element in Case A shall be as shown in Table 13-21.

Table 13-21—Contents of a PERR element in Case A

Field	Value
Element ID	Value given in Table 8-54 for the PERR element
Length	2 + N × 13
Element TTL	The maximum number of hops the element is propagated before being discarded.
Number of Destinations	Number of announced unreachable destinations in the PERR.
Flags #1	Bit 0–5: Reserved Bit 6 (AE): 0 Bit 7: Reserved
Destination Address #1	MAC address of unreachable destination #1.
HWMP Sequence Number #1	HWMP SN for Destination Address #1 from the forwarding information after above increment.
Reason Code #1	“MESH-PATH-ERROR-DESTINATION-UNREACHABLE” (see 8.4.1.7).
...	...

Case B: Original transmission (missing forwarding information)

The mesh STA has not sent a PERR element less than dot11MeshHWMPperrMinInterval TUs ago, and one of the following conditions applies:

— The mesh STA receives an individually addressed frame with a destination address not matching its own MAC address for which it has no forwarding information.

— The mesh STA receives an individually addressed frame with a destination address not matching its own MAC address and dot11MeshForwarding is false.

The contents of a PERR element in Case B shall be as shown in Table 13-22.

Table 13-22—Contents of a PERR element in Case B

Field	Value
Element ID	Value given in Table 8-54 for the PERR element.
Length	2 + N × 13
Element TTL	The maximum number of hops the element is propagated before being discarded.
Number of Destinations	Number of announced destinations with missing forwarding information in the PERR.
Flags #1	Bit 0–5: Reserved Bit 6 (AE): 0 Bit 7: Reserved
Destination Address #1	MAC address of destination with missing forwarding information #1. This is Address 3 of the received individually addressed frame.
HWMP Sequence Number #1	Reserved (0)
Reason Code #1	“MESH-PATH-ERROR-NO-FORWARDING-INFORMATION” (see 8.4.1.7).
...	...

Case C: Original transmission (proxy information is unusable)

The mesh STA has not sent a PERR element less than dot11MeshHWMPperrMinInterval TUs ago, and the following condition applies:

— The mesh STA is a proxy mesh gate and determines that an active proxy information where the mesh STA is the proxy mesh gate is no longer usable.

The contents of a PERR element in Case C shall be as shown in Table 13-23.

Table 13-23—Contents of a PERR element in Case C

Field	Value
Element ID	Value given in Table 8-54 for the PERR element.
Length	2 + N × 19
Element TTL	The maximum number of hops the element is propagated before being discarded.
Number of Destinations	Number of announced unreachable external destinations in the PERR.
Flags #1	Bit 0–5: Reserved Bit 6 (AE): 1 Bit 7: Reserved
Destination Address #1	MAC address of proxy mesh gate #1 with unusable active proxy information.
HWMP Sequence Number #1	Last used HWMP SN for Destination Address #1.

Table 13-23—Contents of a PERR element in Case C *(continued)*

Field	Value
Destination External Address #1	External MAC address of the active proxy information that is not longer usable and for which the mesh STA is the proxy mesh gate.
Reason Code #1	“MESH-PATH-ERROR-NO-PROXY-INFORMATION” (see 8.4.1.7.
...	...

Case D: PERR propagation

The mesh STA has not sent a PERR element less than dot11MeshHWMPperrMinInterval TUs ago, and all of the following conditions apply:

— The mesh STA received a PERR from a neighbor peer mesh STA.

— A destination in the PERR is the same as one of the destinations in the active forwarding information of the mesh STA where the next hop is the transmitter of the received PERR, and the forwarding information or the proxy information has been invalidated according to conditions in 13.10.11.4.3 case b), case c), or case d).

— dot11MeshForwarding is true.

— The Element TTL field in the received PERR element is greater than 1.

The contents of a PERR element in Case D shall be as shown in Table 13-24.

Table 13-24—Contents of a PERR element in Case D

Field	Value
Element ID	Value given in Table 8-54 for the PERR element

Length	if AE subfield = 0: 2 + N × 13
	if AE subfield = 1: 2 + N × 19

Element TTL	Element TTL in received PERR element – 1
Number of Destinations	1 ≤ number of destinations in the PERR ≤ received value Received number of destinations less the number of received destinations for which the transmitter of the PERR is not the next hop
Flags #1	As received
Destination Address #1	MAC address of unreachable destination #1, as received
HWMP Sequence Number #1	If Reason Code #1 = “MESH-PATH-ERROR-NO-FORWARDING-INFORMATION” and received value = 0, then HWMP SN for Destination Address #1 from the forwarding information after the increment of 13.10.10.4.3 step b).?? scan for “step” Otherwise, as received.
Destination External Address #1	As received This field is only present if Bit 6 (AE subfield) of the Flags field #1 is 1.
Reason Code #1	As received
...	...

13.10.11.4 PERR element processing

13.10.11.4.1 General

Received PERR elements are subject to certain acceptance criteria. Processing and actions taken depend on the contents of the PERR and the information available to the receiving mesh STA. See also 13.10.8.

13.10.11.4.2 Acceptance criteria

The PERR shall be accepted (and shall be processed as described in 13.10.11.4.3) if the following applies:

- The mesh STA that receives the PERR has forwarding information stored where
  - The destination is contained in the list of unreachable destinations of the PERR and
  - The next hop is the transmitter of the received PERR

Otherwise, the PERR element shall be discarded.

13.10.11.4.3 Effect of receipt

The following applies only to a PERR element that was accepted according to the acceptance criteria in 13.10.11.4.2:

a) The mesh STA creates a list of unreachable destinations consisting of those destinations from the received PERR for which the next hop in the local active forwarding information is the transmitter of the PERR. Step b) through step e) are applied to the destinations in this list.

b) If the Reason Code is “MESH-PATH-ERROR-NO-FORWARDING-INFORMATION” and the HWMP Sequence Number is 0, the receiving mesh STA increments the HWMP SN in the forwarding information of the listed unreachable destination by 1 and invalidates the forwarding information.

c) If the Reason Code is “MESH-PATH-ERROR-NO-FORWARDING-INFORMATION” and the HWMP Sequence Number is not 0 or the Reason Code is “MESH-PATH-ERROR-DESTINATION-UNREACHABLE” and the received HWMP SN for a listed unreachable destination is higher than the current HWMP SN in the forwarding information for that destination, the receiving mesh STA shall consider that destination unreachable and shall set the HWMP SN in the forwarding information to the HWMP SN received in the PERR and shall invalidate the forwarding information associated with this unreachable destination.

d) If the Reason Code is “MESH-PATH-ERROR-NO-PROXY-INFORMATION,” the receiving mesh STA shall consider the corresponding Destination External Address unreachable and shall invalidate the proxy information associated with this unreachable external destination (proxy mesh gate is the Destination Address of the PERR, external MAC address is the Destination External Address of the PERR, proxy information sequence number is the HWMP Sequence Number).

e) A PERR element is propagated according to the conditions defined in 13.10.11.3 Case D “PERR propagation.”

13.10.12 Root announcement (RANN)

13.10.12.1 General

Subclause 13.10.12 describes the function, generation, and processing of the RANN element.

13.10.12.2 Function

The RANN element, described in 8.4.2.114, is used for announcing the presence of a mesh STA configured as root mesh STA using the proactive RANN mechanism. RANN elements are sent out periodically by the root mesh STA.

The RANN element propagates path metric information across the network so that each mesh STA can select a best metric path to the announced root mesh STA. This mechanism allows bidirectional trees to be built, using a robust procedure based on individually addressed frames initiated by the mesh STAs. This procedure makes the root mesh STA aware of all mesh STAs.

Receiving mesh STAs shall propagate the RANN as described in 13.10.12.3 Case B.

13.10.12.3 Conditions for generating and sending a RANN element

A mesh STA sends out a RANN element in an HWMP Mesh Path Selection frame, as defined in 8.5.17.3, in the following cases:

Case A: Original transmission

All of the following conditions apply:

- The mesh STA is configured as a root mesh STA using the proactive RANN mechanism [dot11MeshHWMProotMode = rann (4)].
- The root mesh STA sent its previous RANN dot11MeshHWMPrannInterval TUs ago.

The contents of a RANN element in Case A shall be as shown in Table 13-25.

Table 13-25—Contents of a RANN element in Case A

Field	Value
Element ID	Value given in Table 8-54 for the RANN element
Length	21
Flags	Bit 0: set to 1 if dot11MeshGateAnnouncements is true, set to 0 otherwise. Bit 1–7: Reserved
Hop Count	0
Element TTL	Maximum number of hops allowed for this element
Root Mesh STA Address	MAC address of the root mesh STA
HWMP Sequence Number	Last used HWMP SN of the root mesh STA + 1
Interval	dot11MeshHWMPrannInterval
Metric	Initial value of active path selection metric

Case B: Propagation

All of the following conditions apply:

— The mesh STA has valid forwarding information to a root mesh STA using the proactive RANN mechanism [dot11MeshHWMProotMode = rann (4)].
— The mesh STA sent its previous RANN dot11MeshHWMPrannInterval TUs ago.
— dot11MeshForwarding is true.
— The Element TTL field is greater than 1—see 13.10.8.2.

The contents of a RANN element in Case B shall be as shown in Table 13-26.

Table 13-26—Contents of a RANN element in Case B

Field	Value
Element ID	Value given in Table 8-54 for the RANN element
Length	As received
Flags	As received
Hop Count	As received + 1
Element TTL	As received – 1
Root Mesh STA Address	As received
HWMP Sequence Number	As received
Interval	As received
Metric	As received ⊕ own link metric toward the transmitting mesh STA

13.10.12.4 RANN element reception

13.10.12.4.1 General

Received RANN elements are subject to certain acceptance criteria. Processing and actions taken depend on the content of the RANN and the forwarding information maintained by the receiving mesh STA. See also 13.10.8.

13.10.12.4.2 Acceptance criteria

The RANN element shall not be accepted (and shall not be processed as described in 13.10.12.4.3) if any of the following is true:

— The HWMP Sequence Number < previous HWMP SN from this originating root mesh STA
— (The HWMP Sequence Number = previous HWMP SN) AND (updated path metric is *worse than* previous path metric)

Otherwise, the RANN element shall be accepted.

13.10.12.4.3 Effect of receipt

The following applies only to a RANN element that was accepted according to the acceptance criteria in 13.10.12.4.2:

a) The receiving mesh STA shall set dot11MeshHWMPrannInterval to the value of the Interval field of the received RANN.
b) The receiving mesh STA may initiate a PREQ/PREP exchange with the root mesh STA to set up or update a path to the root mesh STA. See 13.10.9.3 Case D.
c) The receiving mesh STA may record the Root Mesh STA Address, together with the HWMP Sequence Number, Hop Count, and Metric in order to assist in executing 13.10.9.3 Case D.

The receiving mesh STA shall transmit a RANN if the conditions defined in 13.10.12.3 Case B are true.

13.10.13 Considerations for support of STAs without mesh functionality

The verification, by the mesh STA collocated with the AP, of disjunct MAC addresses between a non-AP STA without mesh functionality and mesh STAs during authentication/association of the non-AP STA without mesh functionality (see 10.3.6) may be done by issuing a PREQ for the MAC address of the non-AP STA without mesh functionality by the mesh STA collocated with the AP. The TO (Target Only) subfield of the Per Target Flags field of the PREQ shall be set to 1.

The MAC address of the non-AP STA already exists in the MBSS if the AP with mesh functionality receives a PREP for the MAC address of the non-AP STA and it can be derived from the PREP that the requested MAC address is originated from a mesh STA. (The AE subfield of the Flags field of the PREP is set to 0; see 8.4.2.116.)

13.11 Interworking with the DS

13.11.1 Overview of interworking between a mesh BSS and a DS

A mesh STA that has access to a DS is called a mesh gate. Mesh STAs in an MBSS access the DS via the mesh gate. An MBSS functions like an IEEE 802 LAN segment that is compatible with IEEE 802.1D. The MBSS appears as a single access domain.

An MBSS may contain two or more mesh gates. When multiple mesh gates in an MBSS have access to the same DS, the MBSS has more than one “port” (in the sense of IEEE Std 802.1D-2004, for example) through which it accesses the DS. Accordingly, broadcast loops may occur. Therefore, mesh gates should implement a loop preventing protocol in the DS.

NOTE—In the DS a typical implementation uses the Rapid Spanning Tree Protocol (RSTP) as specified in IEEE Std 802.1D-2004. With RSTP the resulting active DS topology forms a tree. Then, even if multiple mesh gates connect with the same DS, the MBSS only accesses the DS through a single mesh gate.

When dot11MeshGateAnnouncements is true, the mesh gate announces its presence to other mesh STAs in the MBSS. The mesh gate uses the gate announcement protocol (see 13.11.2) or alternatively one of the HWMP proactive path selection methods with the Gate Announcement field equal to 1:

— The proactive PREQ mechanism (see 13.10.4.2), with the Gate Announcement field equal to 1 (see 13.10.9.3)
— The proactive RANN mechanism (see 13.10.4.3), with the Gate Announcement field equal to 1 (see 13.10.12.3)

When the mesh gate uses one of the HWMP proactive path selection methods, the gate announcement protocol is not used.

A mesh STA discovers the presence of a mesh gate with access to the external network by receiving GANN elements (or PREQ and RANN with the Gate Announcement field equal to 1 if using such mechanisms). Mesh STAs propagate these elements to neighbor mesh STAs in order to propagate the information throughout the MBSS.

NOTE—The decision to set dot11MeshGateAnnouncements to true is beyond the scope of the standard. In general, the mesh gate announces that it has access to a broader network beyond the MBSS, using gate announcement protocol or HWMP proactive path selection methods with the Gate Announcement field equal to 1. One example of this configuration is that the mesh gate has access to a portal through the DS.

When a mesh gate has access to IEEE 802 STAs outside the mesh BSS (a mesh STA collocated with an AP, another mesh STA that belongs to another MBSS, etc.), the mesh gate acts as an intermediary for the IEEE 802 STAs outside the MBSS so that the forwarding information inside the MBSS only contains addresses that belong to the MBSS. The mesh gate acting as an intermediary for external STAs is termed proxy mesh gate. When the end station of an IEEE 802 communication is an external STA, mesh STAs handle addresses of the end-to-end IEEE 802 communication as depicted in Figure 9-42. Proxy mesh gate operation is described in 13.11.4.

13.11.2 Gate announcement (GANN)

13.11.2.1 General

Subclause 13.11.2 describes the function, generation, and processing of the GANN element.

13.11.2.2 Function

The GANN element, described in 8.4.2.113, is used to announce the presence of a mesh gate with dot11MeshGateAnnouncements equal to true in the mesh BSS. Gate announcements allow mesh STAs to discover such a mesh gate and, if necessary, to build a path towards it.

13.11.2.3 Conditions for generating and sending a GANN element

A mesh STA shall send a GANN element in a Gate Announcement frame, as defined in 8.5.17.4, in the following cases:

Case A: Original transmission

The mesh STA is a mesh gate not sending PREQ or RANN with the Gate Announcement field equal to 1 and dot11MeshGateAnnouncements is true. The mesh STA shall transmit the Gate Announcement frame at every dot11MeshGateAnnouncementInterval.

The content of a GANN element in Case A shall be as shown in Table 13-27.

Table 13-27—Contents of a GANN element in Case A

Field	Value
Element ID	Value given in Table 8-54 for the GANN element
Length	15
Flags	Reserved

Table 13-27—Contents of a GANN element in Case A *(continued)*

Field	Value
Hop Count	0
Element TTL	Maximum number of hops allowed for the gate announcement
Mesh Gate Address	Mesh STA MAC address
GANN Sequence Number	Previous GANN sequence number + 1
Interval	dot11MeshGateAnnouncementInterval

The mesh gate shall assign the GANN Sequence Number from a single modulo-2^{32} counter, starting at 0 and incrementing by 1 for each GANN element transmission.

Case B: Propagation

All of the following conditions are met:

— The mesh STA has received and accepted a gate announcement.
— The decremented Element TTL of the gate announcement is equal to or greater than 1.
— dot11MeshForwarding is true.

The content of a GANN element in Case B shall be as shown in Table 13-28.

Table 13-28—Contents of a GANN element in Case B

Field	Value
Element ID	Value given in Table 8-54 for the GANN element
Length	15
Flags	As received
Hop Count	As received + 1
Element TTL	As received – 1
Mesh Gate Address	As received
GANN Sequence Number	As received
Interval	As received

13.11.2.4 GANN element processing

13.11.2.4.1 General

A received gate announcement is subject to certain acceptance criteria. Processing depends on the contents of the gate announcement and the information available at the receiving mesh STA.

13.11.2.4.2 Acceptance criteria

The GANN element shall not be accepted (and shall not be processed as described in 13.11.2.4.3) if the GANN Sequence Number of the gate announcement is equal or lower than the GANN Sequence Number of the most recently accepted gate announcement with the same Mesh Gate Address.

13.11.2.4.3 Effect of receipt

The following applies only to a GANN element that was accepted according to the acceptance criteria in 13.11.2.4.2. The receiving mesh STA shall transmit a gate announcement as described in 13.11.2.3, Case B.

The Mesh Gate Address field of the GANN contains the address of the mesh gate, and may be stored for the purpose of determining paths to the mesh gates. Paths to mesh gates allow mesh STAs to forward MSDUs to addresses for which no path could be determined (see 9.32.9).

13.11.3 Data forwarding at proxy mesh gates

13.11.3.1 General

Forwarding of MSDUs from the DS into the MBSS by a proxy mesh gate follows the procedures given in 9.32.3.

Forwarding of MSDUs from the MBSS into the DS by a proxy mesh gate follows the procedures that apply for the specific collocated network.

A proxy mesh gate learns the addresses of the other proxy mesh gates in the MBSS and of external addresses proxied by them through the receipt of path selection messages and messages carrying proxy information (for example, see 8.4.2.118).

13.11.3.2 Forwarding of MSDUs from the MBSS to the DS

On receipt of an individually addressed Mesh Data frame from the MBSS with Address Extension Mode equal to 10 (binary), a proxy mesh gate shall perform the following:

- If Address 5 is a known destination MAC address in the proxy information (external address) and proxied by the proxy mesh gate, the proxy mesh gate forwards the MSDU to the external address through the DS.
- If Address 5 is a known destination MAC address in the proxy information (external address) and proxied by a different proxy mesh gate, the MSDU is forwarded through the MBSS to the proxy mesh gate that proxies the external address. The MSDU is sent into the MBSS according to the procedures in 9.32.4.1 as an individually addressed Mesh Data frame with Address 3 set to the MAC address of the proxy mesh gate of the proxy information proxying Address 5, Address 4 set to the MAC address of this proxy mesh gate, and Address 5 and Address 6 kept unchanged.
- If Address 5 is unknown to the proxy mesh gate, the mesh gate forwards the MSDU to the DS. The mesh gate may send an error notification to the mesh source of the MSDU. In HWMP, this is done by sending a PERR as described in 13.10.11.3 Case C.

On receipt of group addressed Mesh Data frame from the MBSS with Address Extension Mode equal to 01 (binary), a proxy mesh gate shall forward the MSDU to the DS using a group addressed frame.

13.11.3.3 Forwarding of MSDUs from the DS to the MBSS

On receipt of an individually addressed MSDU from the DS, a proxy mesh gate shall perform the following depending on the possible destination:

a) If the destination of the MSDU is a mesh STA address that the mesh gate knows to be inside the MBSS, the mesh gate forwards the MSDU according to the procedures for frame addressing and data forwarding at source mesh STAs in an MBSS (9.32.4.1). The MSDU shall be transmitted using a frame with the four-address MAC header format (with the Address Extension Mode subfield in the Mesh Control field set to 10 (binary)), where the Mesh Address Extension subfield in the Mesh Control field carries the addresses of the end stations, as specified in row "Mesh Data (proxied, individually addressed)" of Table 9-15. The address fields are set as follows:

   — Address 1: The address of the next-hop mesh STA (toward the destination mesh STA according to the forwarding information—see 9.32.2)
   — Address 2: The address of the proxy mesh gate
   — Address 3: The address of the destination mesh STA
   — Address 4: The address of the proxy mesh gate
   — Address 5: The address of the destination end mesh STA that is the same as Address 3
   — Address 6: The address of the source end mesh STA that is the source address of the MSDU received from the DS

b) If the destination of the MSDU is an external address that is proxied by another proxy mesh gate in the MBSS, the mesh gate forwards the MSDU according to the procedures for frame addressing and data forwarding at source mesh STAs in an MBSS (9.32.4.1). The MSDU shall be transmitted using a frame with the four-address MAC header format [with the Address Extension Mode subfield in the Mesh Control field set to 10 (binary)], where the Mesh Address Extension subfield in the Mesh Control field carries the addresses of the end stations, as specified in row "Mesh Data (proxied, individually addressed)" of Table 9-15. The address fields are set as follows:

   — Address 1: The address of the next-hop mesh STA (toward the proxy mesh gate of the destination of the MSDU as derived from the proxy information (see 13.11.4.2) and according to the forwarding information—9.32.2)
   — Address 2: The address of this proxy mesh gate
   — Address 3: The address of the proxy mesh gate of the destination of the MSDU as derived from the proxy information (see 13.11.4.2)
   — Address 4: The address of this proxy mesh gate
   — Address 5: The address of the destination end mesh STA that is the destination address of the MSDU received from the DS
   — Address 6: The address of the source end mesh STA that is the source address of the MSDU received from the DS

c) If the MSDU has a destination address that is unknown to the mesh gate, the mesh gate forwards the MSDU to other known mesh gates in the MBSS as an individually addressed frame according to the procedures for frame addressing and data forwarding of individually addressed frames at source mesh STAs in an MBSS (9.32.4.1). The MSDU shall be transmitted using a frame with the four-address MAC header format [with the Address Extension Mode subfield in the Mesh Control field set to 10 (binary)], where the Mesh Address Extension subfield in the Mesh Control field carries the addresses of the end stations, as specified in row "Mesh Data (proxied, individually addressed)" of Table 9-15. The address fields are set as follows:

   — Address 1: The address of the next-hop mesh STA (toward the other known mesh gate in the MBSS according to the forwarding information—see 9.32.2)
   — Address 2: The address of this proxy mesh gate
   — Address 3: The address of the other known mesh gate in the MBSS
   — Address 4: The address of this proxy mesh gate
   — Address 5: The address of the destination end mesh STA that is the unknown destination address of the MSDU received from the DS
   — Address 6: The address of the source end mesh STA that is the source address of the MSDU received from the DS

Note that the procedure to determine that an address is unknown depends on the active path selection protocol. It may require an attempt to establish a path to the destination (see 13.8).

On receipt of a group addressed MSDU from the DS, the mesh gate forwards the MSDU according to the procedures for frame addressing and data forwarding of group addressed frames at source mesh STAs in an MBSS (9.32.5.1). The MSDU shall be transmitted using a frame with the three-address MAC header format [with the Address Extension Mode subfield in the Mesh Control field set to 01 (binary)], where the Mesh Address Extension subfield in the Mesh Control field carries the address of the source end stations, as specified in row "Mesh Data (proxied, group addressed)" of Table 9-15. The address fields are set as follows:

- Address 1: The group address
- Address 2: The address of the proxy mesh gate
- Address 3: The address of the proxy mesh gate
- Address 4: The address of the source external STA

13.11.4 Proxy information and proxy update

13.11.4.1 General

Forwarding information of mesh STAs only contains addresses of mesh STAs that belong to the MBSS. However, the end station of the IEEE 802 communication may be an IEEE 802 station outside the MBSS, and such station is called external STA. Examples of external STAs are as follows:

- STAs that are associated with an AP that is collocated with a mesh STA
- STAs that are behind a mesh gate

Mesh STAs forward MSDUs to external STAs by treating MAC addresses of the external STAs as external addresses. The mesh STAs that are the destination mesh STAs of the messages destined to external STAs are called proxy mesh gates, and their MAC addresses are called proxy addresses.

NOTE—External STAs are reached using mesh services solely, i.e., they are not part of an MBSS. The mechanism by which the proxy mesh gate bridges the MBSS and the external STAs are beyond the scope of the standard. However, the standard describes the method by which mesh STAs use the external addresses that are discovered and bridged by the proxy mesh gate.

13.11.4.2 Proxy information

Proxy mesh gates and source mesh STAs of MSDUs destined to external STAs maintain proxy information. Proxy information contains the external address, the corresponding proxy address, the sequence number of the proxy information, and the corresponding proxy information lifetime.

Mesh STAs can learn the addresses of proxy mesh gates and of the external stations proxied by these proxy mesh gates through the receipt of proxy update messages or path selection messages carrying proxy information. Particularly, proxy information is updated in the following circumstances:

- A mesh STA receives and processes a proxy update (see 13.11.4.3)
- A mesh STA receives and processes an element of the active path selection protocol containing proxy information. In HWMP, these are PREQ elements (see 13.10.9.4.3), PREP elements (see 13.10.10.4.3), and PERR elements (see 13.10.11.4.3)

Additionally, proxy mesh gates may also proactively maintain proxy information on external STAs.

When the proxy information lifetime is specified, a mesh STA shall maintain the proxy information as valid information until the lifetime expires. Details of the lifetime are described in 13.11.4.3.4.

The sequence number of the proxy information and the proxy mesh gate address define a chronological order of the proxy information of an external STA at a specific proxy mesh gate.

When the proxy information is created at the proxy mesh gate, the proxy sequence number is initialized to an arbitrary value. The proxy information sequence number in the proxy information at the proxy mesh gate is incremented by 1 before the transmission of the proxy information to another mesh station. The proxy information sequence number shall be incremented if the proxy information is invalidated.

If proxy information is transmitted in HWMP elements (PREQ, PREP, and PERR), the proxy information sequence number is set to the HWMP SN of the HWMP element containing this proxy information.

Comparison of the proxy information sequence numbers is performed using a circular modulo 2^{32} comparison.

Valid proxy information is used to determine and set Address 5 and Address 6 in individually addressed Mesh Data frames, or Address 4 in group addressed Mesh Data frames.

13.11.4.3 Proxy update (PXU)

13.11.4.3.1 General

Subclause 13.11.4.3 describes the function, generation, and processing of the PXU element.

13.11.4.3.2 Function

A mesh STA generates a PXU element to inform a destination mesh STA about proxy information of external addresses that are reachable through the proxy mesh gate specified in the PXU element.

NOTE—Typically, a proxy mesh gate generates and sends a PXU element to another proxy mesh gate in the MBSS or a mesh STA that originates traffic to the external stations proxied by the proxy mesh gate. However, the standard also allows other usage of the PXU element.

The PXU element is transmitted in a Proxy Update frame (an individually addressed frame). The Proxy Update frame may contain multiple PXU elements when needed (for instance, the proxy mesh gate has a large number of proxy information).

13.11.4.3.3 Conditions for generating and sending a PXU element

A proxy mesh gate may transmit a PXU when it adds, updates, or deletes an external address to (or from) its proxy information. A proxy mesh gate may also transmit a PXU at periodic intervals.

A mesh STA that holds proxy information of a proxy mesh gate in the MBSS may also transmit a PXU.

A mesh STA may retransmit the same PXU element repeatedly until the mesh STA receives a PXUC element from the destination mesh STA. See 13.11.4.4.

The content of a PXU element shall be as shown in Table 13-29.

The proxy mesh gate shall assign the PXU ID from a single modulo-256 counter, starting at 0 and incrementing by 1 for each PXU element.

Table 13-29—Contents of a PXU element

Field		Value/description
Element ID		Value given in Table 8-54 for the PXU element
Length		8 + length of N Proxy Information fields
PXU ID		Previous PXU ID + 1
PXU Originator MAC Address		MAC address of the originator of the PXU
Number of Proxy Information (N)		Number of proxy information reported to the destination mesh STA (N ≥ 1).
Per Proxy Information	Flags	Bit 0: 0: add proxy information; 1: delete proxy information Bit 1: 0: Proxy MAC Address field present; 1: Proxy MAC Address = PXU Originator MAC Address, Proxy MAC Address field not present Bit 2: 0: Proxy Information Lifetime field not present; 1: Proxy Information Lifetime field present. If Bit 0 is 1, Bit 1 shall be set to 0. Bit 3–7: Reserved
	External MAC Address	MAC address of the STA proxied by the proxy mesh gate.
	Proxy Information Sequence Number	Proxy information sequence number of the proxy information after being incremented. See 13.11.4.2.
	Proxy MAC Address	MAC address of the proxy mesh gate. This field is only present if Bit 1 of the Flags field is 0.
	Proxy Information Lifetime	The proxy information lifetime of this proxy information as taken from the proxy information of the originator of the PXU.

13.11.4.3.4 Effect of receipt of a PXU element

A mesh STA that receives the PXU element shall update its proxy information with the list of proxy information reported in the PXU under the following conditions:

- Proxy information for the external MAC address and the proxy mesh gate reported in the Proxy Information field of the PXU does not exist at the mesh STA.
- Proxy information for the external MAC address and the proxy mesh gate reported in the Proxy Information field of the PXU does exist at the mesh STA and the value of the Proxy Information Sequence Number subfield in the received PXU is larger than the value of the proxy information sequence number in the proxy information at the mesh STA.

When multiple PXU elements are contained in the received Proxy Update frame, the recipient mesh STA shall process all of the PXU elements in the frame.

The MAC address of the proxy mesh gate is taken from the Proxy MAC Address subfield in the Proxy Information field when bit 1 in the Flags subfield is equal to 0, and from the PXU Originator MAC Address field in the PXU element when bit 1 in the Flags subfield is equal to 1.

The MAC address of the external STA is taken from the External MAC Address subfield of the corresponding Proxy Information field in the received PXU element.

The sequence number of the proxy information is taken from the Proxy Information Sequence Number subfield of the corresponding Proxy Information field in the received PXU element.

If the Proxy Information Lifetime subfield is present (the Lifetime subfield in the Flags subfield is 1) and there is already proxy information stored for the proxy mesh gate and external address reported in the proxy information of the PXU element, the mesh STA shall set the proxy lifetime to the larger one of the proxy lifetime reported by the PXU and the stored proxy information.

If the Proxy Information Lifetime subfield is present (bit 2 of the Flags subfield is 1) and there is proxy information stored for the proxy mesh gate and external address reported in the proxy information of the PXU element, the mesh STA shall set the proxy information lifetime to the value in the Proxy Information Lifetime subfield.

If the Proxy Information Lifetime subfield is not present, the lifetime of the proxy information is the same as the lifetime of the path to the proxy address. Alternatively, the lifetime of the proxy information may be set to a value representing infinity.

The destination mesh STA that received the PXU shall send a PXUC element to the originator mesh STA of the PXU as described in 13.11.4.4.3.

13.11.4.4 Proxy update confirmation (PXUC)

13.11.4.4.1 General

Subclause 13.11.4.4 describes the function, generation, and processing of the PXUC element.

13.11.4.4.2 Function

A PXUC element is generated by the destination mesh STA of a PXU to inform the sender of the PXU that the PXU has been properly received.

The PXUC element is transmitted in a Proxy Update Confirmation frame (an individually addressed frame). The Proxy Update Confirmation frame may contain multiple PXUC elements in order to confirm the reception of multiple PXU elements to the destination of the Proxy Update Confirmation frame.

13.11.4.4.3 Conditions for generating and sending a PXUC element

The destination mesh STA of a Proxy Update frame containing a PXU element shall send a PXUC element to the originator mesh STA of the PXU element.

The content of a PXUC element shall be as shown in Table 13-30.

Table 13-30—Contents of a PXUC element

Field	Value/description
Element ID	Value given in Table 8-54 for the PXUC element
Length	7
PXU ID	PXU ID of the PXU that is being confirmed
Destination mesh STA Address	MAC address of the originator of the PXUC

13.11.4.4.4 Effect of receipt of PXUC element

If a mesh STA receives a PXUC element in a PXUC frame in response to a PXU element it originated, the mesh STA shall no longer send any PXUs with the same PXU ID as given in the received PXUC element.

13.11.5 Mesh STA collocation

A mesh STA collocated with another STA shall use a MAC address that is different from the one used by the collocated STA. This precludes ambiguities relating to the presence of the Mesh Control field in the Frame Body (see 8.2.4.7), GTK use (see 11.5.1.1.10), and proxy information (see 13.11.4.2).

Path selection with collocated mesh STAs using HWMP is described in 13.10.5.

13.12 Intra-mesh congestion control

13.12.1 General

Intra-mesh congestion control is based on the following three main mechanisms:

a) Local congestion monitoring and congestion detection
b) Congestion control signaling
c) Local rate control

A mesh STA shall activate a congestion control protocol specified by dot11MeshActiveCongestionControlMode. At any given time, there is only one congestion control protocol active in a particular MBSS, signalled in the Congestion Control Mode Identifier field of the Mesh Configuration element. This standard specifies the congestion control signaling protocol that shall be available in any MBSS with an activated congestion control.

NOTE—This standard allows for inclusion of more advanced or alternative congestion control schemes through the Congestion Control Mode Identifier in the Mesh Configuration element.

13.12.2 Congestion control signaling protocol

When dot11MeshActiveCongestionControlMode is congestionControlSignaling (1), the mesh STA activates the congestion control signaling protocol. The congestion control signaling protocol specifies the signaling messages used with intra-mesh congestion control. Specific algorithms for local congestion monitoring and congestion detection are beyond the scope of the congestion control signaling protocol.

The congestion control signaling protocol is triggered after congestion is detected at a mesh STA. A mesh STA that detects congestion, and the traffic destination causing this congestion, may transmit a Congestion Control Notification frame to the mesh STAs of the traffic source and other neighboring mesh STAs. The frame contains one or more Congestion Notification elements, each of which specifies the traffic destination causing the congestion and the expected duration of the congestion per AC per mesh destination as estimated by the congested mesh STA.

Upon receipt of a Congestion Notification frame a mesh STA may stop forwarding, or reduce the rate of forwarding, traffic to the destinations listed in the Congestion Notification elements via the mesh STA reporting congestion for the duration specified in the Congestion Notification element. It may also send its own Congestion Control Notification frame to mesh STAs that are the source of the reported congestion, and other neighboring mesh STAs. Any time difference between receipt of the original Congestion Control Notification frame and the transmission of this new Congestion Control Notification frame should be reflected in the duration indicated in the new congestion control notification in such a way that any timers set by mesh STAs in response to the first report of congestion for a given destination all expire at the same time.

If the Destination MAC Address field in a received Congestion Notification element is the group address, it should be interpreted to mean that communication with the transmitter of this frame should be stopped, or reduced, for the duration specified in the Congestion Notification element. This event should not result in the transmission of a Congestion Notification element with a Destination MAC Address field set to the group address to any neighbor mesh STAs.

When the duration of a traffic congestion report has expired, a mesh STA should resume forwarding traffic to the destinations that were listed in the traffic congestion report via the mesh STA that reported congestion.

NOTE 1—Local policies/mechanisms implemented in a mesh STA might be required to ensure timely transmission of the congestion control signaling messages and to avoid transmission of stale messages that might reduce network efficiency.

NOTE 2—A mesh STA that receives a Congestion Control Notification frame might choose to adjust its frame rate, defined by the number of transmitted frames per a unit of time, to the sender of the Congestion Control Notification frame in the identified congested AC(s) for the duration specified in the Congestion Notification element. The reduction of the frame rate to a congested mesh STA avoids waste of the mesh resources for transmission of packets that with high probability will not be handled/forwarded by the congested mesh STA.

13.13 Synchronization and beaconing in MBSSs

13.13.1 TSF for MBSSs

A mesh STA shall initialize and update its TSF timer according to the MBSS's active synchronization method. Each mesh STA shall maintain a TSF timer as described in 10.1.3.1, and conform to the TSF timer accuracy as described in 10.1.3.7.

13.13.2 Extensible synchronization framework

13.13.2.1 General

This standard introduces an extensible framework to enable the implementation of multiple synchronization methods for mesh STAs. Within the extensible synchronization framework, the neighbor offset synchronization method is defined as the default mandatory synchronization method in order to enable minimal synchronization capabilities and interoperability between mesh STAs that use MCCA, MBCA, or operate in light or deep sleep mode. The framework allows the integration of other synchronization methods into MBSSs. A vendor may implement any synchronization method using this framework to meet special application needs. Although a mesh STA may include multiple implementations of synchronisation methods, only one synchronization method at a time shall be used by a mesh STA. All mesh STAs in an MBSS use the same synchronisation method; see 13.2.7 item a). The MBSS's active synchronization method is controlled by the SME and given to the MLME by dot11MeshActiveSynchronizationMethod.

Mesh STAs shall announce the MBSS's active synchronization method using the Synchronization Method Identifier field in the Mesh Configuration element in their Beacon and Probe Response frames.

13.13.2.2 Neighbor offset synchronization method

13.13.2.2.1 General

When dot11MeshActiveSynchronizationMethod is neighborOffsetSynchronization (1), the mesh STA shall use the neighbor offset synchronization method as its active synchronization method, and maintain the timing offset value between its own TSF timer and the TSF timer of each neighbor STA with which it synchronizes. The mesh STA shall set the Synchronization Method Identifier field in the Mesh Configuration element to 1.

The mesh STA shall maintain synchronization with all of its neighbor peer mesh STAs up to dot11MeshNbrOffsetMaxNeighbor mesh STAs. The mesh STA should maintain synchronization with additional neighbor mesh STAs that are in the same MBSS up to a total of dot11MeshNbrOffsetMaxNeighbor mesh STAs and also, additional neighbor mesh STAs that are outside of the MBSS up to a total of dot11MeshNbrOffsetMaxNeighbor mesh STAs.

Upon receipt of an MLME-MESHNEIGHBOROFFSETSYNCSTART.request primitive, the MLME shall start synchronization using the neighbor offset synchronization method with the specified peer STA. Upon receipt of an MLME-MESHNEIGHBOROFFSETSYNCSTOP.request primitive, the MLME shall stop synchronization using the neighbor offset synchronization method with the specified peer STA.

A mesh STA that utilizes the neighbor offset synchronization method may start its TSF timer independently of other mesh STAs. The mesh STA shall calculate the timing offset value with respect to the neighbor STA with which it maintains synchronization, as described in 13.13.2.2.2. The mesh STA shall adjust its TSF timer based on time stamps received in Beacon or Probe Response frames from neighbor STAs with which it maintains synchronization, as described in 13.13.2.2.3.

When the mesh STA alternates Awake state and Doze state, it might not always listen to the Beacon frames of a neighbor mesh STA with which it maintains synchronization. However, it shall conform to the clock drift compensation procedures and TSF jitter allowance as described in 13.13.2.2.3. See W.3.6 for more guidelines.

13.13.2.2.2 Timing offset calculation

When dot11MeshActiveSynchronizationMethod is neighborOffsetSynchronization (1), the mesh STA shall calculate the timing offset value with respect to the neighbor STA with which it maintains synchronization. The calculation of the timing offset value is based on time stamps from the received Beacon and Probe Response frames as follows:

$$T_{offset} = T_t - T_r$$

where

T_{offset} is the timing offset value
T_t is the value in the Timestamp field in the received frame
T_r is the frame reception time measured in the TSF timer of the mesh STA

The offset value is represented as a signed integer. The unit of the offset value is µs. The mesh STA shall keep the T_{offset} value calculated from the latest Beacon or Probe Response frame received from each neighbor STA with which it maintains synchronization.

A mesh STA may translate the time measured in the TSF of the neighbor STA into the time base of its own TSF as follows:

$$T_{self} = T_{neighbor} - T_{offset}$$

where

T_{self} is the translated time in its own TSF
$T_{neighbor}$ is the time measured in the TSF timer of the neighbor STA

Upon receipt of an MLME-MESHNEIGHBOROFFSETCALCULATE.request primitive, the MLME shall receive a Beacon or Probe Response frame from the specified neighbor STA, calculate the T_{offset} from the received frame, and report the calculated T_{offset} to the SME by responding with an MLME-MESHNEIGHBOROFFSETCALCULATE.confirm primitive. T_{offset} is used to provide the timing reference of neighbor STAs.

13.13.2.2.3 Clock drift adjustment

When dot11MeshActiveSynchronizationMethod is neighborOffsetSynchronization (1), the mesh STA shall examine the reception time of the Beacon frames from neighbor STAs with which it maintains synchronization and adjust its TSF timer to compensate the relative timing error among neighbor mesh STAs caused by the clock drift. The mesh STA adjusts its TSF so that its TSF counting is aligned to the most delayed neighbor STA.

When the mesh STA receives a Beacon frame or a Probe Response frame from one of the neighbor STAs with which it maintains synchronization, the mesh STA shall perform the following measurement procedure:

a) The mesh STA checks if the transmitter of the Beacon frame or Probe Response frame is in the process of the TBTT adjustment (see 13.13.4.4.3). If the received frame contains the Mesh Configuration element and the TBTT Adjusting subfield in the Mesh Configuration field is 1, the mesh STA shall invalidate the T_{offset} value for this neighbor STA and shall not perform the following steps.

b) The mesh STA checks if it has a valid T_{offset} value obtained from the previous Beacon or Probe Response frame reception from the transmitter of the received frame. If it does not have the valid T_{offset} value, it shall not perform the following steps.

c) The mesh STA calculates the clock drift amount $T_{ClockDrift}$ by comparing the $T_{offset,p}$,the offset value obtained previously for this neighbor STA, and the $T_{offset,c}$, the offset value obtained from the current frame reception.

   $$T_{ClockDrift} = T_{offset,\,p} - T_{offset,\,c}$$

   where

   $T_{ClockDrift}$ is the clock drift amount in µs represented as a signed integer.

d) The mesh STA shall compare the $T_{ClockDrift}$ value with the $T_{MaxClockDrift}$, the largest $T_{ClockDrift}$ value obtained from other neighbor STA within this beacon period. If the $T_{ClockDrift}$ value for this neighbor STA is greater than the $T_{MaxClockDrift}$ value, the mesh STA replaces the $T_{MaxClockDrift}$ value with the $T_{ClockDrift}$ value, in order to determine the largest $T_{ClockDrift}$ value among neighbor STAs.

When the previous $T_{ClockDrift}$ values have been stable for a neighbor mesh STA, the mesh STA may substitute the previous $T_{ClockDrift}$ value for the $T_{ClockDrift}$ value in the measurement procedure and process the step d), at the time of a TBTT of the neighbor STA, without receiving a Beacon frame.

Before the mesh STA transmits a Beacon frame, it shall perform the following adjustment procedure:

— The mesh STA checks if the current $T_{MaxClockDrift}$ value is greater than zero. If the $T_{MaxClockDrift}$ value is greater than zero, it shall continue the following steps. Otherwise, it shall initialize the $T_{MaxClockDrift}$ with zero and shall not perform the following step.

— If the $T_{MaxClockDrift}$ value is smaller than 0.04% of its beacon interval, the mesh STA shall adjust its TSF timer so that the next TBTT will be delayed for the duration of the $T_{MaxClockDrift}$ and initialize the $T_{MaxClockDrift}$ value with zero. Otherwise, it shall adjust its TSF timer so that the next TBTT will be delayed for the duration of 0.04% of its beacon interval and subtract the value of 0.04% of its beacon interval from the $T_{MaxClockDrift}$.

The mesh STA may adjust its TSF timer only to slow the counting. The mesh STA may adjust its TSF timer within the range of 0.04% in a beacon period.

When the delay amount at each beacon period is not stable, the mesh STA should frequently listen to neighbor STAs' Beacon frames. An implementation might reduce TSF timer jitter caused by the adjustment procedure by making additional adjustments to the $T_{MaxClockDrift}$, as long as the mesh STA's TSF count is aligned to the most delayed neighbor mesh STA. The means of making these additional adjustments is beyond the scope of this standard.

NOTE—This clock drift compensation procedure does not intend to maintain a strict synchronization. It aims to stop TBTT drifting away among neighbor mesh STAs, allowing some jitter of TSF timer.

13.13.3 Beaconing

13.13.3.1 Beacon generation in MBSSs

A mesh STA transmits Beacon frames that are specific to an MBSS. Beacon frames for MBSS, infrastructure BSS, or IBSS are differentiated by the Capability Information field in the Beacon frame as specified in 8.4.1.4. A mesh STA that collocates with an AP generates Beacon frames for the MBSS independently of the AP.

The mesh STA shall define a series of TBTTs exactly dot11BeaconPeriod TUs apart. Time zero is defined to be a TBTT with the Beacon frame containing a DTIM. At each TBTT, the mesh STA shall schedule a Beacon frame as the next frame for transmission according to the medium access rules specified in Clause 9. The beacon period is included in Beacon and Probe Response frames.

The mesh STA shall start beaconing upon the receipt of the MLME-START.request primitive.

13.13.3.2 Beacon reception for mesh STA

A mesh STA shall use information from the Timestamp field without regard to the BSSID or Mesh ID in order to obtain information necessary for synchronization, if the mesh STA maintains synchronization with the transmitter of the Beacon frame. A mesh STA may use information from the Beacon interval field and the Beacon Timing element without regard for the Mesh ID in order to obtain information necessary for MBCA, if the mesh STA maintains synchronization with the transmitter of the Beacon frame and dot11MBCAActivated is true. A mesh STA may use information from the MCCAOP Advertisement Overview element and MCCAOP Advertisement element without regard for the Mesh ID in order to obtain information necessary for MCCA, if the mesh STA maintains synchronization with the transmitter of the Beacon frame and dot11MCCAActivated is true.

A mesh STA in a mesh BSS shall use information that is not in the CF Parameter Set element, the Timestamp field, the Beacon interval field, the Beacon Timing element, the MCCAOP Advertisement Overview element, or the MCCAOP Advertisement element in received Beacon frames only if the mesh STA maintains a mesh peering with the transmitter of the Beacon frame.

13.13.4 Mesh beacon collision avoidance (MBCA)

13.13.4.1 Overview

Mesh STAs use the mesh beacon collision avoidance (MBCA) protocol to detect and mitigate collisions among Beacon frames transmitted by other STAs (including mesh STAs, APs, and STAs in an IBSS) on the same channel within the range of 2 hops. MBCA mitigates hidden node problems with respect to Beacon frames.

NOTE—Beacon frames are transmitted without acknowledgement and might collide with other frames. In a mesh BSS, multiple STAs transmit Beacon frames periodically, and mesh STAs might be located out of range of each other. This implies that Beacon frames might suffer from the so-called hidden node problem and might not be received by neighbor STAs. Once Beacon frames from hidden STAs start to collide, Beacon frames keep on colliding if these hidden STAs transmit Beacon frames at the same beacon interval that is a typical operation. MBCA provides a set of rules to mitigate this problem.

When dot11MBCAActivated is true, the mesh STA shall set the MBCA Enabled subfield in the Mesh Capability field of the Mesh Configuration element to 1.

MBCA is composed of beacon timing advertisements, TBTT selection, and TBTT adjustment. When dot11MBCAActivated is true, the mesh STA advertises the TBTT and beacon interval of its neighbor STAs through the Beacon Timing element as described in 13.13.4.2. Upon reception of the Beacon Timing element, the mesh STA obtains the beacon timing information of its neighbor mesh STAs and uses this information for its TBTT selection and TBTT adjustment as described in 13.13.4.3 and 13.13.4.4. The mesh STA may also perform additional procedures described in 13.13.4.5 and 13.13.4.6.

When dot11MBCAActivated is true, the mesh STA that alternates Awake state and Doze state should listen to Beacon frames from its neighbor STAs, with which it maintains synchronization, often, in order to advertise and obtain the recent TBTT information.

13.13.4.2 Beacon timing advertisement

13.13.4.2.1 General

When dot11MBCAActivated is true, the mesh STA shall contain Beacon Timing element in Beacon and Probe Response frames in order to advertise its beacon timing information. The mesh STA calculates the TBTT of its neighbor STAs with which it maintains synchronization as described in 13.13.4.2.2, and composes beacon timing information as described in 13.13.4.2.3. The mesh STA collects the beacon timing information from each neighbor STA with which it maintains synchronization. The collection of the beacon timing information is termed “beacon timing information set.” The mesh STA contains whole or part of the beacon timing information set in the Beacon Timing element as described in 13.13.4.2.5. The mesh STA also maintains the status number of the beacon timing information set and contains the status number in the Beacon Timing element as described in 13.13.4.2.4. The receiver of the Beacon Timing element uses the received beacon timing information as described in 13.13.4.2.6.

13.13.4.2.2 Calculation of neighbor STA’s TBTT

When a Beacon frame is received from one of its neighbor STAs with which the mesh STA maintains synchronization, the mesh STA shall calculate the TBTT of the received Beacon frame as follows:

$$T_{TBTT} = T_r - (T_t \text{ modulo } (T_{BeaconInterval} \times 1024))$$

where

T_{TBTT} is the calculated TBTT
T_r is the frame reception time measured in the TSF timer of the receiving mesh STA
T_t is the value in the Timestamp field in the received frame
$T_{BeaconInterval}$ is the value in the Beacon interval field in the received frame

The T_{TBTT} is used as described in 13.13.4.2.3.

Further, the mesh STA shall calculate the time difference between the TBTT of the received Beacon frame and the time predicted from the past TBTT as follows:

$$T_{Delta} = |\, T_{TBTT,\,c} - (T_{TBTT,\,p} + (T_{BeaconInterval} \times N_{Count}))\, |$$

where

T_{Delta} is the time difference
$T_{TBTT,\,c}$ is the TBTT calculated from the received Beacon frame
$T_{TBTT,\,p}$ is the TBTT calculated for the first time after the latest status number update (see 13.13.4.2.4)
$T_{BeaconInterval}$ is the value in the Beacon interval field in the received Beacon frame
N_{Count} is the number of TBTTs since $T_{TBTT,\,p}$ has been calculated

T_{Delta} is used to maintain the status number described in 13.13.4.2.4.

13.13.4.2.3 Beacon timing information

The mesh STA shall keep the latest T_{TBTT} together with the Beacon interval contained in the received frame and the identifier of the neighbor STA as the beacon timing information with respect to the neighbor STA. When the elapsed time since the latest Beacon frame reception is smaller than 524 288 TU, the beacon timing information is valid.

NOTE—The beacon timing information provides the time reference for a series of the TBTTs of the corresponding STA. Using the beacon timing information, a mesh STA is able to predict future TBTTs by adding the reported beacon interval to the reported TBTT.

The mesh STA shall collect the valid beacon timing information from each neighbor STA with which it maintains synchronization and keep the collection as the beacon timing information set. The beacon timing information set is advertised to its neighbor mesh STAs through the Beacon Timing element as described in 13.13.4.2.5.

When the amount of neighbors, for which valid beacon timing information is kept, is large, the beacon timing information set may be divided into multiple tuples of beacon timing information. In such case, a tuple of beacon timing information is included in the Beacon Timing element (see 13.13.4.2.5).

13.13.4.2.4 Maintenance of the status number

The mesh STA shall maintain the status number of the beacon timing information set. The status number is set to a value from a single modulo-16 counter, starting at 0 and incrementing by 1 for each transmission of a frame containing the Beacon Timing element after the mesh STA encountered any of the following events:

a) It starts or stops maintaining synchronization with a neighbor STA.
b) It receives a Beacon frame from a neighbor STA with which it maintains synchronization and the calculated T_{Delta} (see 13.13.4.2.2) is larger than 255 µs.
c) It completes the TBTT adjustment procedure described in 13.13.4.4.3.

The mesh STA shall set the Status Number subfield in the Report Control field in the Beacon Timing element to the status number. The Status Number subfield in the Report Control field facilitates the detection of the changes in the beacon timing information set.

13.13.4.2.5 Transmitter's procedure

When dot11MBCAActivated is true, the mesh STA shall report the TBTT and beacon interval of its neighbor STAs through the Beacon Timing element as described in this subclause.

The Beacon Timing element reports on timing information of the Beacon frames that are received from the neighbor STAs with which the mesh STA maintains synchronization on the operating channel. The mesh STA shall include the Beacon Timing element in Probe Response frames and in TBTT Adjustment Request frames. The mesh STA shall also include the Beacon Timing element in Beacon frames as specified by dot11MeshBeaconTimingReportInterval and dot11MeshBeaconTimingReportMaxNum. The Beacon Timing element is present in a Beacon frame when the DTIM Count value in the Beacon frame is zero or equal to an integer multiple of dot11MeshBeaconTimingReportInterval.

The maximum number of Beacon Timing Information fields contained in a Beacon Timing element is limited to dot11MeshBeaconTimingReportMaxNum for Beacon frames, or is limited by the maximum element size for other frames. When the number of neighbors, for which valid beacon timing information is kept, is equal or smaller than the limit, the mesh STA shall include all the beacon timing information in a single Beacon Timing element, setting both Beacon Timing Element Number and More Beacon Timing Elements subfield in the Report Control field to 0. When the number of neighbors, for which valid beacon timing information is kept, exceeds the limit, the mesh STA shall divide the beacon timing information set into multiple tuples and assign each tuple with an index number starting from 0. When the beacon timing information set is divided, the mesh STA shall include the successive tuples of beacon timing information in the Beacon Timing elements. In this case, the mesh STA shall set the Beacon Timing Element Number subfield in the Report Control field to the index number of the tuple. The mesh STA shall set the More Beacon Timing Elements subfield in the Report Control field to 1 when it has one or more beacon timing information tuples with a larger index number. The mesh STA shall divide the beacon timing information set into no more than N_Info tuples, where N_Info = Ceiling(number of valid beacon timing information / maximum number of Beacon Timing Information fields in the Beacon Timing element).

The mesh STA may update the combination of the tuples only when the status number described in 13.13.4.2.4 is updated. The mesh STA shall include newly updated beacon timing information (i.e., beacon timing information that causes an update of the status number as described in 13.13.4.2.4) in the tuple with a smaller index number. When the status number is updated, the mesh STA shall include the tuple of beacon timing information indexed as 0 in the Beacon Timing element in the subsequent Beacon frame. Successive tuples shall be transmitted in ascending order of the index number in the successive Beacon frames.

NOTE—The standard does not impose mesh STAs to advertise a fragmented beacon timing information set sequentially in its Beacon frames at all times. This implies that the mesh STA might advertise tuples with a smaller index number more frequently, which is useful to notify new beacon timing information efficiently.

When the mesh STA receives a Probe Request frame containing a Beacon Timing element ID in its Request element, it shall respond with a Probe Response frame containing the Beacon Timing element. If all beacon timing information cannot be contained in a Beacon Timing element, the mesh STA shall include multiple Beacon Timing elements containing successive tuples of beacon timing information in the order of the Request element (see Table 8-27) so that all tuples are transmitted.

13.13.4.2.6 Receiver's procedure

A mesh STA with dot11MBCAActivated equal to true that receives a Beacon Timing element obtains the beacon timing information of its neighbor mesh STA and uses it for its TBTT selection and TBTT adjustment as described in 13.13.4.3 and 13.13.4.4.

When a mesh STA receives a Beacon frame with a Beacon Timing element that contains only a subset of the beacon timing information set, the mesh STA may transmit a Probe Request frame containing a Beacon Timing element ID in its Request element to the transmitter of the Beacon Timing element, in order to request the rest of the beacon timing information.

NOTE 1—The Report Control field in the Beacon Timing element facilitates the detection of the missing beacon timing information.

NOTE 2—Once the entire beacon timing information set with a particular Status Number is obtained, the mesh STA does not need to retrieve beacon timing information as long as the Status Number remains the same.

A mesh STA that receives the Beacon Timing element shall record the reported TBTT and its successive TBTTs as neighbor's essential beacon reception timing if the MSB of the Neighbor STA ID field in the corresponding Beacon Timing Information field is 0. The essential beacon reception timing is used to control the transmission of frames as described in 13.13.4.5.

A mesh STA can also check if its neighbor mesh STAs received its Beacon frame successfully by checking whether the Beacon Timing elements received from its neighbor mesh STAs contain beacon timing information of the mesh STA. When the Beacon Timing element is received from one of the peer mesh STAs, the mesh STA checks if the MSB of the Neighbor STA ID subfield is set to 0 and the rest of the field matches with the 7 LSBs of the AID value assigned to the mesh STA through the mesh peering establishment. When the Beacon Timing element is received from a nonpeer mesh STA, the mesh STA checks if the MSB of the Neighbor STA ID subfield is set to 1 and the rest of the field matches with the 7 LSBs of its own MAC address (taking the I/G bit as the MSB). If the matching is verified, the corresponding beacon timing information represents the correct beacon reception by the neighbor mesh STA.

If a Beacon frame is received from a neighbor peer mesh STA that is either in active mode or in light sleep mode, the Beacon Timing element is present in the frame, and all beacon timing information is contained in the Beacon Timing element, the mesh STA shall verify whether the neighbor peer mesh STA received its Beacon frame. If the Beacon Timing element does not contain beacon timing information of the mesh STA or the Neighbor TBTT subfield of the corresponding beacon timing information does not reflect the recent TBTT of the mesh STA, the mesh STA considers the previous Beacon frame was not received by the neighbor peer mesh STA.

13.13.4.3 TBTT selection

When dot11MBCAActivated is true, the mesh STA performs the TBTT selection described herein before it starts beaconing (see 13.2.8). The mesh STA selects its TBTTs and its beacon interval so that its Beacon frames do not collide with Beacon frames transmitted by other STAs in its 2 hop range.

Before the mesh STA starts beaconing, it performs scanning and discovered neighbor STAs are reported through an MLME-SCAN.confirm primitive (see 13.2.6). Using TimeStamp, Local Time, and Beacon Period in the BSSDescription parameter provided by the MLME-SCAN.confirm primitive, the mesh STA shall obtain the TBTT and beacon interval of its neighbor STAs operating on the same channel as the mesh STA starts to operate. The mesh STA shall also collect the beacon timing information contained in the Beacon Timing elements received on the channel through Beacon Timing in the BSSDescription parameter provided by the MLME-SCAN.confirm primitive, in order to obtain the TBTT and beacon interval of STAs in 2 hop range. After obtaining this information, the mesh STA shall look for a timing of its beacon transmissions so that its Beacon frames are likely not to collide with Beacon frames transmitted by other STAs in its 2 hop range. The mesh STA shall update its TSF timer and select its beacon interval to set its TBTTs to the appropriate timing, and then it shall start beaconing using the MLME-START.request primitive.

13.13.4.4 TBTT adjustment

13.13.4.4.1 Self-determined TBTT adjustment

When dot11MBCAActivated is true, the mesh STA checks if it does not transmit Beacon frames during the beacon transmissions of other STAs within its 2 hop range using the Beacon Timing element received from its neighbor peer mesh STA.

When the mesh STA discovers that its Beacon frames repeatedly collide with the Beacon frames of a neighbor or a neighbor's neighbor and its TBTT comes later than the TBTT of the colliding STA at the time of collision, it shall perform the TBTT scanning procedure described in 13.13.4.4.3. If the mesh STA finds an alternative TBTT, it shall start the TBTT adjustment procedure as described in 13.13.4.4.3.

13.13.4.4.2 Requested TBTT adjustment

When a mesh STA discovers that Beacon frames from two or more neighbor STAs are colliding repeatedly or a series of TBTTs are close enough to trigger frequent beacon collisions, the mesh STA may transmit a TBTT Adjustment Request frame to the neighbor mesh STA of which the TBTT comes last at a particular collision timing in order to request this neighbor mesh STA to adjust its TBTT. The TBTT Adjustment Request frame may be transmitted only if the following conditions hold:

- The recipient of the TBTT Adjustment Request frame is a peer mesh STA and has set the MBCA Enabled subfield in the Mesh Capability field of the Mesh Configuration element to 1.
- The other colliding STA does not include the Mesh Configuration element in its Beacon frames or the TBTT Adjusting field in the Mesh Configuration element is 0.

When dot11MBCAActivated is true, the mesh STA that receives a TBTT Adjustment Request frame shall perform the TBTT scanning procedure described in 13.13.4.4.3, and determine if it can find an appropriate alternative timing for its TBTTs. After the completion of the TBTT scanning procedure, the mesh STA that receives the TBTT Adjustment Request frame shall respond with a TBTT Adjustment Response frame containing the result of the TBTT scanning in the Status Code field. If the mesh STA finds an alternative TBTT, it shall agree with the request. If it agrees with the request, the Status Code field is set to 0 in the TBTT Adjustment Response frame, and it shall complete the TBTT adjustment procedure described in 13.13.4.4.3. If it does not agree with the request, it shall indicate the reason in the Status Code field in the TBTT Adjustment Response frame. A mesh STA may set the Status Code to either 0, 1, or 78 in the TBTT Adjustment Response frame.

13.13.4.4.3 TBTT scanning and adjustment procedures

When a mesh STA is in need of TBTT adjustment, it tries to find an alternative TBTT first. The mesh STA shall perform the TBTT scanning procedure as follows:

- The mesh STA checks if its beacon timing information and collected neighbor's beacon timing information are sufficiently new. If the mesh STA did not receive a Beacon frame from a neighbor STA with which it maintains synchronization at the latest TBTT, it shall receive a Beacon or Probe Response frame from the neighbor STA and obtain the TBTT of the neighbor STA and the beacon timing information contained in the Beacon Timing element.

  NOTE—This is particularly important if the mesh STA is in deep sleep mode for a neighbor peer mesh STA.

- Using the latest TBTT of its neighbor STAs and the latest beacon timing information of neighbor mesh STAs, the mesh STA shall look for an alternative TBTT that does not cause beacon collision among the STAs in its 2 hop range.

If an alternative TBTT is not available, the mesh STA terminates the procedure. If an alternative TBTT is available, the mesh STA shall start the TBTT adjustment procedure as follows:

- The mesh STA shall set the TBTT Adjusting field in the Mesh Configuration element to 1 in order to announce that the TBTT adjustment procedure is ongoing.
- The mesh STA shall suspend its TSF timer for a period of time, no longer than half of the Group Delivery Idle Time (defined in 13.14.5) within a single beacon period, to slow its TSF.
- The mesh STA shall adjust TBTT information of the neighbor STAs (see 13.13.4.2.3), that are to be contained in the Beacon Timing element, accordingly by subtracting the delay amount.
- When dot11MCCAActivated is true, the mesh STA shall adjust the MCCAOP reservations accordingly by modifying the MCCAOP Offset of each MCCAOP reservation. See 9.20.3.3.
- The mesh STA shall repeat suspending its TSF timer over multiple beacon periods until its TBTT is set to the alternative TBTT.

— Upon completion of the TBTT adjustment, the mesh STA shall update the status number as described in 13.13.4.2.4 and shall set the TBTT Adjusting field in the Mesh Configuration element to 0.

NOTE—A mesh STA in deep sleep mode might interpret its neighbor mesh STA's TBTT adjustment as a large TSF jitter. When a mesh STA in deep sleep mode observes a large TSF jitter and the Status Number in the Report Control field in the Beacon Timing element of the received Beacon frame (or Probe Response frame) has been updated, the mesh STA in deep sleep mode should not take this jitter as clock drift and listen to the next Beacon frame to verify if the clock drift is large.

13.13.4.5 Frame transmission across reported TBTT

When dot11MBCAActivated is true, the mesh STA should not extend its transmissions across TBTT of its neighbor STAs with which it maintains synchronization. Further, the mesh STA should not extend its transmissions, other than Beacon frames, across all essential beacon reception timing (see 13.13.4.2.6) reported from its neighbor mesh STAs with which it maintains synchronization. This operation helps in reducing the hidden STA interference with beacon reception at its neighbor mesh STAs. When both dot11MBCAActivated and dot11MCCAActivated are true, the mesh STA shall not extend its transmissions across TBTT of its neighbor STAs with which it maintains synchronization. Further, the mesh STA shall not extend its transmissions, other than Beacon frames, across all beacon reception timing reported from its neighbor mesh STAs with which it maintains synchronization.

After silencing for dot11MeshAverageBeaconFrameDuration µs from the reported neighbor's TBTT, the mesh STA may start transmitting frames again.

13.13.4.6 Delayed beacon transmissions

A mesh STA may occasionally delay its Beacon frame transmission from its TBTT for a pseudorandom time. This attribute is specified by dot11MeshDelayedBeaconTxInterval, dot11MeshDelayedBeaconTxMinDelay, and dot11MeshDelayedBeaconTxMaxDelay. When dot11MeshDelayedBeaconTxInterval is set to nonzero value, the mesh STA shall delay its Beacon frame transmission from TBTT, once every dot11MeshDelayedBeaconTxInterval. When the mesh STA transmits a Beacon frame with delay from its TBTT, the delay time shall be randomly selected between dot11MeshDelayedBeaconTxMinDelay and dot11MeshDelayedBeaconTxMaxDelay µs.

NOTE—Delayed beacon transmission allows mesh STAs to discover Beacon frames that are transmitted from multiple mesh STAs with TBTTs close to each other. It is recommended to set dot11MeshDelayedBeaconTxMaxDelay to a time longer than the typical duration of Beacon frames.

13.14 Power save in a mesh BSS

13.14.1 General

A mesh STA may use mesh power modes to reduce its power consumption. A mesh STA manages each of its mesh peerings with a peer-specific mesh power mode as described in 13.14.2.2. A mesh STA may set the mesh power mode for a mesh peering independently of the mesh power modes for its other mesh peerings. A mesh STA also manages a nonpeer mesh power mode as described in 13.14.2.3. When a mesh STA is in light sleep mode or in deep sleep mode for a mesh peering, the mesh STA shall maintain its mesh awake window as described in 13.14.6.

A mesh STA shall have the capability to buffer frames and to perform mesh power mode tracking for the peer-specific mesh power modes of its peer mesh STAs, as described in 13.14.7. A mesh STA shall use mesh peer service periods for individually addressed frame transmissions to neighbor peer mesh STAs that are either in light sleep mode or in deep sleep mode towards this mesh STA, as described in 13.14.9. A mesh STA transmits group addressed frames after the Beacon frame containing DTIM when any of its peer mesh STAs is in light sleep mode or deep sleep mode for the mesh peering with the mesh STA (see 13.14.4 and

13.14.5). These capabilities are referred to as support for power save.

13.14.2 Mesh power modes

13.14.2.1 General

A mesh STA is in one of two different power states, Awake or Doze, as defined in 10.2.1.2.

The manner in which a mesh STA transitions between power states is determined by its peer-specific mesh power modes and its nonpeer mesh power mode. A mesh STA shall be in Awake state if any of the conditions specified in 13.14.8.6 is not fulfilled. A mesh STA maintains peer-specific mesh power modes for each of its mesh peerings as described in 13.14.2.2. A mesh STA may have a different peer-specific mesh power mode for each mesh peering. A mesh STA maintains a nonpeer mesh power mode for nonpeer mesh STAs that is described in 13.14.2.3. An example illustration of the use of peer specific and nonpeer mesh power modes is shown in Figure 13-5.

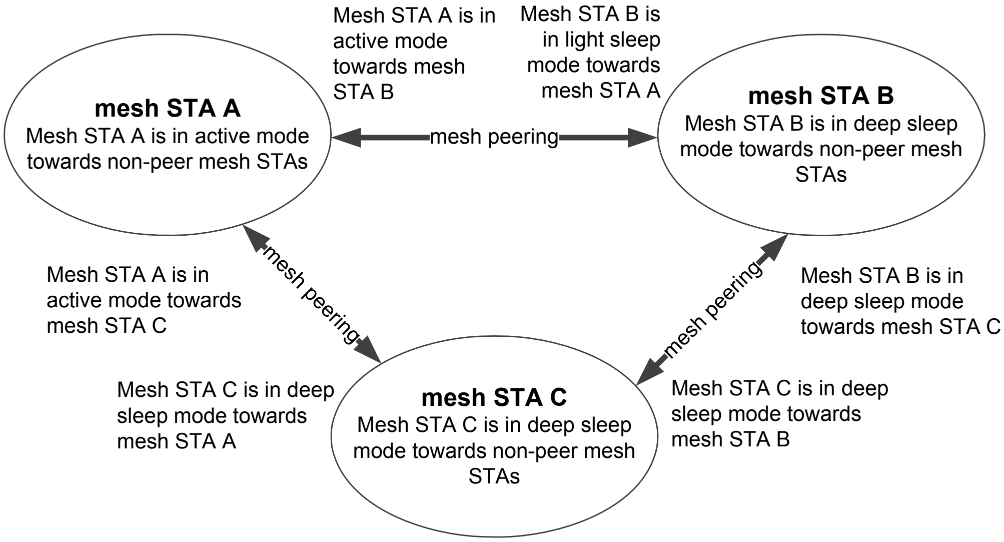

Mesh STA A is in active mode towards mesh STA B
Mesh STA B is in light sleep mode towards mesh STA A
mesh STA A
Mesh STA A is in active mode towards non-peer mesh STAs
mesh peering
mesh STA B
Mesh STA B is in deep sleep mode towards non-peer mesh STAs
mesh peering
Mesh STA A is in active mode towards mesh STA C
mesh peering
Mesh STA B is in deep sleep mode towards mesh STA C
Mesh STA C is in deep sleep mode towards mesh STA A
mesh STA C
Mesh STA C is in deep sleep mode towards non-peer mesh STAs
Mesh STA C is in deep sleep mode towards mesh STA B

Figure 13-5—An example of mesh power mode usage

13.14.2.2 Peer-specific mesh power modes

The peer-specific mesh power mode specifies the activity level of the mesh STA for the corresponding mesh peering. Three mesh power modes are defined: active mode, light sleep mode, and deep sleep mode. The peer-specific mesh power modes are defined as follows:

- *Active mode:* The mesh STA shall be in Awake state all the time.
- *Light sleep mode:* The mesh STA alternates between Awake and Doze states, as specified in 13.14.8.4. The mesh STA shall listen to all the Beacon frames from the corresponding peer mesh STA.
- *Deep sleep mode:* The mesh STA alternates between Awake and Doze states, as specified in 13.14.8.5. The mesh STA may choose not to listen to the Beacon frames from the corresponding peer mesh STA.

The combination of the Power Management field in the Frame Control field and the Mesh Power Save Level subfield in the QoS Control field contained in Mesh Data frames indicates the peer-specific mesh power mode as shown in the Table 13-31.

Table 13-31—Peer-specific mesh power mode definition

Activity level	Peer-specific mesh power mode	Power Management field	Mesh Power Save Level subfield
Highest	Active mode	0	Reserved
↑	Light sleep mode	1	0
Lowest	Deep sleep mode	1	1

13.14.2.3 Nonpeer mesh power modes

The nonpeer mesh power mode indicates the mesh power mode of the mesh STA toward the nonpeer mesh STAs. Two nonpeer mesh power modes are defined: active mode and deep sleep mode. The nonpeer mesh power mode is indicated by the Power Management field in the Frame Control field in group addressed frames, management frames transmitted to nonpeer neighbor STAs, and in Probe Response frames. When the Power Management field in the Frame Control field is set to 1, the nonpeer mesh power mode is deep sleep mode. When the Power Management field in the Frame Control field is set to 0, the nonpeer mesh power mode is active mode.

A mesh STA may send Probe Request and Mesh Peering Open Request frames to a nonpeer mesh STA that sets its nonpeer mesh power mode to deep sleep mode only during the mesh awake window of the mesh STA.

13.14.3 Mesh power mode indications and transitions

13.14.3.1 General

When a mesh STA is in active mode for a mesh peering, it shall set the Power Management field in the Frame Control field to 0 in all individually addressed Mesh Data or QoS Null frames transmitted to the corresponding peer mesh STA.

When a mesh STA is in light sleep mode for a mesh peering, it shall set the Power Management field in the Frame Control field to 1 and the Mesh Power Save Level subfield in the QoS Control field to 0 in all individually addressed Mesh Data or QoS Null frames transmitted to the corresponding peer mesh STA.

When a mesh STA is in deep sleep mode for a mesh peering, it shall set the Power Management field in the Frame Control field to 1 and the Mesh Power Save Level subfield in the QoS Control field to 1 in all individually addressed Mesh Data or QoS Null frames transmitted to the corresponding mesh STA.

When a mesh STA is in deep sleep mode for any of its mesh peerings, the Mesh Power Save Level subfield in the QoS Control field in group addressed Mesh Data frames and the Mesh Power Save Level subfield in the Mesh Capability field in the Mesh Configuration element shall be set to 1. When a mesh STA is not in deep sleep mode for any of its mesh peerings, these subfields shall be set to 1.

To change peer-specific mesh power modes, a mesh STA shall inform its peer mesh STAs through a successful frame exchange initiated by the mesh STA. The Power Management field in the Frame Control field and the Mesh Power Save Level subfield in the QoS Control field of the frame sent by the mesh STA in this exchange indicates the peer-specific mesh power mode that the STA shall adopt upon successful completion of the entire frame exchange.

The algorithm to trigger the change of a peer-specific mesh power mode is beyond the scope of this standard.

The nonpeer mesh power mode is determined by the peer-specific mesh power modes of the mesh STA. When a mesh STA is in light sleep mode or deep sleep mode for at least one mesh peering, it shall set the nonpeer mesh power mode to deep sleep mode.

When a mesh STA is in active mode for nonpeer STAs, it shall set the Power Management field in the Frame Control field to 0 in group addressed frames, in management frames transmitted to nonpeer mesh STAs, and in Probe Response frames.

When a mesh STA is in deep sleep mode for nonpeer STAs, it shall set the Power Management field in the Frame Control field to 1 in group addressed frames, in management frames transmitted to nonpeer mesh STAs, and in Probe Response frames.

13.14.3.2 Transition to a higher activity level

A mesh STA may use group addressed or individually addressed Mesh Data or QoS Null frames to change its mesh power mode to a higher activity level, for example; from deep sleep to light sleep or to active mode; or from light sleep to active mode.

Individually addressed frames may be used to temporarily raise the activity level of the mesh STA for a mesh peering. This is useful in cases when a link temporarily requires efficient data transmission with the peer mesh STA and the mesh STA desires to be able to transit back to lower activity level without performing the mesh power mode transition signaling with all peer mesh STAs.

13.14.3.3 Transition to a lower activity level

A mesh STA shall use acknowledged individually addressed Mesh Data or QoS Null frames to change its peer-specific mesh power mode to a lower activity level, for example; from active mode to light or deep sleep mode; or from light sleep to deep sleep mode.

13.14.4 TIM transmissions in an MBSS

The TIM element identifies the peer mesh STAs for which traffic is pending and buffered in the reporting mesh STA. This information is coded in a partial virtual bitmap, as described in 8.4.2.7. In addition, the TIM contains an indication whether group addressed traffic is pending. Every neighbor peer mesh STA is assigned an AID by the reporting mesh STA as part of the mesh peering establishment process (see 13.3.1). The mesh STA shall identify those peer mesh STAs for which it is prepared to deliver buffered MSDUs and MMPDUs by setting bits in the TIM's partial virtual bitmap that correspond to the appropriate AIDs.

13.14.5 TIM types

There are two different TIM types: TIM and DTIM. A mesh STA shall transmit a TIM with every Beacon frame. Every DTIMPeriod, a TIM of type DTIM is transmitted with a Beacon frame. After transmitting a Beacon containing a DTIM, the mesh STA shall send the buffered group addressed MSDUs and MMPDUs, before transmitting any individually addressed frames. The More Data field of each group addressed frame shall be set to indicate the presence of further buffered group addressed MSDUs and MMPDUs. The mesh STA sets the More Data field to 0 in the last transmitted group addressed frame following the transmission of the DTIM Beacon.

When a mesh STA expects to receive a group addressed frame and CCA is IDLE for the duration of the PHY specific Group Delivery Idle Time, the receiving mesh STA may assume that no more frames destined to group addresses will be transmitted and may return to Doze state. The Group Delivery Idle Time is identical to the TXOP Limit for AC_VI specified by the default EDCA Parameter Set shown in Table 8-105.

13.14.6 Mesh awake window

A mesh STA shall be in Awake state when its mesh awake window is active. A mesh awake window is active after the Beacon and Probe Response frames containing the Mesh Awake Window element. A mesh STA shall include the Mesh Awake Window element in its DTIM Beacon frames and may include the Mesh Awake Window element in its TIM Beacon and Probe Response frames. A mesh STA that operates in light sleep mode or deep sleep mode for any of its mesh peerings shall include the Mesh Awake Window element in its Beacon frame if the Beacon frame indicates buffered traffic for at least one peer mesh STA. The start of the mesh awake window is measured from the end of the Beacon or Probe Response transmission. The duration of the mesh awake window period is specified by dot11MeshAwakeWindowDuration. A mesh STA shall set the Mesh Awake Window field in the Mesh Awake Window element to dot11MeshAwakeWindowDuration. If the Mesh Awake Window element is not contained in the Beacon frame of a mesh STA, the duration of the mesh awake window period following this beacon is zero.

If the mesh STA that has its mesh awake window active transmits frames destined to group addresses, the duration of the mesh awake window is extended by an additional PostAwakeDuration. The PostAwakeDuration follows the group address frame, and the mesh STA that has its mesh awake window active shall stay in Awake state until it has transmitted all of its group addressed frames and the PostAwakeDuration has expired. The PostAwakeDuration is equal to duration of the mesh awake window.

A mesh STA may send a frame to a peer mesh STA that is in light sleep mode or deep sleep mode for the corresponding mesh peering during the mesh awake window of this peer mesh STA. When a peer trigger frame is successfully transmitted it initiates a mesh peer service period as described in 13.14.9.

A mesh STA may send class 1 or class 2 frames, such as Probe Request or Mesh Peering Open frames, to a nonpeer mesh STA that is in deep sleep mode for nonpeer mesh STAs during the mesh awake window of this nonpeer mesh STA.

13.14.7 Power save support

As described in 13.14.2, a mesh STA indicates its peer-specific mesh power modes and performs mesh power mode tracking of the peer-specific mesh power modes of its peer mesh STAs. A mesh STA shall not arbitrarily transmit frames to mesh STAs operating in a light or deep sleep mode, but shall buffer frames and only transmit them at designated times.

A mesh STA shall only transmit frames to a mesh STA operating in a light or deep sleep mode if the recipient mesh STA is in the Awake state as defined in 13.14.8.4, 13.14.8.5, and 13.14.9; otherwise, the mesh STA shall buffer frames.

As described in 13.14.4, a mesh STA indicates the presence of buffered traffic in TIM elements for all peer mesh STAs that operate in light or deep sleep mode towards the mesh STA. The mesh STA sets the bit for AID 0 (zero) in the bit map control field of the TIM element to 1 when group addressed traffic is buffered, according to 8.4.2.7. As described in 13.14.5, a mesh STA transmits its group addressed frames after its DTIM Beacon if any of its peer mesh STA is in light or deep sleep mode towards the mesh STA.

As described in 13.14.9, mesh peer service periods are used for frame transmissions towards a mesh STA that operates in light or deep sleep mode. Mesh peer service periods are not used in frame exchanges towards active mode mesh STAs.

A mesh STA may initiate a mesh peer service period with a peer mesh STA in deep or light sleep mode by transmitting a peer trigger frame when the mesh awake window of the peer mesh STA is active.

13.14.8 Operation in peer-specific and nonpeer mesh power modes

13.14.8.1 General

Detailed operations of mesh STA in each mesh power mode are described in the following subclauses. Figure 13-6 depicts example power state transitions of mesh STAs, when three mesh STAs are in the mesh power modes shown in the Figure 13-5.

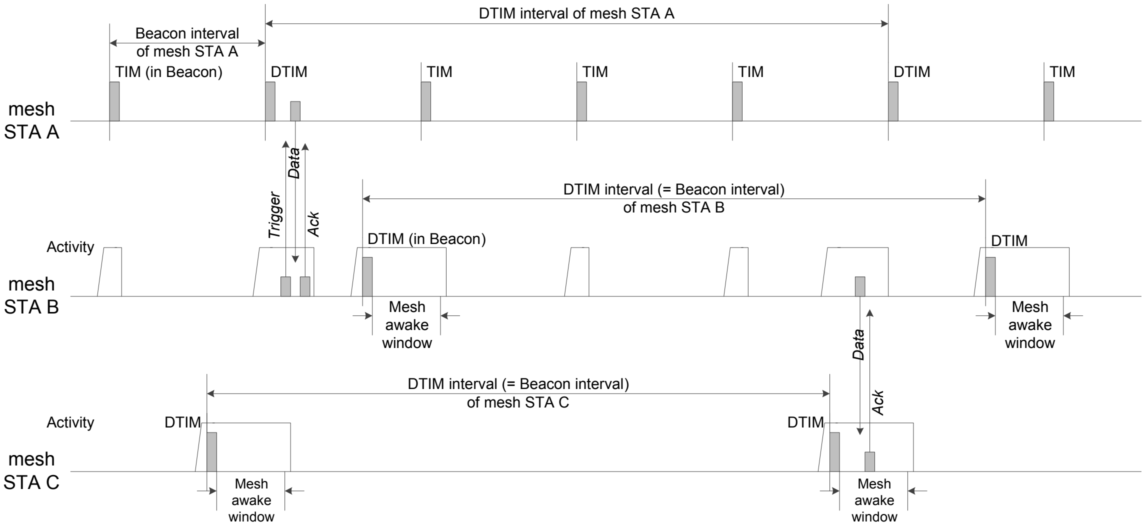


Figure 13-6—Mesh power management operation

13.14.8.2 Operation in active mode

When a mesh STA is in active mode for a mesh peering or for nonpeer mesh STAs, it shall be in Awake state. Mesh peer service periods are not used in frame exchanges towards mesh STAs that are in active mode.

An active mode mesh STA may receive peer trigger frames from a peer mesh STA in light or deep sleep mode when there is no mesh peer service period ongoing between the peer mesh STAs.

13.14.8.3 Operation in deep sleep mode for nonpeer mesh STAs

If a mesh STA is in deep sleep mode for nonpeer mesh STAs, it shall enter the Awake state prior to every TBTT of its own and shall remain in Awake state after the beacon transmission for the duration of the mesh awake window and the duration of its group addressed frame transmissions. The mesh STA may receive frames during its mesh awake window as described in 13.14.6.

When receiving a frame initiating a mesh peering management procedure, an authentication procedure, or a passive scanning procedure, a mesh STA in deep sleep mode for nonpeer mesh STAs shall operate in Awake state at least until the completion of the mesh peering management procedure (see 13.3 and 13.5), until the completion of the authentication procedure (see 13.3.1 and 13.3.3), or the transmission of the Probe Response frame.

If a mesh STA receives a peer trigger frame initiating a mesh peer service period from a peer mesh STA, the mesh STA shall remain in Awake state until the mesh peer service period is terminated as defined in 13.14.9.4.

A mesh STA may return to Doze state after its mesh awake window if no frame initiating a response transaction or a mesh peer service period is received during the mesh awake window.

13.14.8.4 Operation in light sleep mode for a mesh peering

If a mesh STA is in light sleep mode for a mesh peering, it shall enter the Awake state prior to every TBTT of the corresponding peer mesh STA to receive the Beacon frame from the peer mesh STA. The mesh STA may return to the Doze state after the beacon reception from this peer mesh STA, if the peer mesh STA did not indicate buffered individually addressed or group addressed frames. If an indication of buffered individually addressed frames is received, the light sleep mode mesh STA shall send a peer trigger frame with the RSPI field set to 1 to initiate a mesh peer service period with the mesh STA that transmitted the Beacon frame (see 13.14.9.2). If an indication of buffered group addressed frames is received, the light sleep mode mesh STA shall remain in Awake state after the DTIM Beacon reception to receive group addressed frames The mesh STA shall remain Awake state until the More Data field of a received group addressed frame is set to 0 or if no group addressed frame is received within the PHY specific Group Delivery Idle Time. (See 13.14.5.)

NOTE—When a mesh STA is in light sleep mode for a mesh peering, it sets its nonpeer mesh power mode to deep sleep mode. This implies that a mesh STA operating in light sleep mode a mesh peering is required to conform to the rules described in 13.14.8.3.

13.14.8.5 Operation in deep sleep mode for a mesh peering

A mesh STA operating in deep sleep mode for a mesh peering might not receive Beacon frames from the corresponding peer mesh STA. The logic of how the mesh STA in deep sleep mode maintains synchronization among neighbors is beyond the scope of this standard. Guidance for the synchronization maintenance by the mesh STA in deep sleep mode is given in W.3.6.

NOTE—When a mesh STA is in deep sleep mode for a mesh peering, it sets its nonpeer mesh power mode to deep sleep mode. This implies that a mesh STA operating in deep sleep mode for a mesh peering is required to conform to the rules described in 13.14.8.3.

13.14.8.6 Conditions for Doze state

A mesh STA may enter Doze state if all of the following conditions are fulfilled:

- The mesh STA operates in light sleep mode or deep sleep mode for all of its mesh peerings, as described in 13.14.8.4 or 13.14.8.5
- The mesh STA has no mesh peer service period ongoing, as described in 13.14.9
- The mesh STA has no pending transaction of mesh peering management, authentication, nor passive scanning (see 13.14.8.3)
- The mesh awake window indicated by the mesh STA has expired, as described in 13.14.6
- The mesh STA has terminated its group addressed frames delivery sequence after its DTIM Beacon, as described in 13.14.5

Guidance for using the power save in mesh BSS and default parameter values are given in W.3.

13.14.9 Mesh peer service periods

13.14.9.1 General

Mesh peer service periods are used for individually addressed frame exchanges between neighbor peer mesh STAs in which at least one of the mesh STAs is in light or deep sleep mode for the corresponding mesh peering. A mesh peer service period is a contiguous period of time during which one or more individually addressed frames are transmitted between two peer mesh STAs. Within a mesh peer service period, a mesh STA may obtain multiple TXOPs. A mesh peer service period is directional. One mesh STA is the owner of the mesh peer service period. It obtains TXOPs in order to transmit Data frames or Management frames to the recipient in the mesh peer service period. At the end of the frame transmissions, the owner of the mesh peer service period terminates the mesh peer service period. The other mesh STA operates as the recipient of the mesh peer service period and does not obtain TXOPs for transmitting Data frames or Management frames to the owner of the mesh peer service period. A mesh STA may have multiple mesh peer service periods concurrently toward multiple neighbor peer mesh STAs. At most, one mesh peer service period is set up in each direction with each peer mesh STA.

A mesh peer service period is initiated by a peer trigger frame. A peer trigger frame may initiate two mesh peer service periods. This enables both the transmitter and the receiver of the peer trigger frame to become the owner of a mesh peer service period. An example mesh peer service period between two mesh STAs in light or deep sleep mode is shown in Figure 13-7. The numbering on the left-hand-side describes the phase of the operation: 1 indicates the Initiation phase, 2 indicates the data transmission phase, and 3 indicates the termination phase of the mesh peer service period.

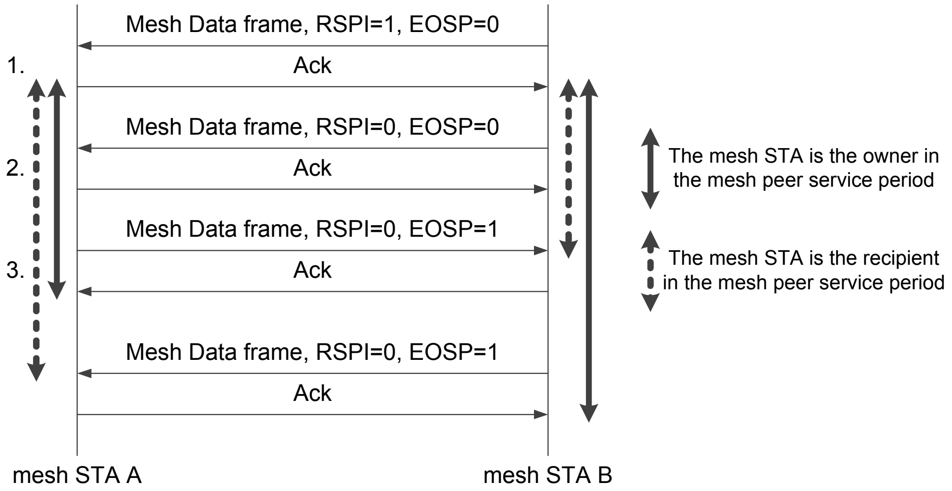

Mesh Data frame, RSPI=1, EOSP=0
1.
Ack
Mesh Data frame, RSPI=0, EOSP=0
2.
Ack
Mesh Data frame, RSPI=0, EOSP=1
3.
Ack
Mesh Data frame, RSPI=0, EOSP=1
Ack
The mesh STA is the owner in the mesh peer service period
The mesh STA is the recipient in the mesh peer service period
mesh STA A
mesh STA B

Figure 13-7—Mesh peer service period

13.14.9.2 Initiation of a mesh peer service period

A Mesh Data frame or a QoS Null frame that requires acknowledgement are used as a peer trigger frame. The RSPI and the EOSP subfields in the QoS Control field control the initiation of a mesh peer service period. Table 13-32 lists how mesh peer service periods shall be initiated with different combinations of RSPI and EOSP field values.

Mesh peer service periods are not used in frame transmissions toward active mode mesh STAs.

Table 13-32—Mesh peer service period triggering with RSPI and EOSP field combinations in peer trigger frame

RSPI	EOSP	Mesh peer service period triggering
0	0	One mesh peer service period is initiated. The transmitter of the trigger frame is the owner in the mesh peer service period.
0	1	No mesh peer service period is initiated.
1	0	Two mesh peer service periods are initiated. Both mesh STAs are owners in a mesh peer service period.
1	1	One mesh peer service period is initiated. The receiver of the trigger frame is the owner in the mesh peer service period.

The mesh peer service period may be initiated in the following cases:

— A mesh STA in light or deep sleep mode receives a peer trigger frame during its mesh awake window as described in 13.14.6

— A mesh STA in active mode receives a peer trigger frame from the peer mesh STA in light or deep sleep mode as described in 13.14.8.2

— A mesh STA receives a peer trigger frame from the peer mesh STA in light sleep mode as described in 13.14.8.4

In addition, when a mesh STA uses MCCA with a neighbor peer mesh STA while in a light sleep mode for the corresponding mesh peering, a scheduled service period begins at the each MCCAOP start time as described in 13.14.10. A mesh STA in a light or deep sleep mode shall enter the Awake state prior to the start time of scheduled service period.

13.14.9.3 Operation during a mesh peer service period

During the mesh peer service period, the owner and the recipient of the mesh peer service period shall operate in Awake state. The mesh peer service period may contain one or more TXOPs.

Reverse Direction Grant (RDG) shall not be used when the receiver of the TXOP operates in light or deep sleep mode for the link and there is no mesh peer service period ongoing toward the TXOP holder.

13.14.9.4 Termination of a mesh peer service period

The mesh peer service period is terminated after a successfully acknowledged QoS Null or Mesh Data frame with the EOSP subfield set to 1 from the owner of the mesh peer service period.

If the mesh STA does not receive an acknowledgement to a frame that requires an acknowledgement and that is sent with the EOSP subfield set to 1, the mesh STA shall retransmit that frame at least once within the same mesh peer service period—subject to applicable retry or lifetime limit. The maximum number of retransmissions within the same mesh peer service period is the lesser of the Max Retry Limit and the MIB attribute dot11MeshSTAMissingAckRetryLimit.

NOTE—If an Ack to the retransmission of this last frame in the same mesh peer service period is not received, the mesh STA might use the next mesh peer service period to further retransmit that frame subject to the applicable retry or lifetime limit.

13.14.10 MCCA use by power saving mesh STA

When dot11MCCAActivated is true and the mesh STA establishes MCCAOPs, the mesh STA shall be in active mode or light sleep mode towards the neighbor peer mesh STAs with which it has established MCCAOPs.

A scheduled mesh peer service period begins at the MCCAOP start time, if the MCCAOP responder operates in light sleep mode for the MCCAOP owner. The MCCAOP owner is the owner of the scheduled mesh peer service period. The MCCAOP responder is the recipient of the scheduled mesh peer service period. Scheduled mesh peer service periods are not used if the MCCAOP responder is in active mode for the MCCAOP owner.

The scheduled mesh peer service period continues until it is successfully terminated by the acknowledged QoS Null or Mesh Data frame with the EOSP subfield set to 1 from the owner of the mesh peer service period to the recipient of the mesh peer service period as described in 13.14.9.

14. Frequency-Hopping spread spectrum (FHSS) PHY specification for the 2.4 GHz industrial, scientific, and medical (ISM) band

14.1 Status of the Frequency Hopping PHY

The mechanisms described in this clause are obsolete. Consequently, this clause may be removed in a later revision of the standard.

14.2 Overview

14.2.1 Overview of FHSS PHY

The PHY services provided to the IEEE 802.11 WLAN MAC for the 2.4 GHz frequency-hopping spread spectrum (FHSS) system are described in this clause. The FHSS PHY consists of the following two protocol functions:

a) A PHY convergence function, which adapts the capabilities of the PMD system to the PHY service. This function is supported by the PLCP, which defines a method of mapping the IEEE 802.11 MPDUs into a framing format suitable for sending and receiving user data and management information between two or more STAs using the associated PMD system.

b) A PMD system, whose function defines the characteristics of, and method of transmitting and receiving data through, a WM between two or more STAs.

14.2.2 FHSS PHY functions

14.2.2.1 General

The 2.4 GHz FHSS PHY architecture is shown in Figure 4-14 (in 4.9). The FHSS PHY contains three functional entities: the PMD function, the PHY convergence function, and the PHY management function. Each of these functions is described in detail in 14.2.2.2 to 14.2.2.4.

The FHSS PHY service is provided to the MAC entity at the STA through a SAP, called the PHY-SAP, as shown in Figure 4-14. A set of primitives might also be defined that describe the interface between the PLCP sublayer and the PMD sublayer, called the PMD_SAP.

14.2.2.2 PLCP sublayer

To allow the IEEE 802.11 MAC to operate with minimum dependence on the PMD sublayer, a PHY convergence sublayer is defined. This function simplifies provision of a PHY service interface to the IEEE 802.11 MAC services.

14.2.2.3 PLME

The PLME performs management of the local PHY functions in conjunction with the MLME.

14.2.2.4 PMD sublayer

The PMD sublayer provides a transmission interface used to send and receive data between two or more STAs.

14.2.3 Service specification method and notation

The models represented by state diagrams in the following subclauses are intended as the primary specifications of the functions provided. It is important to distinguish, however, between a model and a real implementation. The models are optimized for simplicity and clarity of presentation, while any realistic implementation may place heavier emphasis on efficiency and suitability to a particular implementation technology.

The service of a layer or sublayer is the set of capabilities that it offers to a user in the next higher layer (or sublayer). Abstract services are specified here by describing the service primitives and parameters that characterize each service. This definition of service is independent of any particular implementation.

14.3 FHSS PHY-specific service parameter lists

14.3.1 Overview

The architecture of the IEEE 802.11 MAC is intended to be PHY independent. Some PHY implementations require medium management state machines running in the MAC sublayer in order to meet certain PMD requirements. These PHY-dependent MAC state machines reside in a sublayer defined as the MLME. The MLME in certain PMD implementations may need to interact with the PLME as part of the normal PHY-SAP primitives. These interactions are defined by the PLME parameter list currently defined in the PHY service primitives as TXVECTOR and RXVECTOR. The list of these parameters and the values they may represent are defined in the specific PHY specifications for each PMD. Subclause 14.3 addresses the TXVECTOR and RXVECTOR for the FHSS PHY.

All of the values included in the TXVECTOR or RXVECTOR described in 14.3 are considered mandatory unless otherwise specified. The 1 Mb/s and 2 Mb/s data rates are the only rates currently supported. Other indicated data rates are for possible future use.

14.3.2 TXVECTOR parameters

14.3.2.1 General

The parameters in Table 14-1 are defined as part of the TXVECTOR parameter list in the PHY-TXSTART.request primitive.

Table 14-1—TXVECTOR parameters

Parameter	Associated primitive	Value
LENGTH	PHY-TXSTART.request(TXVECTOR)	1–4095
DATARATE	PHY-TXSTART.request(TXVECTOR)	1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5

14.3.2.2 TXVECTOR LENGTH

The LENGTH parameter has the value of 1 to 4095. This parameter is used to indicate the number of octets in the MPDU that the MAC is currently requesting the PHY to transmit. This value is used by the PHY to determine the number of octet transfers that will occur between the MAC and the PHY after receiving a request to start a transmission.

14.3.2.3 TXVECTOR DATARATE

The DATARATE parameter describes the bit rate at which the PLCP should transmit the PLCP service data unit (PSDU). Its value may be any of the rates as defined in Table 14-1, and supported by the conformant FH PHY.

14.3.3 RXVECTOR parameters

14.3.3.1 General

The parameters in Table 14-2 are defined as part of the RXVECTOR parameter list in the PHY-RXSTART.indication primitive.

Table 14-2—RXVECTOR parameters

Parameter	Associatedprimitive	Value
LENGTH	PHY-RXSTART.indication(RXVECTOR)	1–4095
Receive signal strength indicator (RSSI)	PHY-RXSTART.indication(RXVECTOR)	0–RSSI Max
DATARATE	PHY-RXSTART.request(RXVECTOR)	1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5

14.3.3.2 TRXVECTOR LENGTH

The LENGTH parameter has the value of 1 to 4095. This parameter is used to indicate the value contained in the LENGTH field that the PLCP has received in the PLCP header. The MAC and PLCP use this value to determine the number of octet transfers that will occur between the two sublayers during the transfer of the received PSDU.

14.3.3.3 RXVECTOR RSSI

The RSSI is an optional parameter that has a value of 0 to RSSI Max. This parameter is a measure by the PHY of the energy observed at the antenna used to receive the current PPDU. RSSI shall be measured between the beginning of the SFD and the end of the PLCP HEC. RSSI is intended to be used in a relative manner. Absolute accuracy of the RSSI reading is not specified.

14.4 FHSS PLCP sublayer

14.4.1 Overview

Subclause 14.4 provides a convergence procedure to map MPDUs into a frame format designed for FHSS radio transceivers. The procedures for transmission, CS, and reception are defined for single and multiple antenna diversity radios.

14.4.2 State diagram notation

The operation of the procedures is described by state diagrams. Each diagram represents the domain and consists of a group of connected, mutually exclusive states. Only one state is active at any given time. Each state is represented by a rectangle as shown in Figure 14-1. These are divided into two parts by a horizontal line. In the upper part the state is identified by a name. The lower part contains the name of any signal that is generated. Actions described by short phrases are enclosed in brackets.

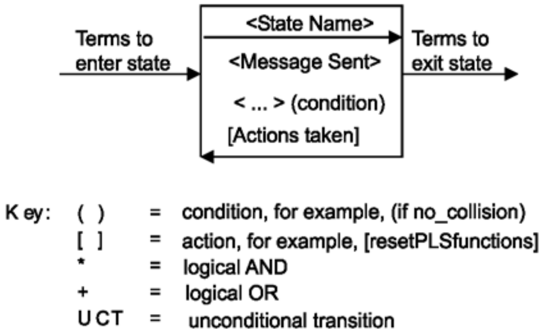

Terms to enter state
<State Name>
<Message Sent>
< ... > (condition)
[Actions taken]
Terms to exit state
Key: ( ) = condition, for example, (if no_collision)
[ ] = action, for example, [resetPLSfunctions]
* = logical AND
+ = logical OR
UCT = unconditional transition

Figure 14-1—State diagram notation example

Each permissible transition between the states is represented graphically by an arrow from the initial to the terminal state. A transition that is global in nature (e.g., an exit condition from all states to the IDLE or RESET state) is indicated by an open arrow. Labels on transitions are qualifiers that need to be fulfilled before the transition is taken. The label UCT designates an unconditional transition. Qualifiers described by short phrases are enclosed in parentheses.

State transitions and sending and receiving of messages occur instantaneously. When a state is entered and the condition to leave that state is not immediately fulfilled, the state executes continuously, sending the messages and executing the actions contained in the state in a continuous manner.

Some devices described in this standard are allowed to have two or more ports. State diagrams capable of describing the operation of devices with an unspecified number of ports require qualifier notation that allows testing for conditions at multiple ports. The notation used is a term that includes a description in parentheses of which ports need to meet the term for the qualifier to be satisfied (e.g., ANY and ALL). It is also necessary to provide for term-assignment statements that assign a name to a port that satisfies a qualifier. The following convention is used to describe a term-assignment statement that is associated with a transition:

a) The character ":" (colon) is a delimiter used to denote that a term assignment statement follows.
b) The character "<" (left arrow) denotes assignment of the value following the arrow to the term preceding the arrow.

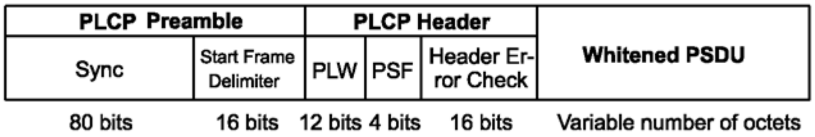
The state diagrams contain the authoritative statement of the procedures they depict; when apparent conflicts between descriptive text and state diagrams arise, the state diagrams are to take precedence. This does not, however, override any explicit description in the text that has no parallel in the state diagrams.

The models presented by state diagrams are intended as the primary specifications to be provided. It is important to distinguish, however, between a model and a real implementation. The models are optimized for simplicity and clarity of presentation, while any realistic implementation may place heavier emphasis on efficiency and suitability to a particular implementation technology. It is the functional behavior of any unit that need to match the standard, not its internal structure. The internal details of the model are useful only to the extent that they specify the external behavior clearly and precisely.

14.4.3 PLCP frame format

14.4.3.1 General

The PPDU frame format provides for the transfer of MAC sublayer MPDUs from any transmitting STA to all receiving STAs within the WLAN's BSS. The PPDU illustrated in Figure 14-2 consists of three parts: a PLCP preamble, a PLCP header, and a PSDU. The PLCP preamble provides a period of time for several receiver functions. These functions include antenna diversity, clock and data recovery, and field delineation of the PLCP header and the PSDU. The PLCP header is used to specify the length of the whitened PSDU field and support any PLCP management information. The PPDU contains the PLCP preamble, the PLCP header, and the PSDU modified by the PPDU data whitener.



PLCP Preamble		PLCP Header			Whitened PSDU
Sync	Start Frame Delimiter	PLW	PSF	Header Error Check	
80 bits	16 bits	12 bits	4 bits	16 bits	Variable number of octets

Figure 14-2—PLCP frame format

14.4.3.2 PLCP Preamble field

14.4.3.2.1 General

The PLCP Preamble field contains two separate subfields, the Preamble Synchronization (SYNC) field and the SFD, to allow the PHY circuitry to reach steady-state demodulation and synchronization of bit clock and frame start.

14.4.3.2.2 Preamble SYNC field

The Preamble SYNC field is an 80-bit field containing an alternating 01 pattern, transmitted starting with 0 and ending with 1, to be used by the PHY to detect a potentially receivable signal, select an antenna if diversity is utilized, and reach steady-state frequency offset correction and synchronization with the received packet timing.

14.4.3.2.3 SFD

The SFD consists of the 16-bit binary pattern 0000 1100 1011 1101 (transmitted leftmost bit first). The first bit of the SFD follows the last bit of the sync pattern. The SFD defines the frame timing.

14.4.3.3 PLCP Header field

14.4.3.3.1 General

The PLCP Header field contains three separate subfields: a 12-bit PSDU Length Word (PLW), a 4-bit PLCP Signaling field (PSF), and a 16-bit PLCP HEC field.

14.4.3.3.2 PLW

The PLW is passed from the MAC as a parameter within the PHY-TXSTART.request primitive. The PLW specifies the number of octets contained in the PSDU. Its valid values are X'001'–X'FFF', representing counts of one to 4095 octets. The PLW is transmitted LSB first and MSB last. The PLW is used by the receiving STA, in combination with the 32/33 coding algorithm specified in this clause, to determine the last bit in the packet.

14.4.3.3.3 PSF

The 4-bit PSF is defined in Table 14-3. The PSF is transmitted bit 0 first and bit 3 last.

Table 14-3—PSF bit descriptions

Bit	Parameter name	Parameter values	Description
0	Reserved	Default = 0	Reserved
1:3	PLCP_BITRATE	b1 b2 b3 = Data Rate 0 0 0 = 1.0 Mb/s, 0 0 1 = 1.5 Mb/s, 0 1 0 = 2.0 Mb/s, 0 1 1 = 2.5 Mb/s, 1 0 0 = 3.0 Mb/s, 1 0 1 = 3.5 Mb/s, 1 1 0 = 4.0 Mb/s, 1 1 1 = 4.5 Mb/s	This field indicates the data rate of the whitened PSDU from 1 Mb/s to 4.5 Mb/s in 0.5 Mb/s increments.

14.4.3.3.4 HEC field

The HEC field is a 16-bit CRC-16 error detection field. The HEC uses the CRC-16 generator polynomial $G(x)$ as follows:

$$G(x) = x^{16}+x^{12}+x^{5}+1$$

The HEC shall be the ones complement of the sum (modulo 2) of the following:

a) The remainder of $x^k \times (x^{15} + x^{14} + \ldots + x^2 + x^1 + 1)$ divided (modulo 2) by $G(x)$, where k is the number of bits in the PSF and PLW fields of the PLCP header;

b) The remainder after multiplication by x^{16} and then division (modulo 2) by $G(x)$ of the content (treated as a polynomial) of the PSF and PLW fields.

The HEC shall be transmitted with the coefficient of the highest term first.

As a typical implementation, at the transmitter, the initial remainder of the division is preset to all ones and is then modified by division of the PSF and PLW fields by the generator polynomial, $G(x)$. The ones complement of this remainder is inserted in the HEC field with the MSB transmitted first.

At the receiver, the initial remainder of the division is again preset to all ones. The division of the received PSF, PLW, and HEC fields by the generator polynomial, $G(x)$, results, in the absence of transmission errors, in a unique nonzero value, which is the following polynomial $R(x)$:

$$R(x) = \mathrm{x}^{12} + x^{11} + x^{10} + x^8 + x^3 + x^2 + x^1 + 1$$

14.4.3.4 PLCP data whitener

The PLCP data whitener uses a length-127 frame-synchronous scrambler followed by a 32/33 bias-suppression encoding to randomize the data and to minimize the data dc bias and maximum run lengths.

Data octets are placed in the transmit serial bit stream LSB first and MSB last. The frame synchronous scrambler uses the generator polynomial *S(x)* as follows:

$$S(x) = x^7+x^4+1$$

and is illustrated in Figure 14-3. The 127-bit sequence generated repeatedly by the scrambler is (leftmost bit used first) 00001110 11110010 11001001 00000010 00100110 00101110 10110110 00001100 11010100 11100111 10110100 00101010 11111010 01010001 10111000 1111111. The same scrambler is used to scramble transmit data and to descramble receive data. The data whitening starts with the first bit of the PSDU, which follows the last bit of the PLCP header. The specific bias suppression encoding and decoding method used is defined in Figure 14-7 and Figure 14-12. The format of the packet after data whitening is as shown in Figure 14-4.

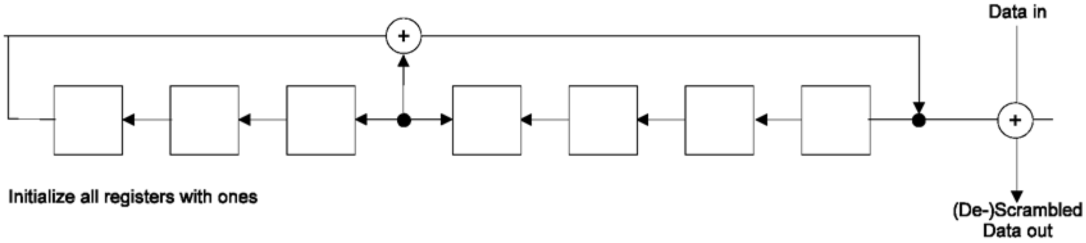


Figure 14-3—Frame synchronous scrambler/descrambler

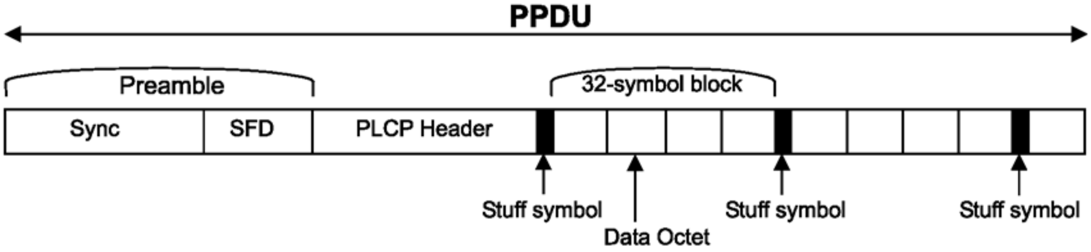


Figure 14-4—PLCP data whitener format

14.4.4 PLCP state machines

14.4.4.1 General

The PLCP consists of three state machines, as illustrated in the overview diagram of Figure 14-5: the TX, the CS/CCA, and the RX state machines. The three PLCP state machines are defined in the subclauses below; Figure 14-5 is not a state diagram itself. Execution of the PLCP state machines normally is initiated by the FH PLME state machine and begins at the CS/CCA state machine. The PLCP returns to the FH PLME state machine upon interrupt to service a PLME service request, e.g., PLME-SET, PLME-RESET.

14.4.4.2 Transmit PLCP

14.4.4.2.1 General

The transmit PLCP is invoked by the CS/CCA procedure immediately upon receiving a PHY-TXSTART.request(TXVECTOR) primitive from the MAC sublayer. The CSMA/CA protocol is performed by the MAC with the PHY PLCP in the CS/CCA procedure prior to executing the transmit procedure.

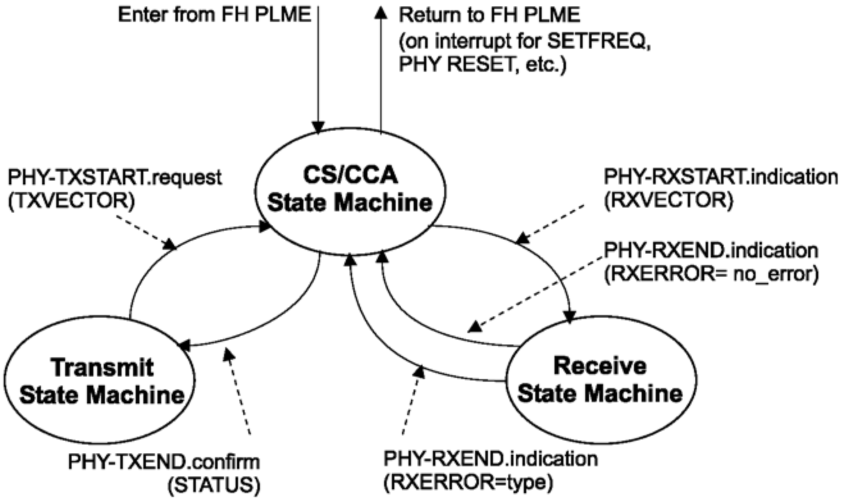


Figure 14-5—PLCP top-level state diagram

14.4.4.2.2 Transmit state machine

The PLCP transmit state machine illustrated in Figure 14-6 includes functions that need to be performed prior to, during, and after PPDU data transmission. Upon entering the transmit procedure in response to a PHY-TXSTART.request(TXVECTOR) primitive from the MAC, the PLCP shall switch the PHY PMD circuitry from receive to transmit state; ramp on the transmit power amplifier in the manner prescribed in 14.7; and transmit the preamble sync pattern and SFD. The PLCP shall generate the PLCP header as defined in 14.4.3.3 in sufficient time to send the bits at their designated bit slot time. The PLCP shall add the PLCP header to the start of the PSDU data.

Prior to transmitting the first PSDU data bit, the PLCP shall send a PHY-TXSTART.confirm primitive to the MAC indicating that the PLCP is ready to receive an MPDU data octet. The MAC passes an MPDU data octet to the PHY with a PHY-DATA.request(DATA) primitive, which the PHY responds to with a PHY-DATA.confirm primitive. This sequence of PHY-DATA.request(DATA) and PHY-DATA.confirm primitives shall be executed until the last data octet is passed to the PLCP. During transmission of the PSDU data, each bit of the PSDU shall be processed by the data whitener algorithm defined in Figure 14-7 and described in 14.4.3.4. Each PSDU data octet is processed and transmitted LSB first and MSB last.

After the last MPDU octet is passed to the PLCP, the MAC indicates the end of the frame with a PHY-TXEND.request primitive). After the last bit of the PSDU data has completed propagation through the radio and been transmitted on the air, the PLCP shall complete the transmit procedure by sending a PHY-TXEND.confirm primitive to the MAC sublayer, ramp off the power amplifier in the manner prescribed in 14.7, and switch the PHY PMD circuitry from transmit to receive state. The execution shall then return to the CS/CCA procedure.

The weights assigned to each value of the symbols are defined in Table 14-4 for the 1 Mb/s [two-level Gaussian frequency shift keying (2GFSK)] and 2 Mb/s [four-level GFSK (4GFSK)] symbols.

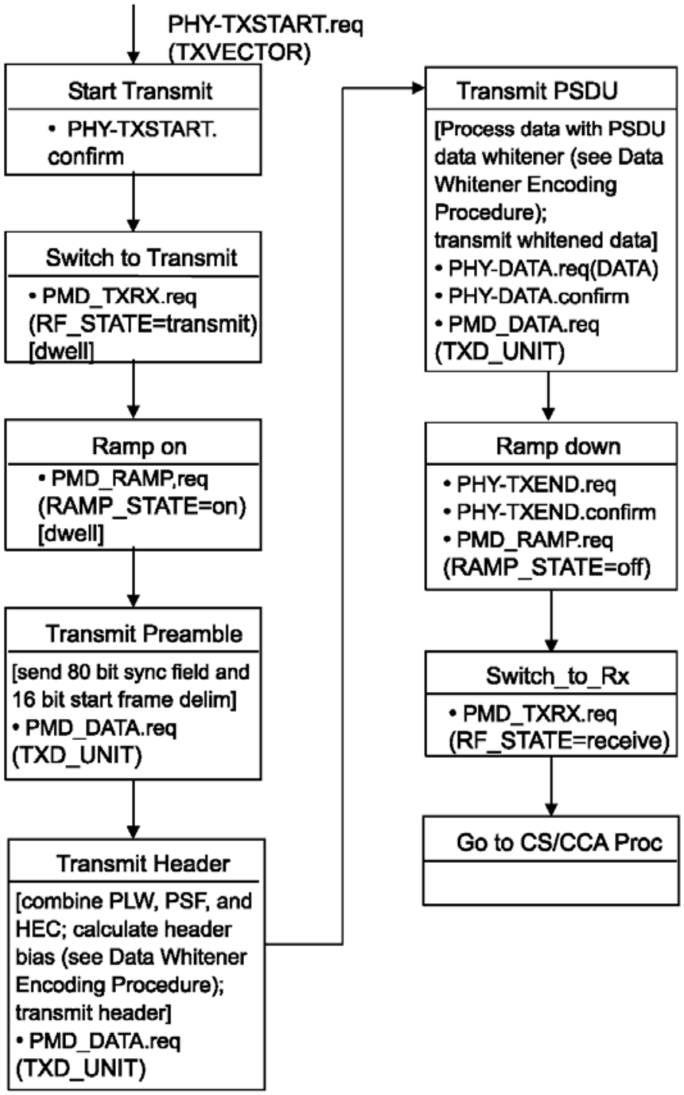


Figure 14-6—Transmit state machine

Data whitener encoding algorithm:

```
/*          If MSB of stuff symbol = 1 then the next block is inverted; 0 = not inverted */
/*          Accumulate PLCP header; begin stuffing on first bit of the PSDU */

/********* Calculate number of 32-symbol BSE blocks required to send PSDU;
                    no padding is necessary when the number of symbols is not a multiple of 32 *********/
Input parameter: number_of_PSDU_octets, rate;                /* rate is 1 or 2*/
number_of_symbols= (number_of_PSDU_octets *8) /rate;
number_of_blocks_in_packet = truncate{(number_of_symbols + 31) / 32)};

/********* Accumulate the bias in the header to use in calculating the inversion state of the first
                    block of PSDU data *********/
Read in header {b(1),...,b(32)};                  /* b(1) is first bit in */
header_bias = Sum{weight(b(1)),...,weight(b(32))};
                              /* calculate bias in header; weights are defined in Table 14-4*/
Transmit {b(1),...,b(32)};                        /* no stuffing on header */
accum=header_bias;                                /* initialize accum */
Initialize scrambler to all ones;

/********* Whiten the PSDU data with scrambler and BSE encoder *********/
For n = 1 to number_of_blocks_in_packet
{
          b(0) = 0 for 1 Mb/s; b(0)=00 for 2 Mb/s;                   /* b(0) is the stuff symbol */
          N = min(32, number_of_symbols);                            /* N= block size in symbols */
          Read in next symbol block {b(1),...,b(N)};                 /* b(n) = {0,1} or {0,1,2,3};
          1 - 8 octets, use PHY-DATA.req(DATA), PHY-DATA.confirm primitive for each octet*/
          Scramble {b(1),...,b(N)};                                         /* see 14.4.3.4*/
          bias_next_block = Sum{weight(b(0)),...,weight(b(N))};             /* calculate bias with b(0)=0 */

          /***** if accum and bias of next block has the same sign, then invert block;
                    if accum=0 or bias_next_block=0, do not invert *****/
          If {[accum * bias_next_block > 0] then
           {
                    Invert {b(0),...,b(N)};           /* Invert deviation, or, negate MSB of symbol */
                    bias_next_block = - bias_next_block;
           }

          accum = accum + bias_next_block;
          transmit {b(0),...,b(N)};                   /* b(0) is first symbol out */
          number_of_symbols = number_of_symbols - N
}
```

Figure 14-7—Data whitener encoding procedure

Table 14-4—PLCP field bit descriptions

2GFSK	4GFSK	Weight
—	10	3
1	—	2
—	11	1
Center	Center	0
—	01	–1
0	—	–2
—	00	–3

14.4.4.2.3 Transmit state timing

The transmit timing illustrated in Figure 14-8 is defined from the instant that the PHY-TXSTART.request(TXVECTOR) primitive is received from the MAC sublayer. The PLCP shall switch the PMD circuitry from receive to transmit, turn on and settle the transmitter, and begin transmitting the first bit of the preamble at the antenna within a maximum of 20 µs of receiving the PHY-TXSTART.request(TXVECTOR) primitive. The PLCP preamble shall be transmitted at 1 Mb/s and be completed in 96 µs. The PLCP header shall be transmitted at 1 Mb/s and be completed in 32 µs. The variable-length PSDU shall be transmitted at the selected data rate. After the last bit of the PSDU data has completed propagation through the radio and been transmitted on the air, the PLCP shall send the PHY-TXEND.confirm primitive to the MAC sublayer. The PLCP shall turn off the transmitter, reducing the output energy to less than the specified off-mode transmit power within the time specified in 14.7. At the end of the power amplifier ramp down period, the PLCP shall switch the PMD circuitry from transmit to receive.

14.4.4.3 CS/CCA procedure

14.4.4.3.1 General

The CS/CCA procedure is executed while the receiver is turned on and the STA is not currently receiving or transmitting a packet. The CS/CCA procedure is used for two purposes: to detect the start of a network signal that can be received (CS) and to determine whether the channel is clear prior to transmitting a packet (CCA).

14.4.4.3.2 CS/CCA state machine

Timing for priority (PIFS, DIFS), contention backoff (slot times), and CS/CCA windows is defined relative to the end of the last bit of the last packet on the air. The CS/CCA state machine is shown in Figure 14-9. The PLCP shall perform a CS/CCA on a minimum of one antenna within a MAC contention backoff slot time of 50 µs. The PLCP shall be capable of detecting within the slot time an FH-PHY-conformant signal that is received at the selected antenna up to 22 µs after the start of the slot time with the synchronous detection performance specified in 14.7.15.4. Detection performance with 01 sync patterns and with random data patterns is specified in 14.7.15.4. If a start of a transmission is asynchronous with the BSS and arrives after the start of the slot but at least 16 µs prior to the end of the slot, the PLCP shall indicate a busy channel prior to the end of the slot time with the asynchronous detection performance specified in 14.7.15.4. The CCA indication immediately prior to transmission shall be performed on an antenna with essentially the same free space gain and gain pattern as the antenna to be used for transmission. The method of determining CS/CCA is unspecified except for the detection performance of a conformant method as specified in 14.7.15.4.

If a PHY-TXSTART.request(TXVECTOR) primitive is received, the CS/CCA procedure shall exit to the transmit procedure within 1 µs. If a PHY-CCARESET.request primitive is received, the PLCP shall reset the CS/CCA state machine to the state appropriate for the end of a complete received frame. This service primitive is generated by the MAC at the end of a NAV period. The PHY shall indicate completion of the request by sending a PHY-CCARESET.confirm primitive to the MAC.

If a CS/CCA returns a channel idle result, the PHY shall send a PHY-CCA.indication(STATUS=idle) primitive to the MAC.

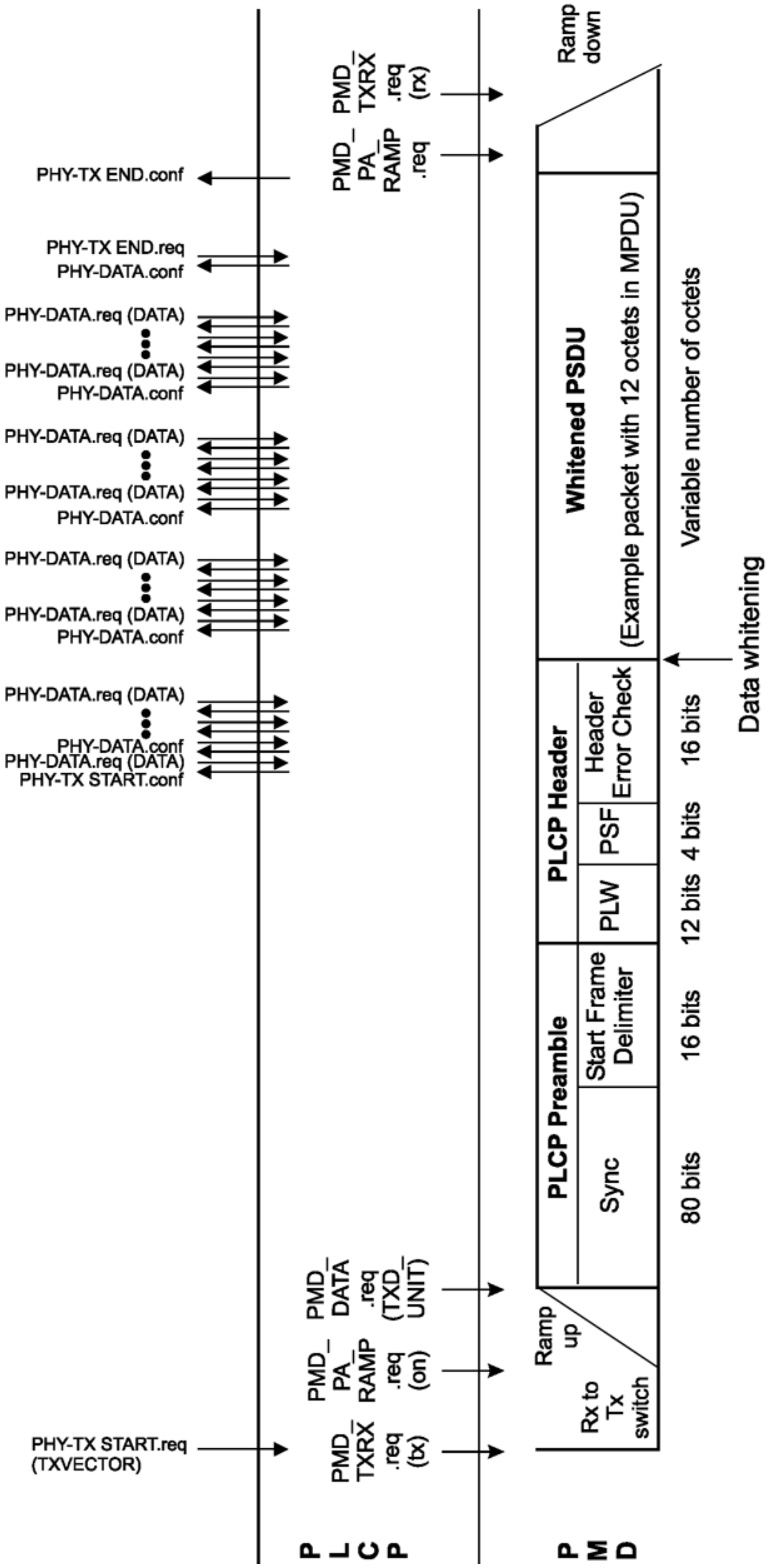


Figure 14-8—Transmit state timing

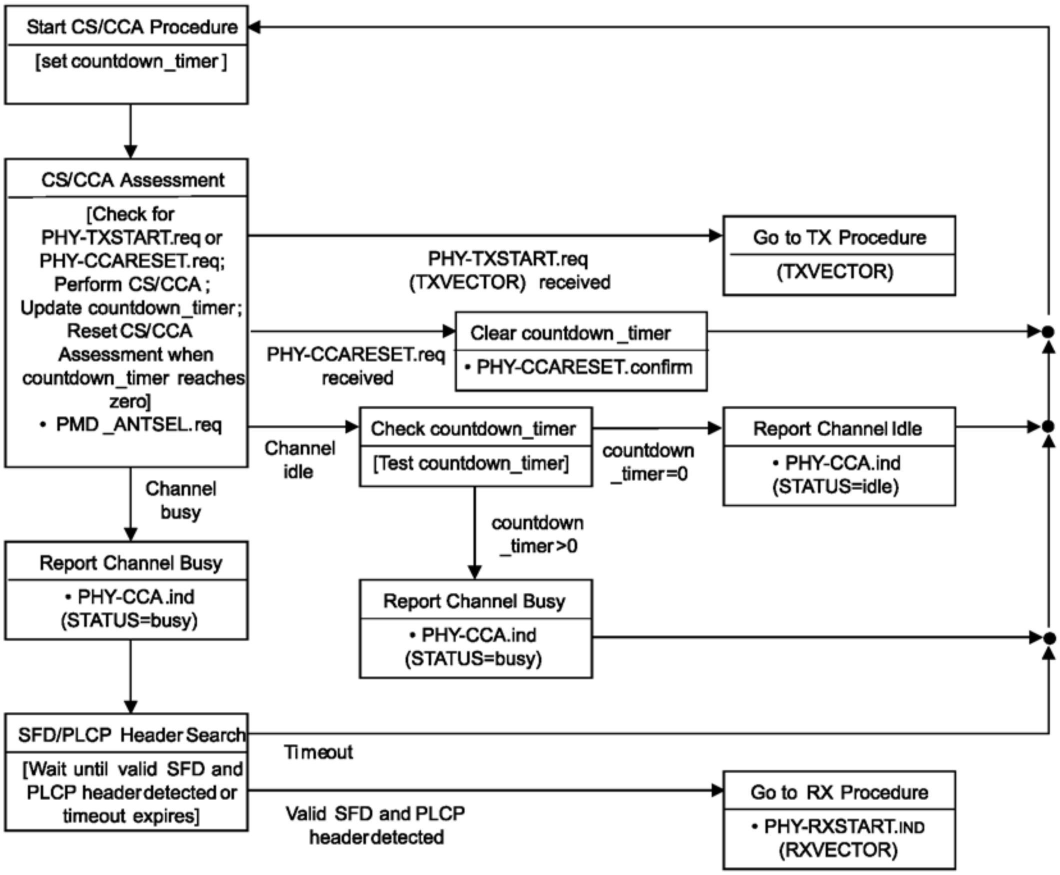

Start CS/CCA Procedure
[set countdown_timer]
CS/CCA Assessment
[Check for PHY-TXSTART.req or PHY-CCARESET.req; Perform CS/CCA; Update countdown_timer; Reset CS/CCA Assessment when countdown_timer reaches zero]
• PMD_ANTSEL.req
PHY-TXSTART.req (TXVECTOR) received
Go to TX Procedure
(TXVECTOR)
PHY-CCARESET.req received
Clear countdown_timer
• PHY-CCARESET.confirm
Channel idle
Check countdown_timer
[Test countdown_timer]
countdown _timer=0
Report Channel Idle
• PHY-CCA.ind (STATUS=idle)
countdown _timer>0
Report Channel Busy
• PHY-CCA.ind (STATUS=busy)
Channel busy
Report Channel Busy
• PHY-CCA.ind (STATUS=busy)
SFD/PLCP Header Search
[Wait until valid SFD and PLCP header detected or timeout expires]
Timeout
Valid SFD and PLCP header detected
Go to RX Procedure
• PHY-RXSTART.IND (RXVECTOR)

Figure 14-9—CS/CCA state machine

If a CS/CCA returns a channel busy result, the PHY shall send a PHY-CCA.indication(STATUS=busy) primitive to the MAC. Upon a channel busy assessment, the PLCP shall stop any antenna switching prior to the earliest possible arrival time of the SFD and detect a valid SFD and PLCP header if received. A valid PLCP header is defined as containing valid PLW and PSF values and a valid HEC field. If a valid SFD/PLCP header is detected, the CS/CCA procedure shall send a PHY-RXSTART.indication(RXVECTOR) primitive to the MAC sublayer and exit to the receive procedure. The PLCP shall dwell and search for the SFD/PLCP header for a minimum period longer than the latest possible arrival time of the SFD/PLCP header. Indication of a busy channel does not necessarily lead to the successful reception of a frame.

The octet/bit count remaining may be a nonzero value when returning from the receive procedure if a signal in the process of being received was lost prior to the end as determined from the LENGTH field of a valid PLCP header. The countdown timer shall be set to the octet/bit count and used to force the CS/CCA indication to remain in the BUSY state until the predicted end of the frame regardless of actual CS/CCA indications.

However, if the CS/CCA procedure indicates the start of a new frame within the countdown timer period, it is possible to transition to the receive procedure prior to the end of the countdown timer period. If the PHY transitions to receive under these conditions, the countdown timer shall be reset to the longer of

- — The remaining time of the current frame or
- — The length of the new frame.

When a nonzero countdown timer reaches 0, the PLCP shall reset the CS/CCA state machine to the state appropriate for the end of a complete received frame and the CS/CCA indication shall reflect the state of the channel.

If the receive procedure encountered an unsupported rate error, the PLCP shall keep the CS/CCA state at Busy for the duration of the frame by setting the countdown timer to the value corresponding to the calculated time based on the information in the PLCP header and the 33/32 expansion factor.

14.4.4.3.3 CS/CCA state timing

Timing for priority (PIFS, DIFS), contention backoff (slot times), and CS/CCA windows is defined relative to the end of the last bit of the last packet on the air. The PLCP shall perform a CS/CCA on a minimum of one antenna within a slot time. The appropriate CS/CCA indication shall be available prior to the end of each 50 µs slot time with the performance specified in 14.7. See Figure 14-10.

If a STA has not successfully received the previous packet, the perceived packet end time and slot boundary times have a higher uncertainty for that STA.

14.4.4.4 Receive PLCP

14.4.4.4.1 General

The receive PLCP is invoked by the CS/CCA procedure upon detecting a portion of the preamble sync pattern followed by a valid SFD and PLCP header.

14.4.4.4.2 Receive state machine

The receive PLCP shown in Figure 14-11 includes functions that need to be performed while the PPDU is being received. The receive PLCP begins upon detection of a valid SFD and PLCP header in the CS/CCA procedure. The PLCP shall set a PPDU octet/bit counter to indicate the last bit of the packet, receive the PPDU bits, and perform the data whitening decoding procedure shown in Figure 14-12 on each PPDU bit. The PLCP shall pass correctly received data octets to the MAC with a series of PHY-DATA.indication(DATA) primitives. After the last PPDU bit is received and the last octet is passed to the MAC, the PLCP shall send a PHY-RXEND.indication(RXERROR=no_error) primitive to the MAC sublayer. Upon error-free completion of a packet reception, the PLCP shall exit the receive procedure and return to the CS/CCA procedure with the octet/bit count set to 0.

If the PLCP header was decoded without a CRC error but encountered an unsupported rate, then the PLCP shall immediately complete the receive procedure with a PHY-RXEND.indication(RXERROR = unsupported_rate) primitive to the MAC, and return to the CS/CCA procedure with the octet/bit count remaining and the data rate value contained in the PLCP header.

If an error was detected during the reception of the PPDU, the PLCP shall immediately complete the receive procedure with a PHY-RXEND.indication(RXERROR=carrier_lost) primitive to the MAC, and return to the CS/CCA procedure with the octet/bit count remaining and the data rate value contained in the PLCP header.

.

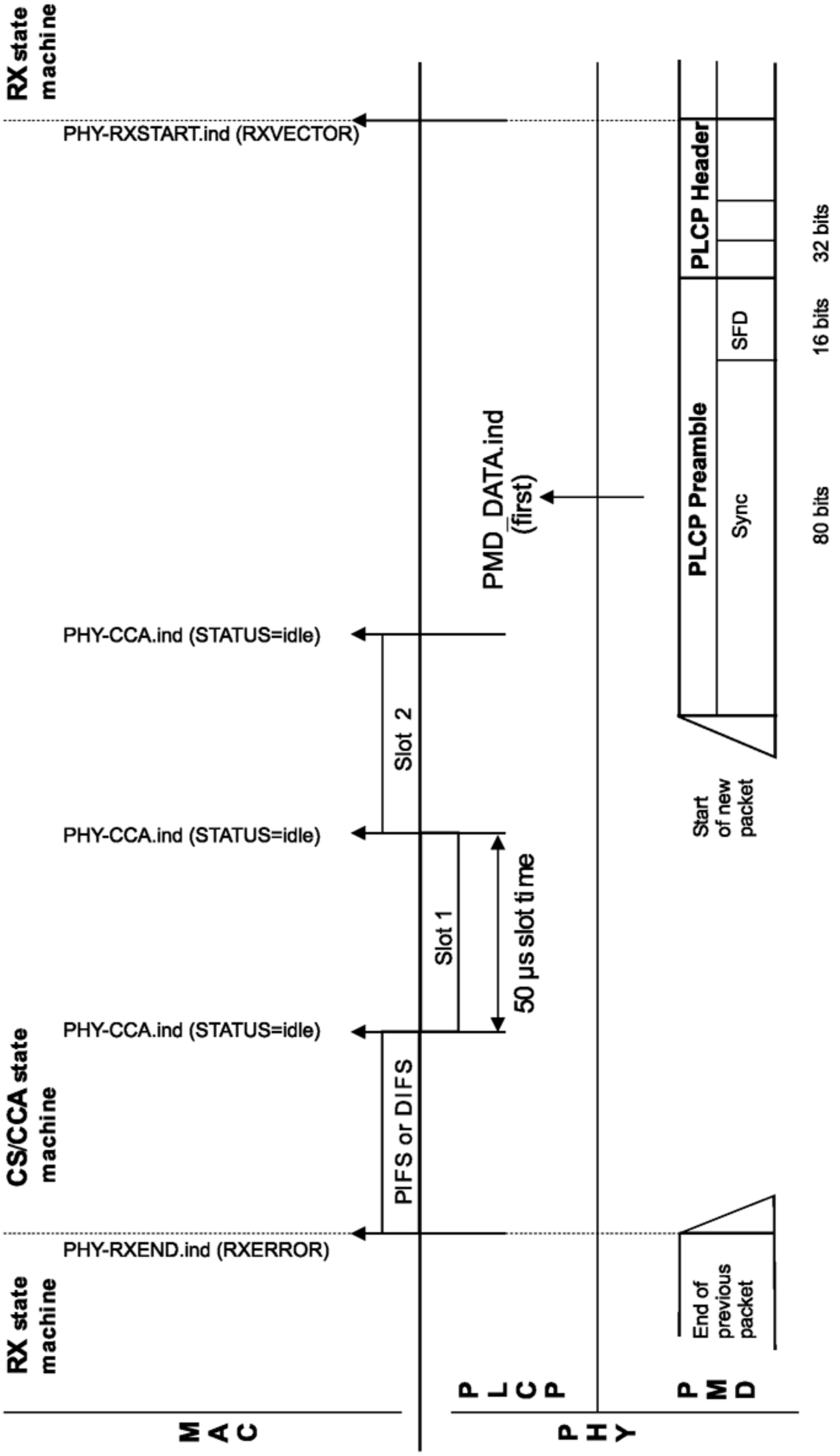


Figure 14-10—CS/CCA state timing

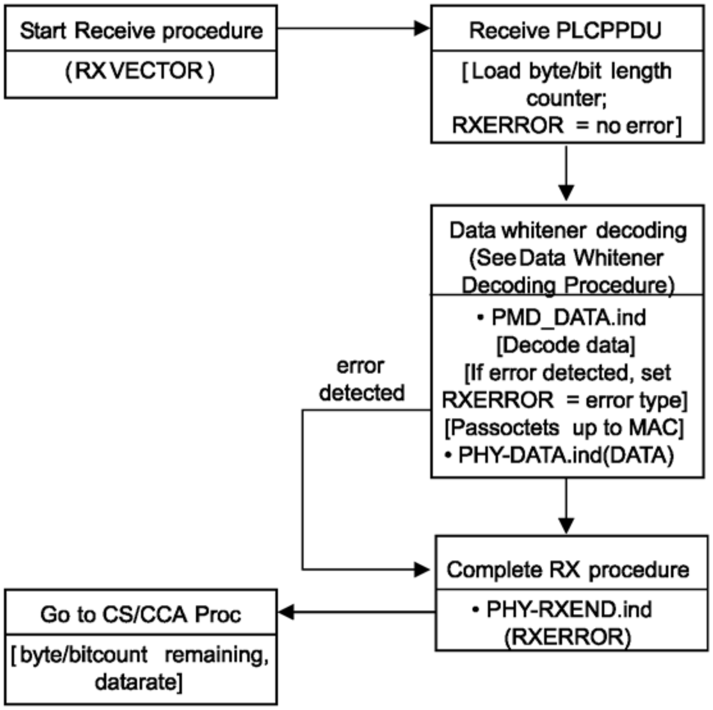


Figure 14-11—Receive state machine

```
Data whitener decoding algorithm:

/*          If MSB of stuff symbol = 1 then the next block is inverted; 0 = not inverted */
/*          Stuffing begins on first symbol of PLCP header following the SFD */
/*          Algorithm begins after verifying validity of header with HEC */

/********* Read header *********/
Read in header {b(1),...,b(32)};                          /* b(1) is first bit in */

Get number_of_PSDU_octets, rate from header;              /* rate is 1 or 2 */
number_of_symbols = (number_of_PSDU_octets*8)/rate
number_of_blocks_in_packet = truncate{(number_of_symbols + 31) / 32};
Initialize scrambler to all ones;

/********* Dewhiten the PPDU data with BSE decoder and descrambler *********/
For n = 1 to number_of_blocks_in_packet
{
            N = min(32, # of symbols remaining);          /* N= block size in symbols */
            Read in next block {b(0),...,b(N)};/                      * b(n) = {0,1} or {0,1,2,3} */

            If {[MSB of b(0)=1] then Invert {b(1),...,b(N)};          /* if invert bit=true */
            Descramble {b(1),...,b(N)};                   /* see 14.4.3.4*/
            Send {b(1),...,b(N)} to MAC
                              /* 1 - 8 octets; use PHY-DATA.indication(DATA) for each octet. */
}
```

Figure 14-12—Data whitener decoding procedure

14.4.4.4.3 Receive state timing

The receive state timing shown in Figure 14-13 is defined to begin upon detection of a valid SFD and PLCP header in the CS/CCA procedure. The PLCP shall begin receiving the variable-length whitened PSDU immediately after the end of the last bit of the PLCP header. The PLCP shall send a PHY-RXEND.indication(RXERROR) primitive after receiving the last PPDU data bit.

If any error was detected during the reception of the PPDU, the PLCP may send a PHY-RXEND.indication(RXERROR) primitive and terminate the receive procedure before the last bit arrives.

14.5 PLME SAP layer management

14.5.1 Overview

Subclause 14.5 describes the services provided by the FHSS PLME to the upper LMEs. The PLME/PMD services are defined in terms of service primitives. These primitives are abstract representations of the services and are not intended to restrict implementations.

14.5.2 FH PHY specific MLME procedures

14.5.2.1 Overview

The specific MLME procedures required for operating the FHSS PHY are specified in this portion of the subclause. The relationship between the MLME and FH PLME procedures is also described.

14.5.2.2 FH synchronization

The MLME of a compliant FH PHY STA shall perform the FH time synchronization procedure as defined in 10.1.6. This procedure provides for synchronized FH for all compliant FH PHY STAs within a single BSS. The FH PLME accepts PLME-SET.request primitives from the MLME to change the tune frequency at the time determined by the MLME. The tune frequency is changed by updating any combination of the Set, Pattern, and Index PHY MIB parameters.

14.5.3 FH PLME state machines

14.5.3.1 Overview

Subclause 14.5.3 describes the FH PLME state machines to turn the PMD on/off, reset the PLCP state machine, and change the frequency hop channel.

14.5.3.2 PLME state machine

The PLME state machine in Figure 14-14 begins with a PLME-SET.request(dot11CurrentPowerState= ON) primitive, which turns on the PHY circuitry, resets the PLME and PLCP state machines, and sends a PLME-SET.confirm primitive. The MAC then sends a series of three PLME-SET.request primitives to update the dot11CurrentSet, dot11CurrentPattern, and dot11CurrentIndex PHY MIB parameters, which together tune the PMD to the selected channel. The PLME then transfers execution to the PLCP state machine as defined in 14.4.4.

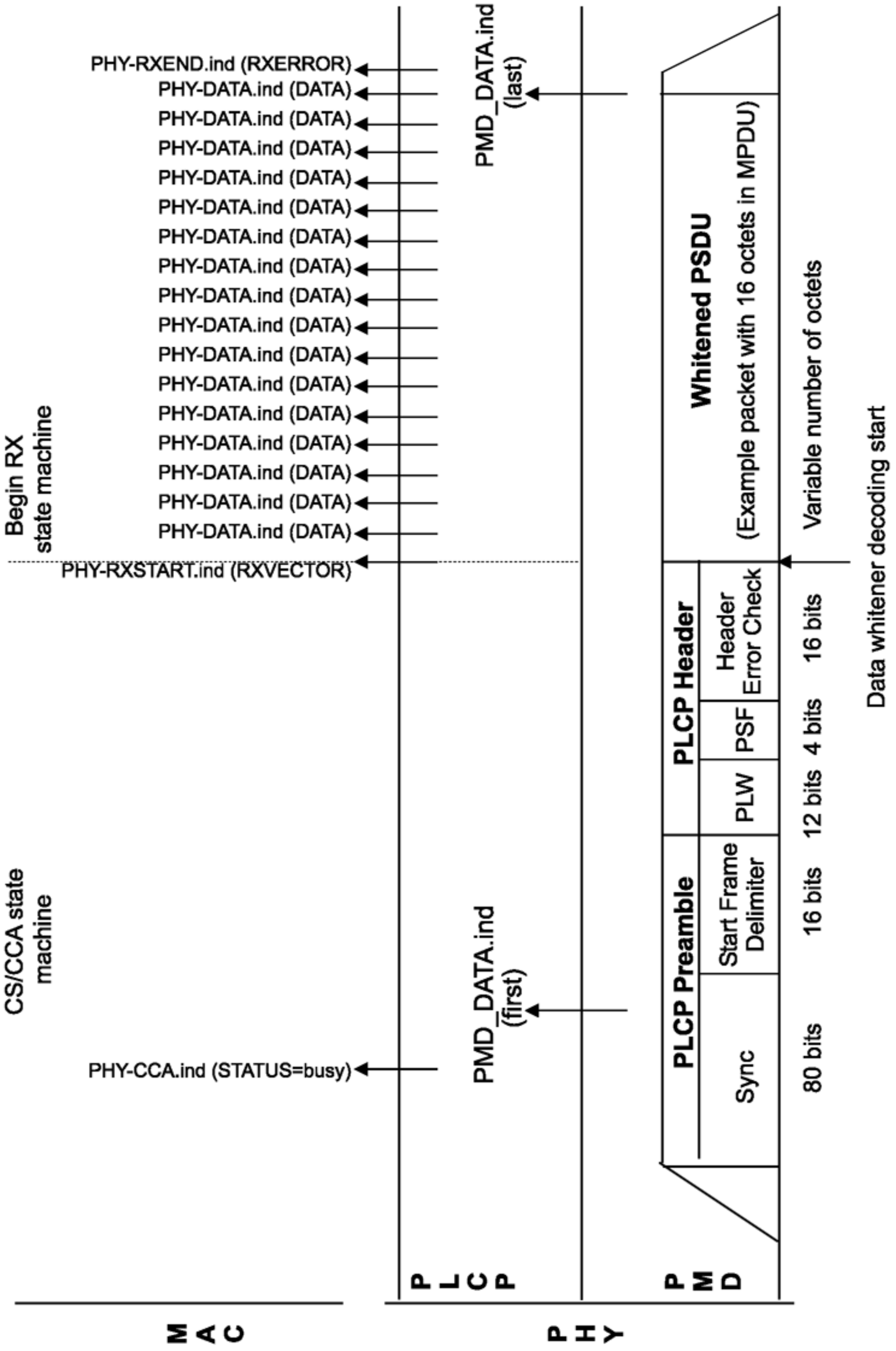


Figure 14-13—Receive timing

Upon receiving a PLME request from a higher level management entity, the PLCP shall return execution to the PLME state machine and process the request. A PLME-RESET.request primitive shall cause a reset to the PLME and PLCP state machines. A PLME-SET.request primitive updating the dot11CurrentIndex or a combination of the dot11CurrentSet, dot11CurrentPattern, and dot11CurrentIndex shall cause the PLCP to terminate a receive or CS/CCA process and change frequency before returning to the PLCP state machine. A PLME-SET.request(dot11CurrentPowerState=OFF) primitive shall cause the PLCP to terminate a receive or CS/CCA process, power-down the PMD circuitry, and return the PLME state machine to the idle state. PLME-SET.request primitives to any parameter other than the ones identified within this paragraph shall be executed and control shall be returned to the PLCP state machine. The MAC should not send a PLME request while the PLCP is in the transmit state.

All PLME-GET.request primitives shall be processed in parallel and with no interruption to the execution of any state machine in process.

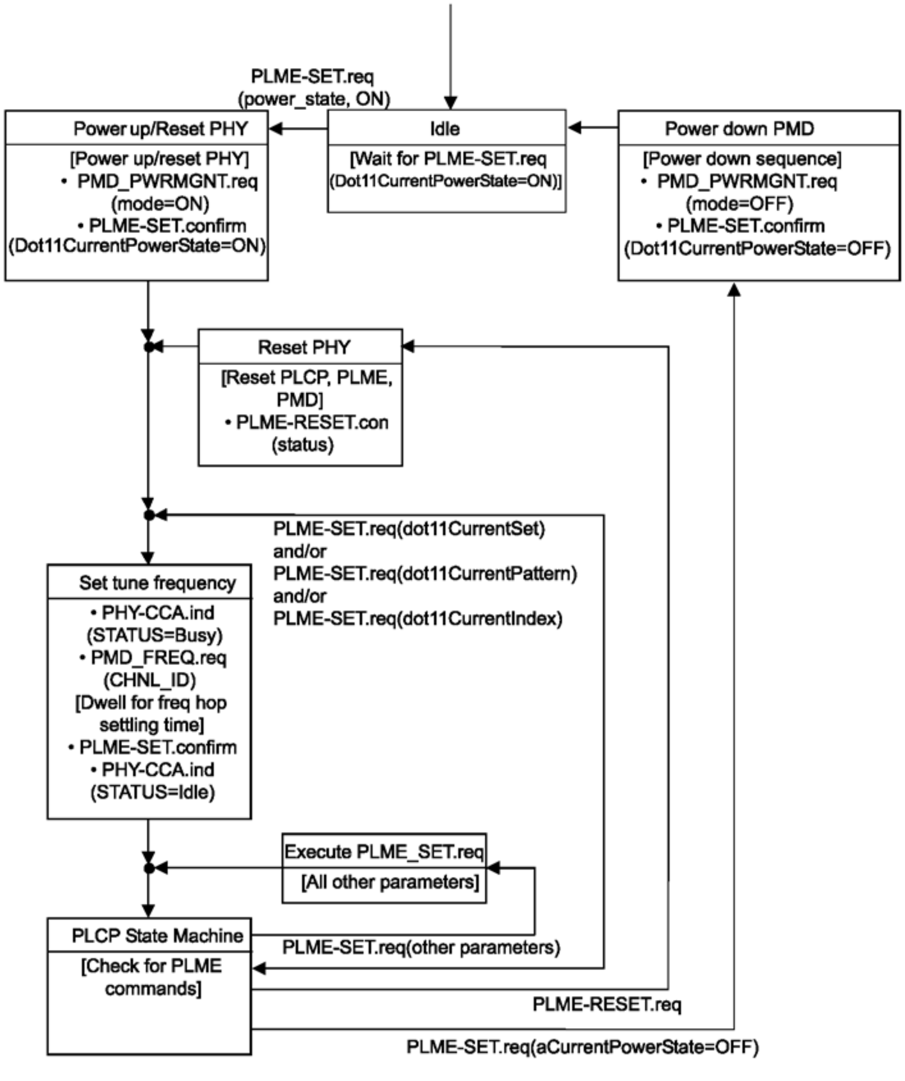


Figure 14-14—PLME state machine

14.5.3.3 PLME management primitives

The FH PLME uses the generic management primitives defined in 6.2 to manage all FH PHY parameters.

14.6 FHSS PMD sublayer services

14.6.1 Scope and field of application

The PMD services provided to the PLCP for the FHSS PHY are described in 14.6. Also defined in 14.6 are the functional, electrical, and radio frequency (RF) characteristics required for interoperability of implementations conforming to this specification. The relationship of this specification to the entire FHSS PHY is shown in Figure 14-15.

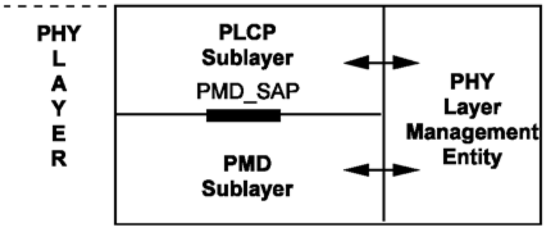


Figure 14-15—PMD layer reference model

14.6.2 Overview of services

In general, the FHSS PMD sublayer accepts PLCP sublayer service primitives and provides the actual means by which the signals required by these primitives are imposed onto the medium. In the FHSS PMD sublayer at the receiver the process is reversed. The combined function of the transmitting and receiving FHSS PMD sublayers results in a data stream, timing information, and receive parameter information being delivered to the receiving PLCP sublayer.

14.6.3 Overview of interactions

The primitives associated with the IEEE 802.11 PLCP sublayer to the FHSS PMD sublayer fall into the following two basic categories:

a) Service primitives that support PLCP peer-to-peer interactions;

b) Service primitives that have local significance and support sublayer-to-sublayer interactions.

14.6.4 Basic service and options

14.6.4.1 General

All of the service primitives described in 14.6.4 are considered mandatory unless otherwise specified.

14.6.4.2 PMD_SAP peer-to-peer service primitives

Table 14-5 indicates the primitives for peer-to-peer interactions.

Table 14-5—PMD_SAP peer-to-peer service primitives

Primitive	Request	Indicate	Confirm	Response
PMD_DATA	X	X	—	—

14.6.4.3 PMD_SAP sublayer-to-sublayer service primitives

Table 14-6 indicates the primitives for sublayer-to-sublayer interactions.

Table 14-6—PMD_SAP sublayer-to-sublayer service primitives

Primitive	Request	Indicate	Confirm	Response
PMD_TXRX	X	—	—	—
PMD_PA_RAMP	X	—	—	—
PMD_ANTSEL	X	—	—	—
PMD_TXPWRLVL	X	—	—	—
PMD_FREQ	X	—	—	—
PMD_RSSI	—	X	—	—
PMD_PWRMGMT	X	—	—	—

14.6.4.4 PMD_SAP service primitives parameters

Table 14-7 shows the parameters used by one or more of the PMD_SAP service primitives.

Table 14-7—List of parameters for PMD primitives

Parameter	Associated primitive	Value
TXD_UNIT	PMD_DATA.request	1 Mb/s: 0, 1 2 Mb/s: 0, 1, 2, 3
RXD_UNIT	PMD_DATA.indication	1 Mb/s: 0, 1 2 Mb/s: 0, 1, 2, 3
RF_STATE	PMD_TXRX.request	TRANSMIT, RECEIVE
RAMP_STATE	PMD_PA_RAMP.request	ON, OFF
ANTENNA_STATE	PMD_ANTSEL.request	1 to 255
TXPWR_LEVEL	PMD_TXPWRLVL.request	LEVEL1, LEVEL2, LEVEL3, LEVEL4
CHNL_ID	PMD_FREQ.request	2–80 inclusive
STRENGTH	PMD_RSSI.indication	0 to RSSI Max
MODE	PMD_PWRMGMT.request	ON, OFF

14.6.5 PMD_SAP detailed service specification

14.6.5.1 Introduction

Subclause 14.6.5 describes the services provided by each PMD primitive.

14.6.5.2 PMD_DATA.request

14.6.5.2.1 Function

This primitive defines the transfer of data from the PLCP sublayer to the PMD entity.

14.6.5.2.2 Semantics of the service primitive

The primitive shall provide the following parameter:
PMD_DATA.request(TXD_UNIT)

The TXD_UNIT parameter takes on one of two values: 1 or 0. This parameter represents a single data bit. The effect of this parameter is that the PMD properly modulates the medium to represent 1s or 0s as defined in the FHSS PMD modulation specifications for a given data rate.

14.6.5.2.3 When generated

This primitive is generated by the PLCP sublayer to request the transmission of a single data bit on the PMD sublayer. The bit clock is assumed to be resident or part of the PLCP and this primitive is issued at every clock edge once the PLCP has begun transmitting data.

14.6.5.2.4 Effect of receipt

The receipt of this primitive causes the PMD entity to encode and transmit a single data bit.

14.6.5.3 PMD_DATA.indication

14.6.5.3.1 Function

This primitive defines the transfer of data from the PMD entity to the PLCP sublayer.

14.6.5.3.2 Semantics of the service primitive

The primitive shall provide the following parameter:
PMD_DATA.indication(RXD_UNIT)

The RXD_UNIT parameter takes on one of two values: 1 or 0. This parameter represents the current state of the medium as determined by the FHSS PMD modulation specifications for a given data rate.

14.6.5.3.3 When generated

The PMD_DATA.indication is generated to all receiving PLCP entities in the network after a PMD_DATA.request primitive is issued.

14.6.5.3.4 Effect of receipt

The effect of receipt of this primitive by the PLCP is unspecified in this standard.

14.6.5.4 PMD_TXRX.request

14.6.5.4.1 Function

This primitive is used to place the PMD entity into the transmit or receive function.

14.6.5.4.2 Semantics of the service primitive

The primitive shall provide the following parameter:
PMD_TXRX.request(RF_STATE)

The RF_STATE parameter takes on one of two values: TRANSMIT or RECEIVE. When the value of the primitive is TRANSMIT, the RF state of the radio is transmit. If the value of the primitive is RECEIVE, the RF state of the radio is receive.

14.6.5.4.3 When generated

This primitive is generated when the mode of the radio needs to be set or when changing from transmit to receive or receive to transmit.

14.6.5.4.4 Effect of receipt

The receipt of this primitive by the PMD entity causes the mode of the radio to be in either transmit or receive.

14.6.5.5 PMD_PA_RAMP.request

14.6.5.5.1 Function

This primitive defines the start of the ramp up or ramp down of the radio transmitter's power amplifier.

14.6.5.5.2 Semantics of the service primitive

The primitive shall provide the following parameter:
  PMD_PA_RAMP.request(RAMP_STATE)

The RAMP_STATE parameter takes on one of two values: ON or OFF. When the value of the primitive is ON, the state of the transmit power amplifier is "on." If the value of the primitive is OFF, the state of the transmit power amplifier is "off."

14.6.5.5.3 When generated

This primitive is issued only during transmit and to establish the initial state. It is generated by the PLCP at the start of the transmit function to turn the transmitter's power amplifier "on." A power amplifier ramp-up period follows the change of state from "off" to "on." After the PLCP has transferred all required data to the PMD entity, this primitive is again issued by the PLCP to place the transmit power amplifier back into the "off" state. A power amplifier ramp-down period follows the change of state from "on" to "off."

14.6.5.5.4 Effect of receipt

The receipt of this primitive by the PMD entity causes the transmit power amplifier to turn on or off.

14.6.5.6 PMD_ANTSEL.request

14.6.5.6.1 Function

This primitive is used to select which antenna the PMD entity uses to transmit or receive data.

14.6.5.6.2 Semantics of the service primitive

The primitive shall provide the following parameter:
  PMD_ANTSEL.request(ANTENNA_STATE)

The ANTENNA_STATE parameter takes on values from one to N (where N is the number of antennas supported). When the value of the primitive is a ONE, the PMD switches to antenna 1 for receive or transmit; if the value of the primitive is TWO, the PMD entity switches to antenna 2 for receive or transmit, etc.

14.6.5.6.3 When generated

This primitive is generated at various times by the PLCP entity to select an antenna. During receive, this primitive can be used to manage antenna diversity. During transmit, this primitive can be use to select a transmit antenna. This primitive is also used during CCA.

14.6.5.6.4 Effect of receipt

The receipt of this primitive by the PMD entity causes the radio to select the antenna specified.

14.6.5.7 PMD_TXPWRLVL.request

14.6.5.7.1 Function

This primitive defines the power level the PMD entity uses to transmit data.

14.6.5.7.2 Semantics of the service primitive

The primitive shall provide the following parameter:
PMD_TXPWRLVL.request(TXPOWER_LEVEL)

The TXPOWER_LEVEL parameter can be one of the values listed in Table 14-8.

Table 14-8—Transmit power levels

TXPWR_LEVEL	Level description
LEVEL1	Defined as TxPowerLevel1 in MIB
LEVEL2	Defined as TxPowerLevel2 in MIB
LEVEL3	Defined as TxPowerLevel3 in MIB
LEVEL4	Defined as TxPowerLevel4 in MIB
LEVEL5	Defined as TxPowerLevel5 in MIB
LEVEL6	Defined as TxPowerLevel6 in MIB
LEVEL7	Defined as TxPowerLevel7 in MIB
LEVEL8	Defined as TxPowerLevel8 in MIB

14.6.5.7.3 When generated

This primitive is generated as part of the transmit sequence.

14.6.5.7.4 Effect of receipt

The receipt of this primitive by the PMD entity causes the transmit power level to be modified.

14.6.5.8 PMD_FREQ.request

14.6.5.8.1 Function

This primitive defines the frequency the PMD entity uses to receive or transmit data. Because changing the RF is not an immediate function, this primitive serves also as an indication of the start of this process. The completion of this process is dictated by other PMD specifications.

14.6.5.8.2 Semantics of the service primitive

The primitive shall provide the following parameter:
PMD_FREQ.request(CHANNEL_ID)

The CHANNEL_ID parameter can be one of the values listed in Table 14-11, Table 14-12, Table 14-13, or Table 14-14 (in 14.7.5).

14.6.5.8.3 When generated

This primitive is generated by the PLCP when a change to a new frequency is required.

14.6.5.8.4 Effect of receipt

The receipt of this primitive by the PMD entity causes the radio to change to a new frequency defined by the value of the CHNL_ID.

14.6.5.9 PMD_RSSI.indication

14.6.5.9.1 Function

This primitive transfers a receiver signal strength indication of the physical medium from the PMD sublayer to the PLCP sublayer. This value is used by the PLCP to perform any diversity or CCA functions required by the PLCP or other sublayers.

14.6.5.9.2 Semantics of the service primitive

The primitive shall provide the following parameter:
PMD_RSSI.indication(STRENGTH)

The STRENGTH parameter can be a value from 0 to 15. This parameter is an indication by the PMD sublayer of the magnitude of the energy observed at the selected antenna. This reported value is used to generate the RSSI term in the PHY-RXSTART.indication(RXVECTOR) primitive and might also be used by any diversity function. Because RSSI is only used in a relative manner by the MAC sublayer, this parameter is defined to have no more than 16 values, ranging from 0 to RSSI_Max. The value 0 is the weakest signal strength, while RSSI_Max is the strongest signal strength.

14.6.5.9.3 When generated

This primitive is generated continually by the PMD entity to transfer a RSSI to the PLCP.

14.6.5.9.4 Effect of receipt

The effect of receipt of this primitive by the PLCP is unspecified in this standard.

14.6.5.10 PMD_PWRMGMT.request

14.6.5.10.1 Function

This primitive is used by the higher layer entities to manage or control the power consumption of the PMD when not in use. This allows higher layer entities to put the radio into a sleep or standby mode when receipt or sending of any data is not expected.

14.6.5.10.2 Semantics of the service primitive

The primitive shall provide the following parameter:
PMD_PWRMGMT.request(MODE)

The MODE parameter takes one of two values: ON or OFF. When the value of the parameter is ON, the PMD entity enters into a fully functional mode that allows it to send or receive data. When the value of the parameter is OFF, the PMD entity places itself in a standby or power-saving mode. In the low-power mode, the PMD entity is not expected to be able to perform any request by the PLCP, nor is it expected to indicate any change in PMD state or status.

14.6.5.10.3 When generated

This primitive is delivered by the PLCP but actually is generated by a higher level LME.

14.6.5.10.4 Effect of receipt

Upon receipt of this primitive, the PMD entity enters a fully functional or low power consumption state depending on the value of the primitive’s parameter.

14.7 FHSS PMD sublayer, 1.0 Mb/s

14.7.1 1 Mb/s PMD operating specifications, general

In general, the PMD accepts convergence layer service primitives and provides the actual means by which the signals required by these primitives are imposed on the medium. In the PMD sublayer at the receiver, the process is reversed. The combined function of the transmitting and receiving PMD sublayers results in a data stream, timing information, and receive parameter information being delivered to the receiving convergence sublayer.

14.7.2 Regulatory requirements

WLANs implemented in accordance with this standard are subject to equipment certification and operating requirements established by regional and national regulatory administrations. The PMD specification establishes minimum technical requirements for interoperability, based upon established regulations at the time this standard was issued. These regulations are subject to revision, or may be superseded. Requirements that are subject to local geographic regulations are annotated within the PMD specification. Regulatory requirements that do not affect interoperability are not addressed within this standard. Implementers are referred to the appropriate regulatory sources for further information. Table 14-9 specify the current regulatory requirements for various geographic areas at the time this standard was developed. They are provided for information only and are subject to change or revision at any time.

14.7.3 Operating frequency range

A conformant PMD implementation shall be able to select the carrier frequency (F_c) from the full geographic-specific set of available carrier frequencies. Table 14-9 summarizes these frequencies for a number of geographic locations.

Table 14-9—Operating frequency range

Lower Limit	Upper limit	Regulatory range	Geography
2.402 GHz	2.480 GHz	2.400–2.4835 GHz	China
2.402 GHz	2.480 GHz	2.400–2.4835 GHz	North America
2.402 GHz	2.480 GHz	2.400–2.4835 GHz	Europe[a]
2.473 GHz	2.495 GHz	2.471–2.497 GHz	Japan
2.447 GHz	2.473 GHz	2.445–2.475 GHz	Spain
2.448 GHz	2.482 GHz	2.4465–2.4835 GHz	France
NOTE—The frequency ranges in this table are subject to the geographic-specific regulatory authorities.			

[a] Excluding Spain and France.

14.7.4 Number of operating channels

The number of transmit and receive frequency channels used for operating the PMD entity is 79 for the United States and Europe, and 23 for Japan. Table 14-10 summarizes these frequencies for a number of geographic locations. This is more fully defined in Table 14-11 to Table 14-14.

Table 14-10—Number of operating channels

Minimum	Hopping set	Geography
75	79	China
75	79	North America
20	79	Europe[a]
N/A	23	Japan
20	27	Spain
20	35	France
NOTE—The number of required hopping channels is subject to the geographic-specific regulatory authorities.		

[a] Excluding Spain and France.

14.7.5 Operating channel center frequency

The channel center frequency is defined in sequential 1.0 MHz steps beginning with the first channel, channel 2.402 GHz for China, the United States, and Europe excluding Spain and France, as listed in Table 14-11. The channel centers for Japan, starting at 2.473 GHz with 1 MHz increments, are listed in Table 14-12. The channel centers for Spain and France are listed in Table 14-13 and Table 14-14, respectively.

Table 14-11—Requirements in China, North America and Europe (excluding Spain and France; values specified in GHz)

Channel #	Value	Channel #	Value	Channel #	Value
2	2.402	28	2.428	54	2.454
3	2.403	29	2.429	55	2.455
4	2.404	30	2.430	56	2.456
5	2.405	31	2.431	57	2.457
6	2.406	32	2.432	58	2.458
7	2.407	33	2.433	59	2.459
8	2.408	34	2.434	60	2.460
9	2.409	35	2.435	61	2.461
10	2.410	36	2.436	62	2.462
11	2.411	37	2.437	63	2.463
12	2.412	38	2.438	64	2.464
13	2.413	39	2.439	65	2.465
14	2.414	40	2.440	66	2.466
15	2.415	41	2.441	67	2.467
16	2.416	42	2.442	68	2.468
17	2.417	43	2.443	69	2.469
18	2.418	44	2.444	70	2.470
19	2.419	45	2.445	71	2.471
20	2.420	46	2.446	72	2.472
21	2.421	47	2.447	73	2.473
22	2.422	48	2.448	74	2.474
23	2.423	49	2.449	75	2.475
24	2.424	50	2.450	76	2.476
25	2.425	51	2.451	77	2.477
26	2.426	52	2.452	78	2.478
27	2.427	53	2.453	79	2.479
—	—	—	—	80	2.480

Table 14-12—Requirements in Japan
(values specified in GHz)

Channel #	Value	Channel #	Value	Channel #	Value
73	2.473	81	2.481	89	2.489
74	2.474	82	2.482	90	2.490
75	2.475	83	2.483	91	2.491
76	2.476	84	2.484	92	2.492
77	2.477	85	2.485	93	2.493
78	2.478	86	2.486	94	2.494
79	2.479	87	2.487	95	2.495
80	2.480	88	2.488	—	—

Table 14-13—Requirements in Spain
(values specified in GHz)

Channel #	Value	Channel #	Value	Channel #	Value
47	2.447	56	2.456	65	2.465
48	2.448	57	2.457	66	2.466
49	2.449	58	2.458	67	2.467
50	2.450	59	2.459	68	2.468
51	2.451	60	2.460	69	2.469
52	2.452	61	2.461	70	2.470
53	2.453	62	2.462	71	2.471
54	2.454	63	2.463	72	2.472
55	2.455	64	2.464	73	2.473

Table 14-14—Requirements in France
(values specified in GHz)

Channel #	Value	Channel #	Value	Channel #	Value
48	2.448	60	2.460	72	2.472
49	2.449	61	2.461	73	2.473
50	2.450	62	2.462	74	2.474
51	2.451	63	2.463	75	2.475
52	2.452	64	2.464	76	2.476
53	2.453	65	2.465	77	2.477
54	2.454	66	2.466	78	2.478

Table 14-14—Requirements in France *(continued)* (values specified in GHz)

Channel #	Value	Channel #	Value	Channel #	Value
55	2.455	67	2.467	79	2.479
56	2.456	68	2.468	80	2.480
57	2.457	69	2.469	81	2.481
58	2.458	70	2.470	82	2.482
59	2.459	71	2.471	—	—

14.7.6 Occupied channel bandwidth

Occupied channel bandwidth shall meet all applicable local geographic regulations for 1 MHz channel spacing. The rate at which the PMD entity hops is governed by the MAC. The hop rate is an attribute with a maximum dwell time subject to local geographic regulations.

14.7.7 Minimum hop rate

The minimum hop rate shall be governed by the regulatory authorities.

14.7.8 Hop sequences

The hopping sequence of an individual PMD entity is used to create a pseudorandom hopping pattern utilizing uniformly the designated frequency band. Sets of hopping sequences are used to collocate multiple PMD entities in similar networks in the same geographic area and to enhance the overall efficiency and throughput capacity of each individual network.

An FH pattern, F_x, consists of a permutation of all frequency channels defined in Table 14-11 and Table 14-12. For a given pattern number, x, the hopping sequence is given by the following:

$$F_x = \{f_x(1), f_x(2), ...f_x(p)\} \quad (14\text{-}1)$$

where

$f_x(i)$ is the channel number (as defined in 14.7.4) for i^{th} frequency in x^{th} hopping pattern;

p is the number of frequency channels in hopping pattern (79 for China, North America and most of Europe, 23 for Japan, 35 for France, 27 for Spain).

Given the hopping pattern number, x, and the index for the next frequency, i (in the range 1 to p), the channel number shall be defined to be as follows:

$f_x(i)$ = $[b(i) + x]$ mod (79) + 2in China, North America and most of Europe,
with $b(i)$ defined in Table 14-15.
= $[(i - 1) \times x]$ mod (23) + 73in Japan.
= $[b(i) + x]$ mod (27) + 47in Spain with $b(i)$ defined in Table 14-16.
= $[b(i) + x]$ mod (35) + 48in France with $b(i)$ defined in Table 14-17.

The sequences are designed to ensure some minimum distance in frequency between contiguous hops. The minimum hop size is 6 MHz for China, North America and Europe, including Spain and France, and 5 MHz for Japan.

Table 14-15—Base-Hopping sequence *b(i)* for China, North America and most of Europe

i	*b(i)*	*i*	*b(i)*	*i*	*b(i)*	*i*	*b(i)*	*i*	*b(i)*	*i*	*b(i)*	*i*	*b(i)*	*i*	*b(i)*
1	0	11	76	21	18	31	34	41	14	51	20	61	48	71	55
2	23	12	29	22	11	32	66	42	57	52	73	62	15	72	35
3	62	13	59	23	36	33	7	43	41	53	64	63	5	73	53
4	8	14	22	24	71	34	68	44	74	54	39	64	17	74	24
5	43	15	52	25	54	35	75	45	32	55	13	65	6	75	44
6	16	16	63	26	69	36	4	46	70	56	33	66	67	76	51
7	71	17	26	27	21	37	60	47	9	57	65	67	49	77	38
8	47	18	77	28	3	38	27	48	58	58	50	68	40	78	30
9	19	19	31	29	37	39	12	49	78	59	56	69	1	79	46
10	61	20	2	30	10	40	25	50	45	60	42	70	28	—	—

Table 14-16—Base-Hopping sequence *b(i)* for Spain

i	*b(i)*	*i*	*b(i)*	*i*	*b(i)*
1	13	10	19	19	14
2	4	11	8	20	1
3	24	12	23	21	20
4	18	13	15	22	7
5	5	14	22	23	16
6	12	15	9	24	2
7	3	16	21	25	11
8	10	17	0	26	17
9	25	18	6	27	26

Table 14-17—Base-Hopping sequence *b(i)* for France

i	*b(i)*	*i*	*b(i)*	*i*	*b(i)*
1	17	13	31	25	15
2	5	14	20	26	3
3	18	15	29	27	11
4	32	16	22	28	30
5	23	17	12	29	24
6	7	18	6	30	9
7	16	19	28	31	27

Table 14-17—Base-Hopping sequence *b(i)* for France *(continued)*

i	*b(i)*	*i*	*b(i)*	*i*	*b(i)*
8	4	20	14	32	19
9	13	21	25	33	2
10	33	22	0	34	21
11	26	23	8	35	34
12	10	24	1	—	—

The hopping pattern numbers *x* are divided into three sets. The sets are designed to avoid prolonged collision periods between different hopping sequences in a set. Hopping sequence sets contain 26 sequences for China, North America and Europe, and 4 sequences per set for Japan:

For China, North America and most of Europe:

x = {0,3,6,9,12,15,18,21,24,27,30,33,36,39,42,45,48,51,54,57,60,63,66,69,72,75}	Set 1
x = {1,4,7,10,13,16,19,22,25,28,31,34,37,40,43,46,49,52,55,58,61,64,67,70,73,76}	Set 2
x = {2,5,8,11,14,17,20,23,26,29,32,35,38,41,44,47,50,53,56,59,62,65,68,71,74,77}	Set 3

For Japan:

x = {6,9,12,15}	Set 1
x = {7,10,13,16}	Set 2
x = {8,11,14,17}	Set 3

For Spain:

x = {0,3,6,9,12,15,18,21,24}	Set 1
x = {1,4,7,10,13,16,19,22,25}	Set 2
x = {2,5,8,11,14,17,20,23,26}	Set 3

For France:

x = {0,3,6,9,12,15,18,21,24,27,30}	Set 1
x = {1,4,7,10,13,16,19,22,25,28,31}	Set 2
x = {2,5,8,11,14,17,20,23,26,29,32}	Set 3

The three sets of hopping sequences for China, North America and most of Europe, of 26 patterns each, are listed in Table I-1, Table I-2, and Table I-3. Similarly, there are three sets for Japan of four patterns each. The three sets for Spain have nine patterns each. The three sets for France have 11 patterns each. The channel numbers listed under each pattern refer to the actual frequency values listed in Table 14-11 and Table 14-12.

14.7.9 Unwanted emissions

Conformant PMD implementations shall limit the emissions that fall outside of the operating frequency range, defined in Table 14-9, to the geographically applicable limits**.**

14.7.10 Modulation

The minimum set of requirements for a PMD to be compliant with the IEEE 802.11 FHSS PHY shall be as follows.

The PMD shall be capable of operating using 2GFSK modulation with a nominal bandwidth bit time (BT) = 0.5. The PMD shall accept symbols from the set {{1},{0}} from the PLCP. The symbol {1} shall be encoded with a peak deviation of ($+f_d$), giving a peak transmit frequency of (F_c+f_d), which is greater than the carrier center frequency (F_c). The symbol {0} shall be encoded with a peak frequency deviation of ($-f_d$), giving a peak transmit frequency of (F_c-f_d).

An incoming bit stream at 1 Mb/s shall be converted to symbols at *Fclk* = 1 *M*symbols/s, as shown in Table 14-18.

Table 14-18—Symbol encoding into carrier deviation (1 Mb/s, 2GFSK)

Symbol	Carrier deviation
1	1/2 × h2 × Fclk
0	–1/2 × h2 × Fclk
NOTE—These deviation values are measured using the center symbol of 7 consecutive symbols of the same value. The instantaneous deviation will vary due to Gaussian pulse shaping.	

The deviation factor h2 for 2GFSK (measured as difference between frequencies measured in the middle of 0000 and 1111 patterns encountered in the SFD, divided by 1 MHz) is nominally 0.32.

The minimum frequency deviation, as shown in Figure 14-16, shall be greater than 110 kHz relative to the nominal center frequency F_c. F_d is the average center frequency of the last 8 bits of the Preamble SYNC field, measured as the deviation at the midsymbol. Midsymbol is defined as the point that is midway between the zero crossings derived from a best fit to the last 8 bits of the Sync field. Maximum deviation is not specified, but modulation is subject to the occupied bandwidth limits of 14.7.5.

The zero crossing error shall be less than ±1/8 of a symbol period. The zero crossing error is the time difference between the ideal symbol periods and measured crossings of F_c. This is illustrated in Figure 14-16.

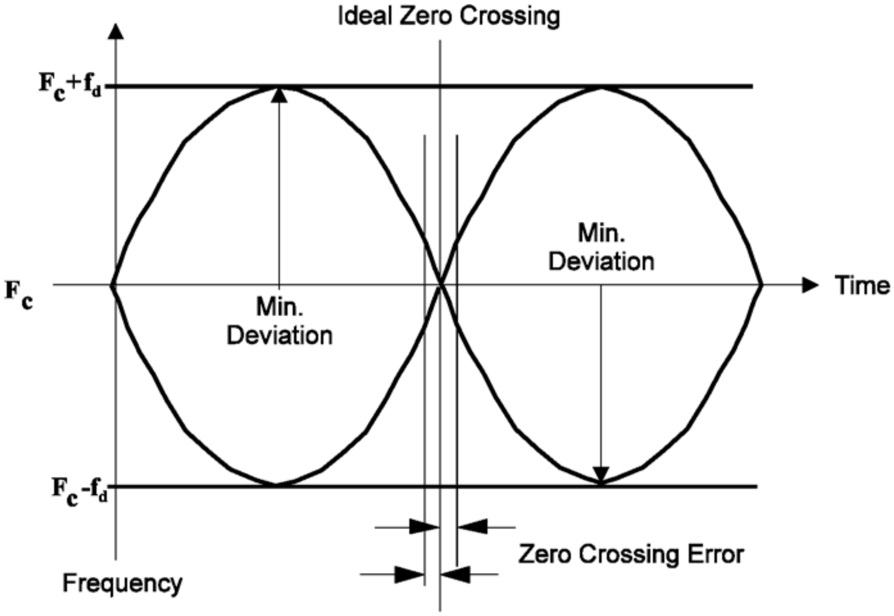


Figure 14-16—Transmit modulation mask

14.7.11 Channel data rate

A compliant IEEE 802.11 FHSS PMD shall be capable of transmitting and receiving at a nominal data rate of 1.0 Mb/s ±50 ppm.

14.7.12 Channel switching/settling time

The time to change from one operating channel frequency, as specified in 14.7.3, is defined as 224 µs. A conformant PMD meets this switching time specification when the operating channel center frequency has settled to within ±60 kHz of the nominal channel center frequency as outlined in 14.7.3.

14.7.13 Receive to transmit switch time

The maximum time for a conformant PMD to switch the radio from the receive state to the transmit state and place the start of the first bit on the air shall be 19 µs. At the end of this 19 µs, the RF carrier shall be within the nominal transmit power level range, and within the described modulation specifications.

14.7.14 PMD transmit specifications

14.7.14.1 Introduction

Subclause 14.7.14 describes the transmit functions and parameters associated with the PMD sublayer. In general, these are specified by primitives from the PLCP, and the transmit PMD entity provides the actual means by which the signals required by the PLCP primitives are imposed onto the medium.

14.7.14.2 Nominal transmit power

The nominal transmit power of a frame is defined as the power averaged between the start of the first symbol in the PLCP header to the end of the last symbol in the PLCP header. When in the transmit state, the transmit power shall be within 2 dB of the nominal transmit power from the start of the Preamble SYNC field to the last symbol at the end of the frame.

14.7.14.3 Transmit power levels

Unless governed by more stringent local geographic regulations, the radiated emissions from compliant devices shall meet IEEE Std C95.1 limits for controlled or uncontrolled environments, in accordance with their intended usage. In addition, all conformant PMD implementations shall support at least one power level with a minimum equivalent isotropically radiated power (EIRP) of 10 mW.

14.7.14.4 Transmit power level control

If a conformant PMD implementation has the ability to transmit in a manner that results in the EIRP of the transmit signal exceeding the level of 100 mW, at least one level of TPC shall be implemented. This TPC shall be such that the level of the emission is reduced to a level at or below 100 mW under the influence of said power control.

14.7.14.5 Transmit spectrum shape

Within the operational frequency band the transmitter shall pass a spectrum mask test. The duty cycle between Tx and Rx is nominally 50% and the transmit frame length is nominally 400 µs. The adjacent channel power is defined as the sum of the power measured in a 1 MHz band. For a pseudorandom data pattern, the adjacent channel power shall be a function of the offset between channel number *N* and the assigned transmitter channel *M*, where *M* is the actual transmitted center frequency and *N* is a channel separated from it by an integer number of megahertz.

Channel offset:
$|N–M|=2$ –20 dBm or –40 dBc, whichever is the lower power.
$|N–M|\geq3$ –40 dBm or –60 dBc, whichever is the lower power.

The levels given in dBc are measured relative to the transmitter power measured in a 1 MHz channel centered on the transmitter center frequency. The adjacent channel power and the transmitter power for this subclause of the specification shall be measured with a resolution bandwidth of 100 kHz, a video bandwidth of 300 kHz, and a peak detector, and with the measurement device set to maximum hold.

For any transmit center frequency M, two exceptions to the spectrum mask requirements are permitted within the operational frequency band, provided the exceptions are less than –50 dBc, where each offset channel exceeded counts as a separate exception. An exception occurs when the total energy within a given 1 MHz channel as defined in 14.7.5 exceeds the levels specified in 14.7.14.2 to 14.7.14.5.

14.7.14.6 Transmit center frequency tolerance

The PMD transmit center frequency shall be within ±60 kHz of the nominal center frequency as specified in 14.7.5.

14.7.14.7 Transmitter ramp periods

The transmitter shall go from off to within 2 dB of the nominal transmit power in 8 µs or less. The transmitter shall go from within 2 dB of the nominal transmit power to off (less than –50 dBm) in 8 µs or less.

14.7.15 PMD receiver specifications

14.7.15.1 Introduction

Subclause 14.7.15 describes the receive functions and parameters associated with the PMD sublayer. In general, these are specified by primitives from the PLCP. The Receive PMD entity provides the actual means by which the signals required by the PLCP primitives are recovered from the medium. The PMD sublayer monitors signals on the medium and returns symbols from the set {{1},{0}} to the PLCP sublayer.

14.7.15.2 Input signal range

The PMD shall be capable of recovering a conformant PMD signal from the medium, as described in related subclauses, with a frame error ratio (FER) ≤ 3% for PSDUs of 400 octets generated with pseudorandom data, for receiver input signal levels in the range from –20 dBm to the receiver sensitivity (as specified in 14.7.15.5), across the frequency band of operation.

14.7.15.3 Receive center frequency acceptance range

An IEEE 802.11 FHSS-compliant PMD shall meet all specifications with an input signal having a center frequency range of ±60 kHz from nominal.

14.7.15.4 CCA power threshold

In the presence of any IEEE 802.11-compliant 1 Mb/s FH PMD signal above –85 dBm that starts synchronously with respect to slot times as specified in 14.4.4.3.2, the PHY shall signal busy, with a 90% probability of detection, during the preamble within the CCA window. In the presence of any IEEE 802.11-compliant 1 Mb/s FH PMD signal above –85 dBm that starts asynchronously with respect to slot times as specified in 14.4.4.3.2, the PHY shall signal busy, with a 70% probability of detection, during the preamble within the CCA window. In the presence of any IEEE 802.11-compliant 1 Mb/s FH PMD signal above –65 dBm, the PHY shall signal busy, with a 70% probability of detection, during random data within the CCA window. This specification applies to a PMD operating with a nominal EIRP of < 100 mW. A compliant PMD operating at a nominal output power greater than 100 mW shall use the following equation to define the CCA threshold, where P_t represents transmit power.

$$\text{CCA threshold (preamble)} = -85\ \text{dBm} - \left[5 \infty \log_{10}\left(\frac{P_t}{100\ \text{mW}}\right)\right]\ \text{dBm}$$

$$\text{CCA threshold (random data)} = \text{CCA threshold (preamble)} + 20\ \text{dB}$$

14.7.15.5 Receiver sensitivity

The sensitivity is defined as the minimum signal level required for an FER of 3% for PSDUs of 400 octets generated with pseudorandom data. The sensitivity shall be less than or equal to –80 dBm. The reference sensitivity is defined as –80 dBm for the 1 Mb/s FH PHY specifications.

14.7.15.6 Intermodulation

Intermodulation protection (IMp) is defined as the ratio of the minimum amplitude of one of two equal interfering signals to the desired signal amplitude, where the interfering signals are spaced 4 MHz and 8 MHz removed from the center frequency of the desired signal, both on the same side of center frequency. The IMp ratio is established at the interfering signal level that causes the FER of the receiver to be increased to 3% for PSDUs of 400 octets generated with pseudorandom data, when the desired signal is –77 dBm. Each interfering signal is modulated with the FH PMD modulation uncorrelated in time to each other or the desired signal. The PMD shall have the IMp for the interfering signal at 4 MHz and 8 MHz be ≥ 30 dB.

14.7.15.7 Desensitization (Dp)

Desensitization (Dp) is defined as the ratio to measured sensitivity of the minimum amplitude of an interfering signal that causes the FER at the output of the receiver to be increased to 3% for PSDUs of 400 octets generated with pseudorandom data, when the desired signal is –77 dBm. The interfering signal shall be modulated with the FHSS PMD modulation uncorrelated in time to the desired signal. The minimum Dp shall be as given in Table 14-19. The spectral purity of the interferer shall be sufficient to ensure that the measurement is limited by the receiver performance.

Table 14-19—1 Mb/s Dp

Interferer frequency[a]	Dp minimum
$M = N \pm 2$	30 dB
$M = N \pm 3$ or more	40 dB

[a] Where M is the interferer frequency and N is the desired channel frequency.

14.7.15.8 Receiver radiation

The signal leakage when receiving shall not exceed –50 dBm EIRP in the operating frequency range. The FHSS PHY shall conform with out-of-band spurious emissions by regulatory bodies.

14.8 FHSS PMD sublayer, 2.0 Mb/s

14.8.1 Overview

Subclause 14.8 details the RF specification differences of the optional 2 Mb/s operation from the baseline 1 Mb/s PMD as contained in 14.7. Unless otherwise specified in 14.8, the compliant PMD shall also meet all requirements of 14.7 when transmitting at 2 Mb/s. When implementing the 2 Mb/s option, the preamble and PHY header shall be transmitted at 1 Mb/s. STAs implementing the 2 Mb/s option shall also be capable of transmitting and receiving PPDUs at 1 Mb/s.

14.8.2 4GFSK modulation

For an FHSS 2 Mb/s PMD, the modulation scheme shall be 4GFSK, with a nominal symbol-period bandwidth product (BT) of 0.5. The four-level deviation factor, defined as the frequency separation of adjacent symbols divided by symbol rate, h4, shall be related to the deviation factor of the 2GFSK modulation, h2, by the following equation:

$$h4/h2 = 20055 \pm 0.01$$

An incoming bit stream at 2 Mb/s shall be converted to 2-bit words or symbols, with a rate of Fclk = 1 Msymbol/s. The first received bit shall be encoded as the LMB of the symbol in Table 14-20. The bits shall be encoded into symbols as shown in Table 14-20.

Table 14-20—Symbol encoding into carrier deviation

1 Mb/s, 2GFSK	
Symbol	**Carrier deviation**
1	1/2 × h2 × Fclk
0	–1/2 × h2 × Fclk
2 Mb/s, 4GFSK	
Symbol	**Carrier deviation**
10	3/2 × h4 × Fclk
11	1/2 × h4 × Fclk
01	–1/2 × h4 × Fclk
00	–3/2 × h4 × Fclk
NOTE—These deviation values are measured using the center symbol of 7 consecutive symbols of the same value. The instantaneous deviation will vary due to Gaussian pulse shaping.	

The deviation factor h2 for 2GFSK (measured as the difference between frequencies measured in the middle of 0000 and 1111 patterns encountered in the SFD, divided by 1 MHz) is nominally 0.32. The deviation factor h2 shall be no less than 0.30 (with maximum dictated by regulatory bandwidth requirement). Accordingly, h4 (measured as a difference between the outermost frequencies, divided by 3, divided by 1 MHz) is nominally 0.45 × 0.32 = 0.144, and shall be no less than 0.45 × 0.3 = 0.135.

The modulation error shall be less than ±15 kHz at the midsymbol time for 4GFSK, from the frequency deviations specified above, for a symbol surrounded by identical symbols, and less than ±25 kHz for any symbol. The deviation is relative to the actual center frequency of the RF carrier. For definition purposes, the actual center frequency is the midfrequency between symbols 11 and 01. The actual center frequency shall be within ±60 kHz of the nominal channel center frequency defined in 14.7.5 and shall not vary by more than ±10 kHz/ms, from the start to end of the PPDU. The peak-to-peak variation of the actual center frequency over the PPDU shall not exceed 15 kHz. Symbols and terms used within this subclause are illustrated in Figure 14-17.

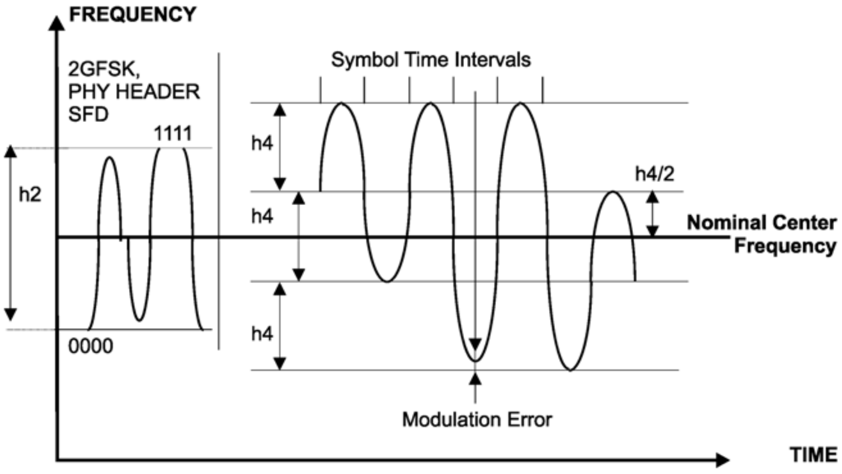


Figure 14-17—4GFSK transmit modulation

14.8.3 Frame structure for HS FHSS PHY

The high rate FHSS PPDU consists of PLCP preamble, PLCP header, and whitened PSDU. The PLCP preamble and PLCP header format are identical to the 1 Mb/s PHY, as described in 14.4.3. The whitened PSDU is transmitted in 2GFSK, 4GFSK, or potentially a higher-rate format, according to the rate chosen. The rate is indicated in a 3-bit field in a PLCP header, having a value of 1 or 2 bits per symbol (or Mb/s).

The PPDU is transmitted as four-level symbols, with the amount determined by number_of_symbols = (number_of_PSDU_octets × 8)/rate.

The input bits are scrambled according to the method in 14.4.3.4.

The scrambled bit stream is divided into groups of rate (1 or 2) consecutive bits. The bits are mapped into symbols according to Table 14-20.

A bias suppression algorithm is applied to the resulting symbol stream. The bias suppression algorithm is defined in 14.4.3.4, Figure 14-4, and Figure 14-7. A polarity control symbol is inserted prior to each block of 32 symbols (or less for the last block). The polarity control signals are 4GFSK symbols 10 or 00. The algorithm is equivalent to the case of 2GFSK, with the polarity symbol 2GFSK “1” replaced with 4GFSK symbol “10,” and the 2GFSK polarity symbol “0” replaced with a 4GFSK symbol “00.”

14.8.4 Channel data rate

The data rate for the whitened PSDU at the optional rate shall be 2.0 Mb/s ±50 ppm.

14.8.5 Input dynamic range

The PMD shall be capable of recovering a conformant PMD signal from the medium, as described in related subclauses, with an FER ≤ 3% for PSDUs of 400 octets generated with pseudorandom data, for receiver input signal levels in the range from –20 dBm to the receiver sensitivity (as specified in 14.8.6), across the frequency band of operation.

14.8.6 Receiver sensitivity

The sensitivity is defined as the minimum signal level required for an FER of 3% for PSDUs of 400 octets generated with pseudorandom data. The sensitivity shall be less than or equal to –75 dBm. The reference sensitivity is defined as –75 dBm for the 2 Mb/s FH PHY specifications.

14.8.7 IMp

IMp is defined as the ratio to –77 dBm of the minimum amplitude of one of the two equal-level interfering signals at 4 MHz and 8 MHz removed from center frequency, both on the same side of center frequency, that cause the FER of the receiver to be increased to 3% for PSDUs of 400 octets generated with pseudorandom data, when the desired signal is –72 dBm (3 dB above the specified sensitivity specified in 14.8.6). Each interfering signal is modulated with the FH 1 Mb/s PMD modulation uncorrelated in time to each other or the desired signal. The FHSS optional 2 Mb/s rate IMp shall be ≥ 25 dB.

14.8.8 Dp

Dp is defined as the ratio to measured sensitivity of the minimum amplitude of an interfering signal that causes the FER of the receiver to be increased to 3% for PSDUs of 400 octets generated with pseudorandom data, when the desired signal is –72 dB (3 dB above sensitivity specified in 14.8.6). The interfering signal shall be modulated with the FHSS PMD modulation uncorrelated in time to the desired signal. The minimum Dp shall be as given in Table 14-21.

Table 14-21—2 Mb/s Dp

Interferer frequency[a]	DP minimum
$M = N \pm 2$	20 dB
$M = N \pm 3$ or more	30 dB

[a] Where M is the interferer frequency and N is the desired channel frequency.

14.9 FHSS PHY MIB

14.9.1 FH PHY attributes

Subclause 14.9 defines the attributes for the FHSS MIB. Table 14-22 lists these attributes and the default values. A description of each attribute is given in 14.9.2.

Table 14-22—FHSS PHY attributes

Attribute	Default value	Operational semantics	Operational behavior
dot11PHYType	FHSS = X'01'	Static	Identical for all FH PHYs

dot11RegDomainsImplementedValue	FCC = X'10'
	IC = X'20'
	ETSI = X'30'
	Spain = X'31'
	France = X'32'
	Japan = X'40'
	China = X’50’
	Other = X'00'
		Static	Implementation dependent

dot11CurrentRegDomain	X'00'	Dynamic LME	Implementation dependent
dot11SupportedDataRatesTX	1 Mb/s = X'02' mandatory 2 Mb/s = X'04' optional	Static	Identical for all FH PHYs
dot11SupportedDataRatesRX	1 Mb/s = X'02' mandatory 2 Mb/s = X'04' optional	Static	Identical for all FH PHYs
dot11TxAntennaImplemented	Ant 1 = X'01' Ant 2 = X'02' Ant 3 = X'03' Ant n = n	Static	Implementation dependent
dot11CurrentTxAntenna	Ant 1 = default	Dynamic LME	Implementation dependent
dot11RxAntennaImplemented	Ant 1 = X'01' Ant 2 = X'02' Ant 3 = X'03' Ant n = n	Static	Implementation dependent
dot11DiversitySupportImplemented	Available = X'01' Not avail. = X'02' Control avail. = X'03'	Static	Implementation dependent
dot11DiversitySelectionRxImplemented	Ant 1 = X'01' Ant 2 = X'02' Ant 3 = X'03' Ant 4 = X'04' Ant 5 = X'05' Ant 6 = X'06' Ant 7 = X'07' Ant 8 = X'08'	Dynamic LME	Implementation dependent

dot11NumberSupportedPowerLevelsImplemented	Lvl1 = X'01'
	Lvl2 = X'02'
	Lvl3 = X'03'
	Lvl4 = X'04'
	Lvl5 = X'05'
	Lvl6 = X'06'
	Lvl7 = X'07'
	Lvl8 = X'08'
		Static	Implementation dependent

dot11TxPowerLevel1	Factory default	Static	Implementation dependent
dot11TxPowerLevel2	Factory default	Static	Implementation dependent
dot11TxPowerLevel3	Factory default	Static	Implementation dependent
dot11TxPowerLevel4	Factory def.	Static	Implementation dependent
dot11TxPowerLevel5	Factory def.	Static	Implementation dependent
dot11TxPowerLevel6	Factory def.	Static	Implementation dependent
dot11TxPowerLevel7	Factory def.	Static	Implementation dependent
dot11TxPowerLevel8	Factory def.	Static	Implementation dependent
dot11CurrentTxPowerLevel	TxPowerLevel1	Dynamic LME	Implementation dependent

Table 14-22—FHSS PHY attributes *(continued)*

Attribute	Default value	Operational semantics	Operational behavior
dot11HopTime	224 µs	Static	Identical for all FH PHYs
dot11CurrentChannelNumber	X'00'	Dynamic PLME	
dot11MaxDwellTime	390 TU	Static	Regulatory domain dependent
dot11CurrentSet	X'00'	Dynamic PLME	
dot11CurrentPattern	X'00'	Dynamic PLME	
dot11CurrentIndex	X'00'	Dynamic PLME	
dot11CurrentPowerState	X'01' off X'02' on	Dynamic LME	
NOTE—The column titled “Operational semantics” contains two types: static and dynamic. Static MIB attributes are fixed and cannot be modified for a given PHY implementation. MIB attributes defined as dynamic can be modified by some management entity. When an attribute is defined as dynamic, the column also shows which entity has control over the attribute. LME refers to the MLME, while PHY refers to the PLME.			

14.9.2 FH PHY attribute definitions

14.9.2.1 dot11PHYType

The dot11PHYType is FHSS. The LME uses this attribute to determine what PLCP and PMD are providing services to the MAC. It also is used by the MAC to determine what MAC sublayer management state machines need to be invoked to support the PHY. The value of this attribute is defined as the integer 01 to indicate the FHSS PHY.

14.9.2.2 dot11RegDomainsImplementedValue

Operational requirements for FHSS PHY are defined by agencies representing certain geographical regulatory domains. These regulatory agencies may define limits on various parameters that differ from region to region. This parameters may include dot11TxPowerLevels, and dot11MaxDwellTime, as well as the total number of frequencies in the hopping pattern. The values shown in Table 14-23 indicate regulatory agencies supported by this document.

Table 14-23—Regulatory domain codes

Code point	Regulatory agency	Region
X'10'	FCC	United States
X'20'	IC	Canada
X'30'	ETSI	Most of Europe
X'31'	Spain	Spain
X'32'	France	France
X'40'	Japan	Japan
X’50’	Radio Administration of Information Industry Ministry	China

Because a PLCP and PMD might be designed to support operation in more than one regulatory domain, this attribute can actually represent a list of agencies. This list may be one or more of the above agencies and shall be terminated using the null terminator. Upon activation of the PLCP and PMD, the information in this list shall be used to set the value of the dot11CurrentRegDomain attribute.

14.9.2.3 dot11CurrentRegDomain

The dot11CurrentRegDomain attribute for the FHSS PHY is defined as the regulatory domain under which the PMD is currently operating. This value shall be one of the values listed in the dot11RegDomainsImplementedValue list. This MIB attribute is managed by the LME.

14.9.2.4 dot11CurrentPowerState

The dot11CurrentPowerState attribute for the FHSS PHY allows the MLME to control the power state of the PHY. This attribute can be updated using the PLMESET.request primitive. The permissible values are ON and OFF.

14.9.2.5 dot11SupportedDataRatesTX

The dot11SupportedDataRatesTX attribute for the FHSS PHY is defined as a null terminated list of supported data rates in the transmit mode for this implementation. Table 14-24 shows the possible values appearing in the list.

Table 14-24—Supported data rate codes (dot11SupportedDataRatesTX)

Code point	Data rate
X'02'	1 Mb/s
X'04'	2 Mb/s
X'00'	Null terminator

14.9.2.6 dot11SupportedDataRatesRX

The dot11SupportedDataRatesRX attribute for the FHSS PHY is defined as a null terminated list of supported data rates in the receive mode for this implementation. Table 14-25 shows the possible values appearing in the list.

Table 14-25—Supported data rate codes (dot11SupportedDataRatesRX)

Code point	Data rate
X'02'	1 Mb/s
X'04'	2 Mb/s
X'00'	Null terminator

14.9.2.7 aMPDUMaxLength

The aMPDUMaximumLength attribute for the FHSS PHY is defined as the maximum PSDU, in octets, that the PHY shall ever be capable of accepting. This value for the FHSS PHY is set at 4095 octets. The recommended value for maximum PSDU length in an FHSS PHY system is 400 octets at 1 Mb/s and 800 octets at 2 Mb/s, which corresponds to a frame duration less than 3.5 ms. These values are optimized to achieve high performance in a variety of RF channel conditions, particularly with respect to indoor multipath, channel stability for moving STAs, and interference in the 2.4 GHz band.

14.9.2.8 dot11TxAntennaImplemented

The dot11TxAntennaImplemented attribute for the FHSS PHY is defined as a null terminated list of antennas that this implementation can use to transmit data. Table 14-26 shows the possible values appearing in the list, where $N \leq 255$.

Table 14-26—Number of transmit antennas

Code point	Antenna number
X'01'	Tx Antenna 1
X'02'	Tx Antenna 2
X'03'	Tx Antenna 3
…	…
N	Tx Antenna *N*
X'00'	Null terminator

14.9.2.9 dot11CurrentTxAntenna

The dot11CurrentTxAntenna attribute for the FHSS PHY is used to describe the current antenna the implementation is using for transmission. This value should represent one of the antennas appearing in the dot11TxAntennaImplemented list.

14.9.2.10 dot11RxAntennaImplemented

The dot11RxAntennaImplemented attribute for the FHSS PHY is defined as a null terminated list of antennas that this implementation can use to receive data. In the FHSS PHY primitives, one of these values is passed as part of the PHY-RXSTART.indication primitive to the MAC sublayer for every received packet. Table 14-27 shows the possible values appearing in the list, where $N \leq 255$.

Table 14-27—Number of receive antennas

Code point	Antenna number
X'01'	Rx Antenna 1
X'02'	Rx Antenna 2
X'03'	Rx Antenna 3
…	…
N	Rx Antenna *N*
X'00'	Null terminator

14.9.2.11 dot11DiversitySupportImplemented

The dot11DiversitySupportImplemented attribute for the FHSS PHY is used to describe the implementation's diversity support. Table 14-28 shows the possible values appearing in the list.

Table 14-28—Diversity support codes

Code point	Diversity support
X'01'	Diversity available
X'02'	No diversity
X'03'	Control available

The value X'01' indicates that this implementation uses two or more antennas for diversity. The value X'02' indicates that the implementation has no diversity support. The value X'03' indicates that the choice of antennas used during diversity is programmable. (See 14.9.2.12.)

14.9.2.12 dot11DiversitySelectionRxImplemented

The dot11DiversitySelectionRxImplemented attribute for the FHSS PHY is a null terminated list describing the receive antenna or antennas currently in use during diversity and packet reception. Table 14-29 shows the possible values appearing in the list, where $N \leq 255$.

Table 14-29—Diversity select antenna codes

Code point	Antenna number
X'01'	Rx Antenna 1
X'02'	Rx Antenna 2
X'03'	Rx Antenna 3
…	…
N	Rx Antenna *N*
X'00'	Null terminator

The null terminated list enumerates one or more of the receive antennas listed in the dot11RxAntennaImplemented attribute. This attribute can be changed dynamically by the LME.

14.9.2.13 dot11NumberSupportedPowerLevelsImplemented

The dot11NumberSupportedPowerLevelsImplemented attribute for the FHSS PHY describes the number of power levels this implementation supports. This attribute is an integer of value 1 to 8, inclusive.

14.9.2.14 dot11TxPowerLevel1-8

Some implementations may provide up to eight different transmit power levels. The dot11TxPowerLevels attribute for the FHSS PHY is a list of up to eight power levels supported. Table 14-30 describes the list.

Table 14-30—Transmit power levels

Attribute	Power level
TxPowerLevel1	Default setting
TxPowerLevel2	Level 2
TxPowerLevel3	Level 3
TxPowerLevel4	Level 4
TxPowerLevel5	Level 5
TxPowerLevel6	Level 6
TxPowerLevel7	Level 7
TxPowerLevel8	Level 8

14.9.2.15 dot11CurrentTxPowerLevel

The dot11CurrentTxPowerLevel attribute for the FHSS PHY is defined as the current transmit output power level. This level shall be one of the levels implemented in the list of attributes called dot11TxPowerLevel*N* (where *N* is 1–8). This MIB attribute is also used to define the sensitivity of the CCA mechanism when the output power exceeds 100 mW. This MIB attribute is managed by the LME.

14.9.2.16 dot11HopTime

The dot11HopTime attribute for the FHSS PHY describes the time allocated for the PHY to change to a new frequency. For the FHSS PHY, this time period is 224 μs.

14.9.2.17 dot11CurrentChannelNumber

The dot11CurrentChannelNumber attribute for the FHSS PHY is defined as the current operating channel number of the PMD. The values of this attribute correspond to the values shown in Table 14-11. This MIB attribute is managed by the PLME and is updated as the result of a PLMESET.request primitive to dot11CurrentSet, dot11CurrentPattern, or dot11CurrentIndex.

14.9.2.18 dot11MaxDwellTime

The dot11MaxDwellTime attribute for the FHSS PHY is defined as the maximum time the PMD can dwell on a channel and meet the requirements of the current regulatory domain. For the FCC regulatory domain, this number is 390 TU (FCC = 400 ms). The recommended dwell time for the FHSS PHY is 19 TU.

14.9.2.19 dot11CurrentSet

The FHSS PHY contains three sets of hopping patterns. The dot11CurrentSet attribute for the FHSS PHY defines what set the STA is using to determine the hopping pattern. Its value is 1, 2, or 3. This attribute is managed by the PLME. When dot11MultiDomainCapabilityImplemented is true, this value may also be 0. A value of 0 indicates that the hopping pattern is to be obtained from the Hopping Pattern Table element most recently received in a Beacon or Probe Response frame.

14.9.2.20 dot11CurrentPattern

There are up to 78 patterns in each hopping set used by the FHSS PHY. The dot11CurrentPattern attribute for the FHSS PHY defines the *x* value used in Equation (14-1) in 14.7.8 to calculate the current channel number. Its value has various ranges, always within the overall range of 0 to 77, depending on the dot11CurrentRegDomain. This attribute is managed by the PLME.

14.9.2.21 dot11CurrentIndex

The FHSS PHY addresses each channel in the selected hopping pattern through an index. The dot11CurrentIndex attribute for the FHSS PHY defines the *i* value used in the equation for $f_x(i)$ in 14.7.8 to calculate the current channel number. Its value has various ranges, always within the overall range of 1 to 79, depending on the dot11CurrentRegDomain. This attribute is managed by the PLME.

14.9.2.22 dot11CurrentPowerState

The parameter dot11CurrentPowerState defines the operational state of the FHSS PHY. When this attribute has a value of X'01', the PHY is “OFF.” When this attribute has a value of X'02', the PHY is “ON.” This attribute is managed by the PLME.

14.10 FH PHY characteristics

Table 14-31 gives the static FH PHY characteristics, provided through the PLME-CHARACTERISTICS service primitive. The definitions of these characteristics are in 6.5.4.

Table 14-31—FH PHY characteristics

Characteristic	Value	Notes
aSlotTime	50 µs	—
aSIFSTime	28 µs	In order to account for variations between implementations, this value has a tolerance as specified in 9.3.2.3.3.
aCCATime	27 µs	This period includes the aRxRFDelay and the aRxPLCPDelay.
aPHY-RX-START-Delay	128 µs	The delay from the start of the preamble to the issuance of the RX-START.indication primitive by the PHY.
aRxTxTurnaroundTime	20 µs	—
aTxPLCPDelay	1 µs	Implementers may choose to increase or decrease this delay as long as the requirements of aRxTxTurnaroundTime are met.
aRxPLCPDelay	2 µs	Implementers may choose to increase or decrease this delay as long as the requirements of aSIFSTime and aCCATime are met.
aRxTxSwitchTime	10 µs	Implementers may choose to increase or decrease this delay as long as the requirements of aRxTxTurnaroundTime are met.
aTxRampOnTime	8 µs	Implementers may choose to increase or decrease this delay as long as the requirements of aRxTxTurnaroundTime are met.
aTxRampOffTime	8 µs	—
aTxRFDelay	1 µs	Implementers may choose to increase or decrease this delay as long as the requirements of aRxTxTurnaroundTime are met.
aRxRFDelay	4 µs	Implementers may choose to increase or decrease this delay as long as the requirements of aSIFSTime and aCCATime are met.
aAirPropagationTime	1 µs	Variations in the actual propagation time are accounted for in the allowable range of aSIFSTime.

Table 14-31—FH PHY characteristics *(continued)*

Characteristic	Value	Notes
aMACProcessingDelay	2 µs	Implementers may choose to increase or decrease this delay as long as the requirements of aSIFSTime are met.
aPreambleLength	96 µs	—
aPLCPHeaderLength	32 µs	—
aMPDUDurationFactor	31250000	This factor is calculated as $[(33/32) -1] \times 10^9$ to account for the expansion due to the data whitener encoding algorithm.
aMPDUMaxLength	4095	The recommended value for maximum PSDU length in an FHSS PHY system is 400 octets at 1 Mb/s and 800 octets at 2 Mb/s, which corresponds to a frame duration less than 3.5 ms. These values are optimized to achieve high performance in a variety of RF channel conditions, particularly with respect to indoor multipath, channel stability for moving STAs, and interference in the 2.4 GHz band.
aCWmin	15	—
aCWmax	1023	—

15. Infrared (IR) PHY specification

15.1 Status of the Infrared PHY

The mechanisms described in this clause are obsolete. Consequently, this clause may be removed in a later revision of the standard.

This clause is no longer maintained and may not be compatible with all features of this standard.

15.2 Overview

15.2.1 General

The PHY for the infrared (IR) system is specified in this clause. The IR PHY uses near-visible light in the 850 nm to 950 nm range for signaling. This is similar to the spectral usage of both common consumer devices such as IR remote controls, as well as other data communications equipment, such as IR data association (IrDA) devices.

Unlike many other IR devices, however, the IR PHY is not directed. That is, the receiver and transmitter do not have to be aimed at each other and do not need a clear line-of-sight. This permits the construction of a true LAN system, whereas with an aimed system, it would be difficult or impossible to install a LAN because of physical constraints.

A pair of conformant IR devices would be able to communicate in a typical environment at a range up to about 10 m. This standard allows conformant devices to have more sensitive receivers, and this may increase range up to about 20 m.

The IR PHY relies on both reflected IR energy as well as line-of-sight IR energy for communications. Most designs anticipate that *all* of the energy at the receiver is reflected energy. This reliance on reflected IR energy is called *diffuse IR* transmission.

This standard specifies the transmitter and receiver in such a way that a conformant design operates well in most environments where there is no line-of-sight path from the transmitter to the receiver. However, in an environment that has few or no reflecting surfaces, and where there is no line-of-sight, an IR PHY system may suffer reduced range.

The IR PHY operates only in indoor environments. IR radiation does not pass through walls, and is significantly attenuated passing through most exterior windows. This characteristic can be used to “contain” an IR PHY in a single physical room, like a classroom or conference room. Different LANs using the IR PHY can operate in adjacent rooms separated only by a wall without interference, and without the possibility of eavesdropping.

At the time of this standard’s preparation, the only known regulatory standards that apply to the use of IR radiation are safety regulations, such as IEC 60825-1:1993 [B18] and ANSI Z136.1-1993 [B5]. While a conformant IR PHY device can be designed to also comply with these safety standards, conformance with this standard does not ensure conformance with other standards.

Worldwide, no frequency allocation or bandwidth allocation regulatory restrictions currently exist on IR emissions.

Emitter (typically LED) and detector (typically PIN diode) devices for IR communications are relatively inexpensive at the IR wavelengths specified in the IR PHY, and at the electrical operating frequencies required by this PHY.

While many other devices in common use also use IR emissions in the same optical band, these devices usually transmit IR intermittently and do not interfere with the proper operation of a compliant IR PHY. If such a device does interfere, by transmitting continuously and with a very strong signal, it can be physically isolated (placing it in a different room) from the IEEE 802.11 LAN.

15.2.2 Scope

The PHY services provided to the IEEE 802.11 WLAN MAC by the IR system are described in this clause. The IR PHY consists of two protocol functions as follows:

a) A PHY convergence function, which adapts the capabilities of the PMD system to the PHY service. This function is supported by the PLCP, which defines a method of mapping the IEEE 802.11 MPDUs into a framing format suitable for sending and receiving user data and management information between two or more STAs using the associated PMD system.

b) A PMD system, whose function defines the characteristics of, and method of transmitting and receiving data through, the WM between two or more STAs.

15.2.3 IR PHY functions

15.2.3.1 General

The IR PHY contains three functional entities: the PMD function, the PHY convergence function, and the layer management function. Each of these functions is described in detail below.

The IR PHY service is provided to the MAC entity at the STA through a SAP as described in Clause 7. For a visual guide to the relationship of the IR PHY to the remainder of the system, refer to Figure 4-14 (in 4.9).

15.2.3.2 PLCP sublayer

To allow the IEEE 802.11 MAC to operate with minimum dependence on the PMD sublayer, a PLCP sublayer is defined. This function simplifies the PHY service interface to the IEEE 802.11 MAC services. The PHY-specific preamble is normally associated with this convergence layer.

15.2.3.3 PMD sublayer

The PMD sublayer provides a CCA mechanism, transmission mechanism, and reception mechanism that are used by the MAC via the PLCP to send or receive data between two or more STAs.

15.2.3.4 PLME

The PLME performs management of the local PHY functions in conjunction with the MLME. See 15.5 for the MIB variables that may be accessed by the PHY entities and intralayer of higher level LMEs. These variables are accessed via the PLME-GET, PLME-SET, and PLME-RESET primitives defined in Clause 6.

15.2.4 Service specification method and notation

The models represented by figures and state diagrams are intended as illustrations of functions provided. It is important to distinguish between a model and a real implementation. The models are optimized for simplicity and clarity of presentation; the actual method of implementation is left to the discretion of the

IEEE 802.11 IR-PHY-compliant developer. Conformance to this standard is not dependent on following the model, and an implementation that follows the model closely may not be conformant.

Abstract services are specified here by describing the service primitives and parameters that characterize each service. This definition is independent of any particular implementation. In particular, the PHY-SAP operations are defined and described as instantaneous; however, this may be difficult to achieve in an implementation.

15.3 IR PLCP sublayer

15.3.1 General

While the PLCP sublayer and the PMD sublayer are described separately, the separation and distinction between these sublayers is artificial, and is not meant to imply that the implementation separates these functions. This distinction is made primarily to provide a point of reference from which to describe certain functional components and aspects of the PMD. The functions of the PLCP can be subsumed by a PMD sublayer; in this case, the PMD incorporates the PHY-SAP as its interface, and does not offer a PMD_SAP.

15.3.2 Overview

A convergence procedure is provided by which MPDUs are converted to and from PPDUs. During transmission, the MPDU (PSDU) is prepended with a PLCP preamble and PLCP header to create the PPDU. At the receiver, the PLCP preamble is processed and the internal data fields are processed to aid in demodulation and delivery of the MPDU (PSDU).

15.3.3 PLCP frame format

Figure 15-1 shows the format for the PPDU including the PLCP preamble, the PLCP header, and the PSDU. The PLCP preamble contains the following fields: SYNC and SFD. The PLCP header contains the following fields: data rate (DR), dc level adjustment (DCLA), length (LENGTH), and CRC. Each of these fields is described in detail in 15.3.5.

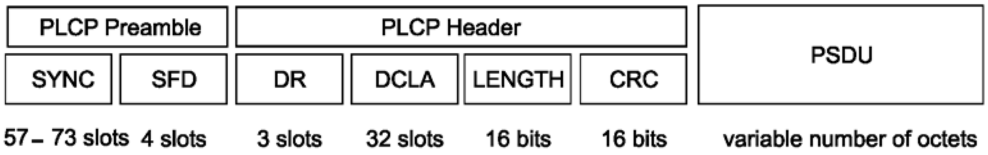


Figure 15-1—PPDU frame format

15.3.4 PLCP modulation and rate change

The PLCP preamble shall be transmitted using the basic pulse defined in 15.4.4.3. The PSDU, LENGTH, and CRC fields shall be transmitted using pulse position modulation (PPM). PPM maps bits in the octet into symbols: 16-PPM maps four bits into a 16-position symbol, and 4-PPM maps two bits into a 4-position symbol. The basic L-PPM TU is the slot. A slot corresponds to one of the L positions of a symbol and has a 250 ns duration. The PSDU, LENGTH, and CRC fields are transmitted at one of two bit rates: 1 Mb/s or 2 Mb/s. The DR field indicates the data rate that is used to transmit the PSDU, LENGTH, and CRC fields. The 1 Mb/s data rate uses 16-PPM (basic access rate), and the 2 Mb/s data rate uses 4-PPM (enhanced access rate). The transmitter and receiver shall initiate the modulation or demodulation indicated by the DR field starting with the first 4 bits (in 16-PPM) or 2 bits (in 4-PPM) of the LENGTH field. The PSDU transmission rate is set by the DATARATE parameter in the PHY-TXSTART.request primitive. Any conformant IR PHY shall be capable of receiving at 1 Mb/s and 2 Mb/s. Transmission at 2 Mb/s is optional.

A PHY-TXSTART.request primitive that specifies a data rate that is not supported by a PHY instance causes the PHY to indicate an error to its MAC instance. A PHY is not permitted under any circumstance to transmit at a different rate than the requested rate.

15.3.5 PLCP field definitions

15.3.5.1 PLCP SYNC field

The SYNC field consists of a sequence of alternated presence and absence of a pulse in consecutive slots. The SYNC field has a minimum length of 57 L-PPM slots and a maximum length of 73 L-PPM slots and shall terminate with the absence of a pulse in the last slot. This field is provided so that the receiver can perform clock recovery (slot synchronization), automatic gain control (optional), signal-to-noise ratio estimation (optional), and diversity selection (optional).

The SYNC field is not modulated using L-PPM, but instead consists of transitions in L-PPM slots that would otherwise constitute an illegal symbol. See 15.4.3.2 for legal symbols.

15.3.5.2 PLCP SFD field

The SFD field length is four L-PPM slots and consists of the binary sequence 1001, where 1 indicates a pulse in the L-PPM slot and 0 indicates no pulse in the L-PPM slot. The leftmost bit shall be transmitted first. The SFD field is provided to indicate the start of the PLCP preamble and to perform bit and symbol synchronization.

The SFD field is not modulated using L-PPM, but instead consists of transitions in L-PPM slots that would otherwise constitute an illegal symbol.

15.3.5.3 PLCP DR field

The DR field indicates to the PHY the data rate that shall be used for the transmission or reception of the PSDU, LENGTH, and CRC fields. The transmitted value shall be provided by the PHY-TXSTART.request primitive as described in Clause 7. The DR field has a length of three L-PPM slots. The leftmost bit, as shown below, shall be transmitted first. The IR PHY currently supports two data rates defined by the slot pattern shown for the three L-PPM slots following the SFD, where 1 indicates a pulse in the L-PPM slot and 0 indicates no pulse in the L-PPM slot:

1 Mb/s:	000
2 Mb/s:	001

The DR field is not modulated using L-PPM, but instead consists of transitions in L-PPM slots that would otherwise constitute an illegal symbol.

15.3.5.4 PLCP DCLA field

The DCLA field is required to allow the receiver to stabilize the dc level after the SYNC, SFD, and DR fields. The leftmost bit, as shown below, shall be transmitted first. The length of the DCLA field is 32 L-PPM slots and consists of the contents shown, where 1 indicates a pulse in the L-PPM slot and 0 indicates no pulse in the L-PPM slot:

1 Mb/s:	00000000100000000000000010000000
2 Mb/s:	00100010001000100010001000100010

The DCLA field is not modulated using L-PPM, but instead consists of transitions in L-PPM slots that would otherwise constitute an illegal symbol.

15.3.5.5 PLCP LENGTH field

The LENGTH field is an unsigned 16-bit integer that indicates the number of octets to be transmitted in the PSDU. The transmitted value shall be provided by the PHY-TXSTART.request primitive as described in Clause 7. The LSB shall be transmitted first. This field is modulated and sent in L-PPM format. This field is protected by the CRC described in 15.3.5.6.

15.3.5.6 PLCP CRC field

The LENGTH field shall be protected by a 16-bit CRC. The CRC is the ones complement of the remainder generated by the modulo 2 division of the LENGTH field by the polynomial:

$$x^{16} + x^{12} + x^5 + 1$$

The protected bits are processed in transmit order. The MSB of the 16-bit CRC shall be transmitted first. This field shall be modulated and sent in L-PPM format. All CRC calculations shall be made prior to L-PPM encoding on transmission and after L-PPM decoding on reception.

15.3.5.7 PSDU field

This field is composed of a variable number of octets. The minimum is 0 (zero) and the maximum is 2500. The LSB of each octet shall be transmitted first. All the octets of this field shall be modulated and sent in L-PPM format.

15.3.6 PLCPs

15.3.6.1 Transmit PLCP

All commands issued by the MAC require that a confirmation primitive be issued by the PHY. The confirmation primitives provide flow control between the MAC and the PHY.

The transmit PLCP is as follows:

a) Based on the status of CCA, the MAC shall determine whether the channel is clear.

b) If the channel is clear, transmission of the PSDU shall be initiated by a PHY-TXSTART.request primitive with parameters LENGTH and DATARATE.

c) The PHY entity shall immediately initiate transmission of the PLCP preamble and PLCP header based on the LENGTH and DATARATE parameters passed in the PHY-TXSTART.request primitive. Once the PLCP preamble and PLCP header transmission is completed, the PHY entity shall issue a PHY-TXSTART.confirm primitive.

d) Each octet of the PSDU is passed from the MAC to the PHY by a single PHY-DATA.request primitive. Each PHY-DATA.request primitive shall be confirmed by the PHY with a PHY-DATA.confirm primitive before the next request can be made.

e) At the PHY each PSDU octet shall be divided into symbols of 2 bits or 4 bits each. The symbols shall be modulated using L-PPM and transmitted into the medium.

f) Transmission is terminated by the MAC through the PHY-TXEND.request primitive. The PHY shall confirm the resulting end of transmission with a PHY-TXEND.confirm primitive.

15.3.6.2 Receive PLCP

The receive PLCP is as follows:

a) CCA is provided to the MAC via the PHY-CCA.indication primitive. When the PHY senses activity on the medium, it shall indicate that the medium is busy with a PHY-CCA.indication primitive with a value of BUSY. This normally occurs during the SYNC field of the PLCP preamble.

b) The PHY entity shall begin searching for the SFD field. Once the SFD field is detected, the PHY entity shall attempt to receive the PLCP header. After receiving the DR and DCLA fields, the PHY shall initiate processing of the received CRC and LENGTH fields. The data rate indicated in the DR field applies to all symbols in the latter part of the received PSDU, commencing with the first symbol of the LENGTH field. The CRC shall be checked for correctness immediately after its reception.

c) If the CRC check fails, or the value received in the DR field is not one supported by the PHY, then a PHY-RXSTART.indication primitive shall not be issued to the MAC. When the medium is again free, the PHY shall issue a PHY-CCA.indication primitive with a value of IDLE.

d) If the PLCP preamble and PLCP header reception is successful, the PHY shall send a PHY-RXSTART.indication primitive to the MAC; this includes the parameters DATARATE and LENGTH.

   In the absence of errors, the receiving PHY shall report the same length to its local MAC, in the RXVECTOR parameter of the PHY-RXSTART.indication primitive, that the peer MAC presented to its local PHY entity in the TXVECTOR parameter of its respective PHY-TXSTART.request primitive.

e) The received PSDU L-PPM symbols shall be assembled into octets and presented to the MAC using a series of PHY-DATA.indication primitives, one per octet.

f) Reception shall be terminated after the reception of the final symbol of the last PSDU octet indicated by the PLCP header's LENGTH field. After the PHY-DATA.indication primitive for that octet is issued, the PHY shall issue a PHY-RXEND.indication primitive to its MAC.

g) After issuing the PHY-RXEND.indication primitive, and when the medium is no longer busy, the PHY shall issue a PHY-CCA.indication primitive with a value of IDLE.

15.3.6.3 CCA procedure

CCA is provided to the MAC via the PHY-CCA.indication primitive.

The CCA procedure is as follows:

a) When the PHY senses activity on the medium, a PHY-CCA.indication primitive with a value of BUSY shall be issued. This normally occurs during reception of the SYNC field of the PLCP preamble.

b) When the PHY senses that the medium is free, a PHY-CCA.indication primitive with a value of IDLE shall be issued.

c) At any time, the MAC may issue a PHY-CCARESET.request primitive, which resets the PHY's internal CCA detection mechanism to the medium not-busy (IDLE) state. This primitive is acknowledged with a PHY-CCARESET.confirm primitive.

15.3.6.4 PMD_SAP peer-to-peer service primitive parameters

Several service primitives include a parameter vector. This vector shall be a list of parameters that may vary depending on PHY type. Table 15-1 indicates the parameters required by the MAC or IR PHY in each of the parameter vectors used for peer-to-peer interactions.

Table 15-1—IR PMD_SAP peer-to-peer service primitives

Parameter	Associated primitive	Value
LENGTH	RXVECTOR, TXVECTOR	4 to $2^{16} – 1$
DATARATE	RXVECTOR, TXVECTOR	PHY dependent

15.4 IR PMD sublayer

15.4.1 General

The IR PMD sublayer does not define PMD SAPs. The mechanism for communications between the PLCP and PMD sublayers, as well as the distinction between these two sublayers, if any, is left to implementers. In particular, it is possible to design and implement, in a conformant way, a single sublayer that subsumes the functions of both the PLCP and PMD, presenting only the PHY-SAP.

15.4.2 Overview

The PMD functional, electrical, and optical characteristics required for interoperability of implementations conforming to this specification are described in 15.4. The relationship of this specification to the entire IR PHY is shown in Figure 4-14 (in 4.9).

15.4.3 PMD operating specifications, general

15.4.3.1 General

General specifications for the IR PMD sublayer are provided in 15.4.3. These specifications apply to both the receive and transmit functions and general operation of a compliant IR PHY.

15.4.3.2 Modulation and channel data rates

Two modulation formats and data rates are specified for the IR PHY: a *basic access rate* and an *enhanced access rate*. The basic access rate is based on 1 Mb/s 16-PPM modulation. The 16-PPM encoding is specified in Table 15-2. Each group of 4 data bits is mapped to one of the 16-PPM symbols. The enhanced access rate is based on 2 Mb/s 4-PPM. The 4-PPM encoding is specified in Table 15-3. Each group of 2 data bits is mapped to one of the 4-PPM symbols. Transmission order of the symbol slots is from left to right, as shown in the table, where a 1 indicates in-band energy in the slot, and a 0 indicates the absence of in-band energy in the slot.

The data in Table 15-2 and Table 15-3 have been arranged (Gray coded) so that a single out-of-position-by-one error in the medium, caused, for example, by intersymbol interference, results in only a single bit error in the received data, rather than in a multiple bit error.

Table 15-2—Sixteen-PPM basic rate mapping

Data	16-PPM symbol
0000	0000000000000001
0001	0000000000000010
0011	0000000000000100
0010	0000000000001000
0110	0000000000010000

Table 15-2—Sixteen-PPM basic rate mapping *(continued)*

Data	16-PPM symbol
0111	0000000000100000
0101	0000000001000000
0100	0000000010000000
1100	0000000100000000
1101	0000001000000000
1111	0000010000000000
1110	0000100000000000
1010	0001000000000000
1011	0010000000000000
1001	0100000000000000
1000	1000000000000000

Table 15-3—Four-PPM enhanced rate mapping

Data	4-PPM symbol
00	0001
01	0010
11	0100
10	1000

15.4.3.3 Octet partition and PPM symbol generation procedure

Because PPM is a block modulation method, with the block size less than a full octet, octets have to be partitioned prior to modulation (mapping into PPM symbols).

Octet partition depends on the PPM order being used.

Assume an octet is formed by eight bits numbered 7 6 5 4 3 2 1 0, where bit 0 is the LSB. Partition the octet as follows:

For 16-PPM, create two PPM symbols:

- The symbol using bits 3 2 1 0 shall be transmitted onto the medium first.
- The symbol using bits 7 6 5 4 shall be transmitted onto the medium last.

For 4-PPM, create four PPM symbols:

- The symbol using bits 1 0 shall be transmitted onto the medium first.
- The symbol using bits 3 2 shall be transmitted onto the medium second.
- The symbol using bits 5 4 shall be transmitted onto the medium third.
- The symbol using bits 7 6 shall be transmitted onto the medium last.

15.4.3.4 Operating environment

The IR PHY operates only in indoor environments. IR PHY interfaces cannot be exposed to direct sunlight. The IR PHY relies on reflected IR energy and does not require a line-of-sight between emitter and receiver in order to work properly. The range and bit error rate of the system may vary with the geometry of the environment and with natural and artificial illumination conditions.

15.4.4 PMD transmit specifications

15.4.4.1 Introduction

Subclause 15.4.4 describes the transmit functions and parameters associated with the PMD sublayer.

15.4.4.2 Transmitted peak optical power

The peak optical power of an emitted pulse shall be as specified in Table 15-4.

Table 15-4—Peak optical power as a function of emitter radiation pattern mask

Emitter radiation pattern mask	Peak optical power
Mask 1	2 W ±20%
Mask 2	0.55 W ±20%

15.4.4.3 Basic pulse shape and parameters

The basic pulse width, measured between the 50% amplitude points, shall be 250 ±10 ns. The pulse rise time, measured between the 10% and 90% amplitude points, shall be no more than 40 ns. The pulse fall time, measured between the 10% and 90% amplitude points, shall be no more than 40 ns. The edge jitter, defined as the absolute deviation of the edge from its correct position, shall be no more than 10 ns. The basic pulse shape is shown in Figure 15-2.

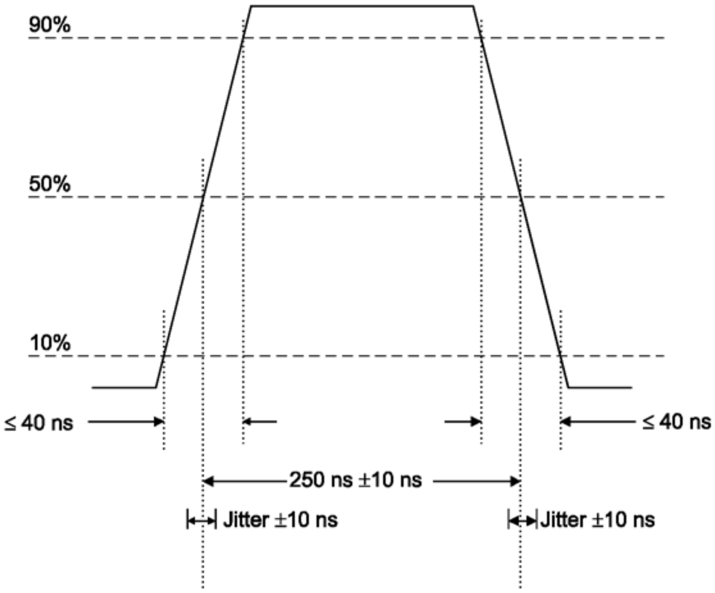


Figure 15-2—Basic pulse shape

15.4.4.4 Emitter radiation pattern mask

The standard contains two emitter radiation pattern masks. Mask 1 is defined in Table 15-5 and illustrated in Figure 15-3.

Table 15-5—Definition of the emitter radiation pattern Mask 1

Declination angle	Normalized irradiance
$\alpha \le 60°$	$> 3.5 \times 10^{-6}$
$\alpha \le 29°$	$\le 2.2 \times 10^{-5}$
$29° < \alpha \le 43°$	$\le -1.06 \times 10^{-4} + (0.44 \times 10^{-5})\ \alpha$
$43° < \alpha \le 57°$	$\le 1.15 \times 10^{-4} - (7.1 \times 10^{-7})\ \alpha$
$57° < \alpha \le 74°$	$\le 2.98 \times 10^{-4} - (3.9 \times 10^{-6})\ \alpha$
$74° < \alpha \le 90°$	$\le 4.05 \times 10^{-5} - (4.5 \times 10^{-7})\ \alpha$

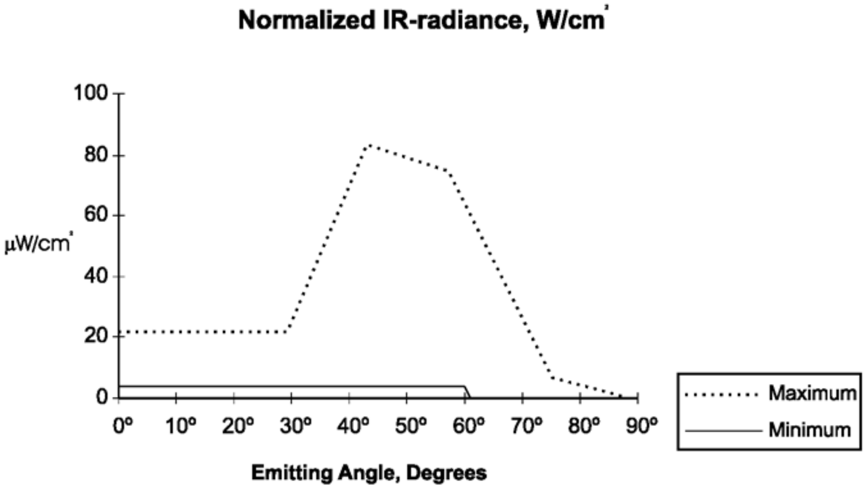


Figure 15-3—Emitter radiation pattern Mask 1

Mask 2 is defined in Table 15-6 and illustrated in Figure 15-4.

Table 15-6—Definition of emitter radiation pattern Mask 2

Declination angle	Pitch angle	Normalized irradiance
$\alpha \le 60$	$\alpha = 0$	0.05 ± 15%
$\alpha \le 90$	$\alpha = 0$	0.025 ± 15%
$\alpha \ge 100$	$\alpha = 0$	≤ 0.015
$0 \le \alpha \le 60$	$0 \le \alpha \le 10$	$0.035 \le I \le 0.055$
$0 \le \alpha \le 60$	$10 \le \alpha \le 20$	$0.0225 \le I \le 0.05$
$0 \le \alpha \le 60$	$\alpha \ge 30$	≤ 0.015

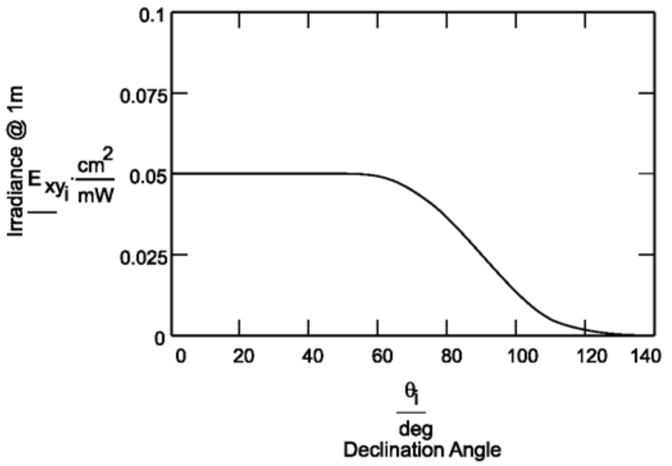


Figure 15-4—Emitter radiation pattern Mask 2

Following is a description of how to interpret the Mask 1 table and figure. Position the conformant Mask 1 device in its recommended attitude. Define the conformant Mask 1 device axis as the axis passing through the emitter center and having the direction perpendicular to the floor. The mask represents the irradiance normalized to the total peak emitted power, as a function of the angle between the conformant Mask 1 device axis and the axis from the emitter center to the test receiver center (declination angle). The distance between emitter and test receiver is 1 m. The test receiver normal is always aimed at the emitter center. The azimuth angle is a rotation angle on the conformant device axis.

A device is conformant if for any azimuth angle its radiation pattern as a function of declination angle falls within the pattern mask.

Figure 15-5 is a description of how to interpret the Mask 2 table with reference to Figure 15-4.

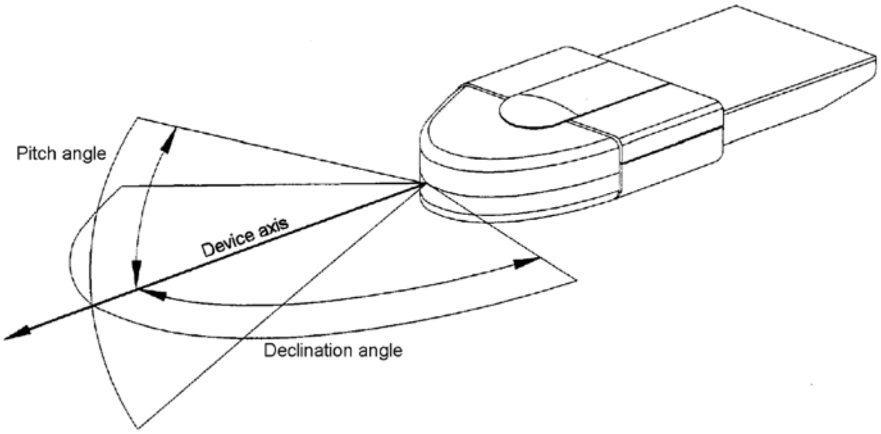


Figure 15-5—Mask 2 device orientation drawing

Position the conformant Mask 2 device in its recommended attitude. Define the conformant Mask 2 device axis as passing through the emitter center and having the direction relative to the device as defined by the manufacturer. The declination angle plane is as defined by the manufacturer. The mask represents the irradiance normalized to the peak emitted power on the conformant Mask 2 device axis, as a function of the angle between the conformant device axis and the axis from the emitter center to the test receiver center (declination angle) in the declination plane. The distance between emitter and test receiver is 1 m. The test receiver normal is always aimed at the emitter center. The pitch angle is an angle relative to the conformant device axis which is perpendicular to the declination plane.

The device is conformant if, for a pitch angle of 0 degrees, at any declination angle from 0 to 100 degrees, and if, for any declination angle from 0 to 60 degrees, at any pitch angle from 0 to 20 degrees, its radiation pattern as a function of angle falls within the pattern mask.

Other radiation patterns are for future study.

15.4.4.5 Optical emitter peak wavelength

The optical emitter peak wavelength shall be between 850 nm and 950 nm.

15.4.4.6 Transmit spectrum mask

Define the transmit spectrum of a transmitter as the Fourier Transform, or equivalent, of a voltage (or current) signal whose amplitude, as a function of time, is proportional to the transmitted optical power.

The transmit spectrum of a conformant transmitter shall be 20 dB below its maximum for all frequencies above 15 MHz. The transmit spectrum mask is shown in Figure 15-6.

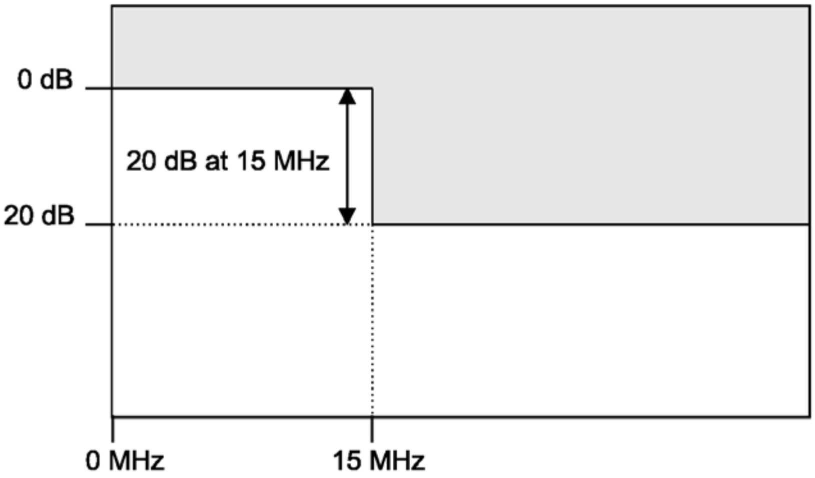


Figure 15-6—Transmit spectrum mask

15.4.5 PMD receiver specifications

15.4.5.1 Introduction

Subclause 15.4.5 describes the receive functions and parameters associated with the PMD sublayer.

15.4.5.2 Receiver sensitivity

The receiver sensitivity, defined as the minimum irradiance (in mW/cm^2) at the photodetector plane required for a FER of 4×10^{-5} with a PSDU of 512 octets and with an unmodulated background IR source between 800 nm and 1000 nm with a level of 0.1 mW/cm^2, shall be

1 Mb/s: 2×10^{-5}mW/cm^2
2 Mb/s: 8×10^{-5}mW/cm^2

15.4.5.3 Receiver dynamic range

The receiver dynamic range, defined as the ratio between the maximum and minimum irradiance at the plane normal to the receiver axis that assures an FER lower than or equal to 4×10^{-5} with a PSDU of 512 octets and with an unmodulated background IR source between 800 nm and 1000 nm with a level of 0.1 mW/cm^2, shall be ≥ 30 dB.

15.4.5.4 Receiver field of view (FOV)

The receiver axis is defined as the direction of incidence of the optical signal at which the received optical power is maximum.

The received optical power shall be greater than the values given in Table 15-7, at the angles indicated, where “angle of incidence” is the angle of the optical signal relative to the receiver axis, and “received power” is the received optical power as a percentage of that measured at the receiver axis.

Table 15-7—Definition of the receiver FOV

Angle of incidence	Received power
$\alpha \leq 20°$	≥ 65%
$\alpha \leq 40°$	≥ 55%
$\alpha \leq 60°$	≥ 35%
$\alpha \leq 80°$	≥ 10%

15.4.6 ED, CS, and CCA definitions

15.4.6.1 ED signal

The ED signal shall be set true when IR energy variations in the band between 1 MHz and 10 MHz exceed 0.001 mW/cm^2.

The ED shall operate independently of the CS. The ED shall not be asserted at the minimum signal level specified in 15.4.5.2, which is below the level specified in 15.4.6.

This signal is not directly available to the MAC.

15.4.6.2 CS signal

The CS shall be asserted by the PHY when it detects and locks onto an incoming PLCP preamble signal. Conforming PHYs shall assert this condition within the first 12 µs of signal reception, at the minimum signal level equal to the receiver sensitivity specified in 15.4.5.2, with a background IR level as specified in 15.4.5.2.

The CS shall be deasserted by the PHY when the receiving conformant device loses carrier lock.

NOTE—The 12 µs specification is somewhat less than the minimum length of the PLCP SYNC interval, which is 14.25 µs.

The CS shall operate independently of the ED and shall not require a prior ED before the acquisition and assertion of CS. This permits reception of signals at the minimum signal level specified in 15.4.5.2, even though these signals fall below the ED level.

This signal is not directly available to the MAC.

15.4.6.3 CCA

CCA shall be asserted IDLE by the PHY when the CS and the ED are both false, or when ED has been continuously asserted for a period of time defined by the product of dot11CCAWatchdogTimerMax and dot11CCAWatchdogCountMax without CS becoming active. When either CS or ED go true, CCA is indicated as BUSY to the MAC via the primitive PHY-CCA.indication primitive. CS and ED behavior are defined in 15.4.6.2.

Normally, CCA is held BUSY throughout the period of the PLCP header. After receiving the last PLCP bit and the first data octet, the PHY shall signal PHY-RXSTART.indication primitive with the parameters LENGTH and RATE. CCA shall be held BUSY until the number of octets specified in the decoded PLCP header are received. At that time the PHY shall signal PHY-RXEND.indication primitive. The CCA may remain BUSY after the end of data if some form of energy is still being detected. The PHY signals PHY-CCA.indication primitive with a value of IDLE only when the CCA goes CLEAR.

The transition of CCA from BUSY to IDLE is indicated to the MAC via the primitive PHY-CCA.indication primitive.

If CS and ED go false before the PHY generates a PHY-RXSTART.indication primitive, CCA is set to IDLE and *immediately* signaled to the MAC via a PHY-CCA.indication primitive with a value of IDLE. If CS and ED go false after the PHY has signaled PHY-RXSTART.indication, implying that the PLCP header has been properly decoded, then the PHY shall not signal a change in state of CCA until the proper interval has passed for the number of octets indicated by the received PLCP LENGTH field. At that time, the PHY shall generate a PHY-RXEND.indication primitive with an RXERROR parameter of CarrierLost followed by a PHY-CCA.indication primitive with a value of IDLE.

The transition of CCA from CLEAR to BUSY resets the CCA watchdog timer and CCA watchdog counter. dot11CCAWatchdogTimerMax and dot11CCAWatchdogCountMax are parameters available via MIB entries and can be read and set via the LME.

Rise and fall times of CCA relative to the OR’ing of the CS and ED signals shall be less than 30 ns. CS and ED are both internal signals to the PHY and are not available directly to the MAC, nor are they defined at any exposed interface.

15.4.6.4 CHNL_ID

For the IR PHY, CHNL_ID = X'01' is defined as the baseband modulation method. All other values are not defined.

15.5 PHY attributes

PHY attributes have allowed values and default values that are PHY dependent. Table 15-8 and Table 15-9 describe those values, and further specify whether they are permitted to vary from implementation to implementation.

Table 15-8 does not provide the definition of the attributes, but only provides the IR PHY-specific values for the attributes whose definitions are in C.

Table 15-8—IR PHY MIB attributes

PHY MIB object	Default value	Operational semantics	Operational behavior
dot11CCAWatchdogTimerMax	Implementation dependent	Dynamic	A conformant PHY may set this via the LME
dot11CCAWatchdogCountMax	Implementation dependent	Dynamic	A conformant PHY may set this via the LME
dot11CCAWatchdogTimerMin	22 µs	Static	Identical for all conformant PHYs
dot11CCAWatchdogCountMin	1	Static	Identical for all conformant PHYs
dot11SupportedDataRatesTx	Implementation dependent	Static	All conformant PHYs shall include the value X'02' (1 Mb/s).
dot11SupportedDataRatesRx	Implementation dependent	Static	All conformant PHYs shall include the values X'02' (1 Mb/s) and X'04' (2 Mb/s).
dot11PhyType	03	Static	Identical for all conformant PHYs

The static IR PHY characteristics, provided through the PLME-CHARACTERISTICS service primitive, are shown in Table 15-9. The definitions of these characteristics are in 6.5.4.

Table 15-9—IR PHY characteristics

Characteristic	Value
aSlotTime	8 µs
aSIFSTime	10 µs
aCCATime	5 µs
aPHY-RX-START-Delay	57 µs
aRxTxTurnaroundTime	0 µs
aTxPLCPDelay	Implementers may choose any value for this delay as long as the requirements of aRxTxTurnaroundTime are met.
aRxPLCPDelay	1 µs
aRxTxSwitchTime	0 µs
aTxRampOnTime	0 µs
aTxRampOffTime	0 µs
aTxRFDelay	Implementers may choose any value for this delay as long as the requirements of aRxTxTurnaroundTime are met.
aRxRFDelay	Implementers may choose any value for this delay as long as the requirements of aSIFSTime and aCCATime are met.
aAirPropagationTime	1 µs
aMACProcessingDelay	2 µs
aPreambleLength	16 µs (1 Mb/s) 20 µs (2 Mb/s)
aPLCPHeaderLength	41 µs (1 Mb/s) 25 µs (2 Mb/s)
aMPDUDurationFactor	0
aMPDUMaxLength	2500
aCWmin	63
aCWmax	1023

16. DSSS PHY specification for the 2.4 GHz band designated for ISM applications

16.1 Overview

16.1.1 General

The PHY for the DSSS system is described in this clause. The RF LAN system is aimed for the 2.4 GHz band designated for ISM applications.

The DSSS system provides a WLAN with both a 1 Mb/s and a 2 Mb/s data payload communication capability. The DSSS system uses baseband modulations of differential binary phase shift keying (DBPSK) and differential quadrature phase shift keying (DQPSK) to provide the 1 Mb/s and 2 Mb/s data rates, respectively.

16.1.2 Scope

The PHY services provided to the IEEE 802.11 WLAN MAC by the 2.4 GHz DSSS system are described in this clause. The DSSS PHY consists of two protocol functions:

a) A PHY convergence function, which adapts the capabilities of the PMD system to the PHY service. This function shall be supported by the PLCP, which defines a method of mapping the IEEE 802.11 MPDUs into a framing format suitable for sending and receiving user data and management information between two or more STAs using the associated PMD system.
b) A PMD system, whose function defines the characteristics of, and method of transmitting and receiving data through, a WM between two or more STAs each using the DSSS system.

16.1.3 DSSS PHY functions

16.1.3.1 General

The 2.4 GHz DSSS PHY architecture is depicted in the reference model shown in Figure 4-14 (in 4.9). The DSSS PHY contains three functional entities: the PMD function, the PHY convergence function, and the layer management function. Each of these functions is described in detail in the following subclauses.

The DSSS PHY service shall be provided to the MAC through the PHY service primitives described in Clause 7.

16.1.3.2 PLCP sublayer

To allow the IEEE 802.11 MAC to operate with minimum dependence on the PMD sublayer, a PLCP sublayer is defined. This function simplifies the PHY service interface to the IEEE 802.11 MAC services.

16.1.3.3 PMD sublayer

The PMD sublayer provides a means to send and receive data between two or more STAs. This clause is concerned with the 2.4 GHz ISM bands using direct sequence modulation.

16.1.3.4 PLME

The PLME performs management of the local PHY functions in conjunction with the MLME.

16.1.4 Service specification method and notation

The models represented by figures and state diagrams are intended to be illustrations of functions provided. It is important to distinguish between a model and a real implementation. The models are optimized for simplicity and clarity of presentation; the actual method of implementation is left to the discretion of the IEEE 802.11 DSSS-PHY-compliant developer.

The service of a layer or sublayer is a set of capabilities that it offers to a user in the next-higher layer (or sublayer). Abstract services are specified here by describing the service primitives and parameters that characterize each service. This definition is independent of any particular implementation.

16.2 DSSS PLCP sublayer

16.2.1 Overview

Subclause 16.2 provides a convergence procedure in which MPDUs are converted to and from PPDUs. During transmission, the MPDU shall be prepended with a PLCP preamble and header to create the PPDU. At the receiver, the PLCP preamble and header are processed to aid in demodulation and delivery of the MPDU.

16.2.2 PLCP frame format

Figure 16-1 shows the format for the PPDU including the DSSS PLCP preamble, the DSSS PLCP header, and the MPDU. The PLCP preamble contains the following fields: SYNC and SFD. The PLCP header contains the following fields: IEEE 802.11 Signaling (SIGNAL), IEEE 802.11 Service (SERVICE), length (LENGTH), and CRC-16 (CRC). Each of these fields is described in detail in 16.2.3.

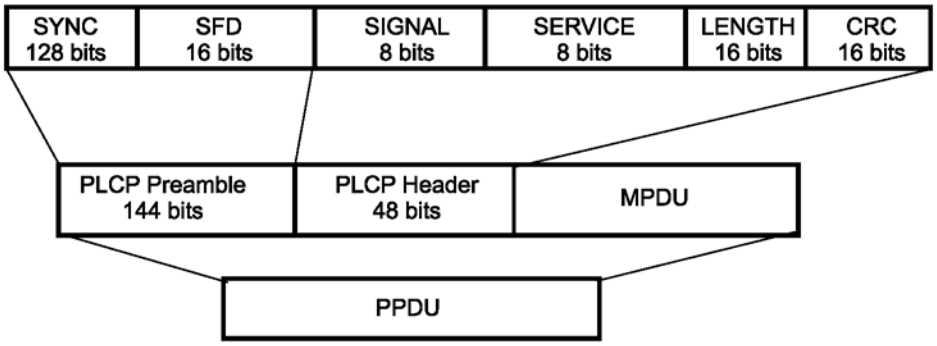


Figure 16-1—PLCP frame format

16.2.3 PLCP field definitions

16.2.3.1 General

The entire PLCP preamble and header shall be transmitted using the 1 Mb/s DBPSK modulation described in 16.4.7. All transmitted bits shall be scrambled using the feedthrough scrambler described in 16.2.4.

16.2.3.2 PLCP SYNC field

The SYNC field shall consist of 128 bits of scrambled ones. This field shall be provided so that the receiver can perform the necessary operations for synchronization.

16.2.3.3 PLCP SFD

The SFD shall be provided to indicate the start of PHY-dependent parameters within the PLCP preamble. The SFD shall be a 16-bit field, X'F3A0' (MSB to LSB). The LSB shall be transmitted first in time.

16.2.3.4 PLCP IEEE 802.11 SIGNAL field

The 8-bit IEEE 802.11 SIGNAL field indicates to the PHY the modulation that shall be used for transmission (and reception) of the MPDU. The data rate shall be equal to the signal field value multiplied by 100 kbit/s. The DSSS PHY currently supports two mandatory modulation services given by the following 8-bit words, where the LSB shall be transmitted first in time:

a) X'0A' (MSB to LSB) for 1 Mb/s DBPSK
b) X'14' (MSB to LSB) for 2 Mb/s DQPSK

The DSSS PHY rate change capability is described in 16.2.5. This field shall be protected by the CRC-16 FCS described in 16.2.3.7.

16.2.3.5 PLCP IEEE 802.11 SERVICE field

The 8-bit IEEE 802.11 SERVICE field shall be reserved for future use. The LSB shall be transmitted first in time. This field shall be protected by the CRC-16 FCS described in 16.2.3.7.

16.2.3.6 PLCP LENGTH field

The PLCP LENGTH field shall be an unsigned 16-bit integer that indicates the number of microseconds (16 to $2^{16}–1$ as defined by aMPDUMaxLength) required to transmit the MPDU. The transmitted value shall be determined from the LENGTH parameter in the TXVECTOR issued with the PHY-TXSTART.request primitive described in 7.3.5.5. The LENGTH field provided in the TXVECTOR is in octets and is converted to microseconds for inclusion in the PLCP LENGTH field. The LSB shall be transmitted first in time. This field shall be protected by the CRC-16 FCS described in 16.2.3.7.

16.2.3.7 PLCP CRC field

The IEEE 802.11 SIGNAL, IEEE 802.11 SERVICE, and LENGTH fields shall be protected with a CRC-16 FCS. The CRC-16 FCS shall be the ones complement of the remainder generated by the modulo 2 division of the protected PLCP fields by the polynomial:

$$x^{16} + x^{12} + x^5 + 1$$

The protected bits shall be processed in transmit order. All FCS calculations shall be made prior to data scrambling.

As an example, the SIGNAL, SERVICE, and LENGTH fields for a DBPSK signal with a packet length of 192 µs (24 octets) would be given by the following:

0101 0000 0000 0000 0000 0011 0000 0000 (leftmost bit transmitted first in time)

The ones complement FCS for these protected PLCP preamble bits would be the following:

0101 1011 0101 0111 (leftmost bit transmitted first in time)

Figure 16-2 depicts this example.

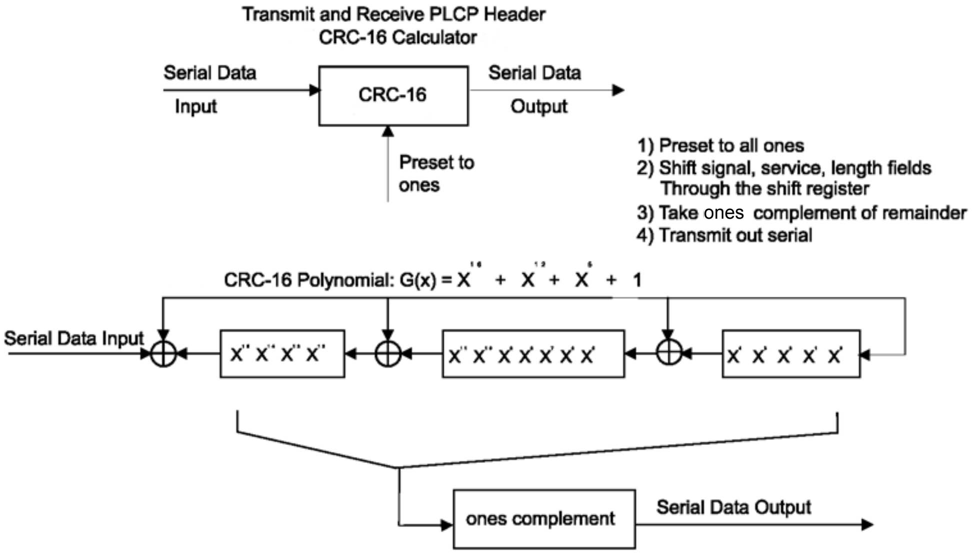

Transmit and Receive PLCP Header
CRC-16 Calculator
Serial Data
Input
CRC-16
Serial Data
Output
Preset to
ones
1) Preset to all ones
2) Shift signal, service, length fields
Through the shift register
3) Take ones complement of remainder
4) Transmit out serial
CRC-16 Polynomial: G(x) = X^16 + X^12 + X^5 + 1
Serial Data Input
ones complement
Serial Data Output

Figure 16-2—CRC-16 implementation

An illustrative example of the CRC-16 FCS using the information from Figure 16-2 follows in Figure 16-3.

Data	CRC registers MSB LSB	
	1111111111111111	; initialize preset to ones
0	1110111111011111	
1	1101111110111110	
0	1010111101011101	
1	0101111010111010	
0	1011110101110100	
0	0110101011001001	
0	1101010110010010	
0	1011101100000101	
0	0110011000101011	
0	1100110001010110	
0	1000100010001101	
0	0000000100111011	
0	0000001001110110	
0	0000010011101100	
0	0000100111011000	
0	0001001110110000	
0	0010011101100000	
0	0100111011000000	
0	1001110110000000	
0	0010101100100001	
0	0101011001000010	
0	1010110010000100	
1	0101100100001000	
1	1010001000110001	
0	0101010001000011	
0	1010100010000110	
0	0100000100101101	
0	1000001001011010	
0	0001010010010101	
0	0010100100101010	
0	0101001001010100	
0	1010010010101000	
	0101101101010111	; ones complement, result = CRC FCS parity

Figure 16-3—Example CRC calculation

16.2.4 PLCP/DSSS PHY data scrambler and descrambler

The polynomial $G(z) = z^{-7} + z^{-4} + 1$ shall be used to scramble all bits transmitted by the DSSS PHY. The feedthrough configuration of the scrambler and descrambler is self-synchronizing, which requires no prior knowledge of the transmitter initialization of the scrambler for receive processing. Figure 16-4 and Figure 16-5 show typical implementations of the data scrambler and descrambler, but other implementations are possible.

The scrambler should be initialized to any state except all ones when transmitting.

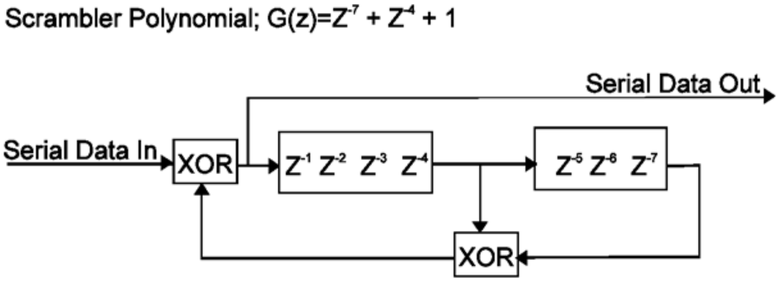


Figure 16-4—Data scrambler

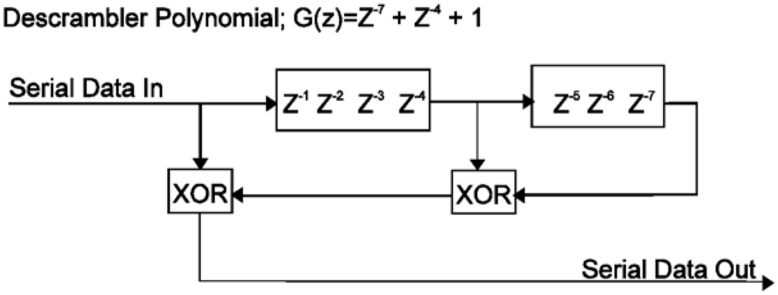


Figure 16-5—Data descrambler

16.2.5 PLCP data modulation and modulation rate change

The PLCP preamble shall be transmitted using the 1 Mb/s DBPSK modulation. The IEEE 802.11 SIGNAL field shall indicate the modulation that shall be used to transmit the MPDU. The transmitter and receiver shall initiate the modulation indicated by the IEEE 802.11 SIGNAL field starting with the first symbol (1 bit for DBPSK or 2 bits for DQPSK) of the MPDU. The MPDU transmission rate shall be set by the DATARATE parameter in the TXVECTOR issued with the PHY-TXSTART.request primitive described in 16.4.4.2.

16.2.6 Transmit PLCP

The transmit PLCP is shown in Figure 16-6.

In order to transmit data, PHY-TXSTART.request primitive shall be enabled so that the PHY entity shall be in the transmit state. Further, the PHY shall be set to operate at the appropriate channel through STA management via the PLME. Other transmit parameters such as DATARATE, TX antenna, and TX power are set via the PHY-SAP with the PHY-TXSTART.request(TXVECTOR) primitive as described in 16.4.4.3.

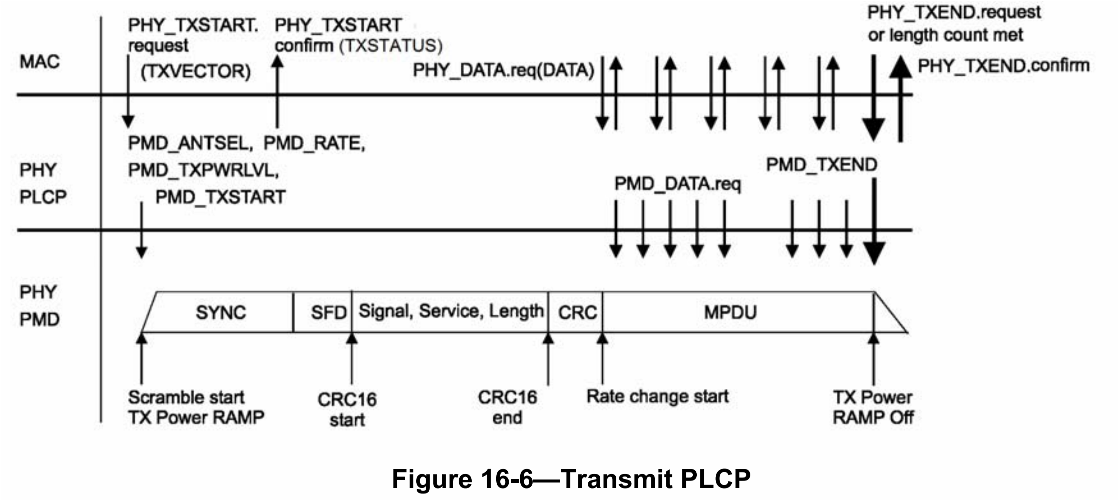


Figure 16-6—Transmit PLCP

Based on the status of CCA indicated by PHY-CCA.indication primitive, the MAC assesses that the channel is clear. A clear channel shall be indicated by PHY-CCA.indication(IDLE) primitive. If the channel is clear, transmission of the PPDU shall be initiated by issuing the PHY-TXSTART.request(TXVECTOR) primitive. The TXVECTOR elements for the PHY-TXSTART.request primitive are the PLCP header parameters SIGNAL (DATARATE), SERVICE, and LENGTH, and the PMD parameters of TX_ANTENNA, TXPWR_LEVEL, and TIME_OF_DEPARTURE_REQUESTED. The PLCP header parameter LENGTH is calculated from the TXVECTOR element by multiplying by 8 for 1 Mb/s and by 4 for 2 Mb/s.

The PLCP shall issue PMD_ANTSEL, PMD_RATE, and PMD_TXPWRLVL primitives to configure the PHY. The PLCP shall then issue a PMD_TXSTART.request primitive and the PHY entity shall immediately initiate data scrambling and transmission of the PLCP preamble based on the parameters passed in the PHY-TXSTART.request primitive. The time required for transmit power-on ramp described in 16.4.7.8 shall be included in the PLCP SYNC field. If dot11MgmtOptionTODImplemented and dot11MgmtOptionTODActivated are set to true and the TXVECTOR parameter TIME_OF_DEPARTURE_REQUESTED is true, then the PLCP shall issue a PHY_TXSTART.confirm(TXSTATUS) primitive to the MAC, forwarding the TIME_OF_DEPARTURE corresponding to the time when the first frame energy is sent by the transmitting port, and the TIME_OF_DEPARTURE_ClockRate parameters within the TXSTATUS vector. If dot11MgmtOptionTimingMsmtActivated is true, then the PLCP shall forward the value of TX_START_OF_FRAME_OFFSET in TXSTATUS vector. Once the PLCP preamble transmission is complete, data shall be exchanged between the MAC and the PHY by a series of PHY-DATA.request(DATA) primitives issued by the MAC and PHY-DATA.confirm primitives issued by the PHY. The modulation rate change, if any, shall be initiated with the first data symbol of the MPDU as described in 16.2.5. The PHY proceeds with MPDU transmission through a series of data octet transfers from the MAC. At the PMD layer, the data octets are sent in LSB-to-MSB order and presented to the PHY through PMD_DATA.request primitives. Transmission can be prematurely terminated by the MAC through the PHY-TXEND.request primitive. PHY-TXSTART shall be disabled by the issuance of the PHY-TXEND.request primitive. Normal termination occurs after the transmission of the final bit of the last MPDU octet according to the number supplied in the TXVECTOR LENGTH field. The packet transmission shall be completed and the PHY entity shall enter the receive state (i.e., PHY-TXSTART shall be disabled). It is recommended that chipping continue during power-down. Each PHY-TXEND.request primitive is acknowledged with a PHY-TXEND.confirm primitive from the PHY.

A typical state machine implementation of the transmit PLCP is provided in Figure 16-7.

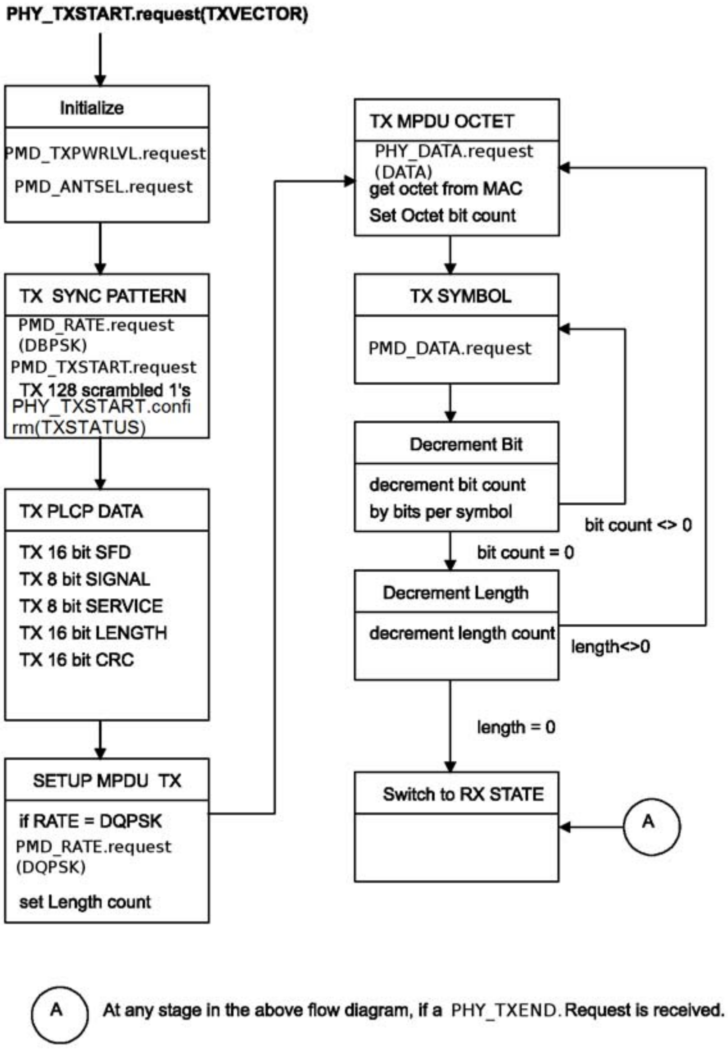


Figure 16-7—PLCP transmit state machine

16.2.7 Receive PLCP

The receive PLCP is shown in Figure 16-8.

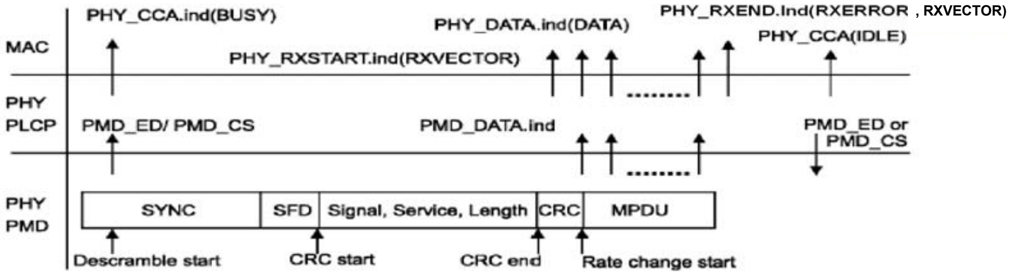


Figure 16-8—Receive PLCP

In order to receive data, PHY-TXSTART.request primitive shall be disabled so that the PHY entity is in the receive state. Further, through STA management via the PLME, the PHY is set to the appropriate channel and the CCA method is chosen. Other receive parameters such as RSSI, RCPI, signal quality (SQ), and indicated DATARATE may be accessed via the PHY-SAP.

Upon receiving the transmitted energy, according to the selected CCA mode, the PMD_ED shall be enabled (according to 16.4.8.5) as the RSSI reaches the ED_THRESHOLD and/or PMD_CS shall be enabled after code lock is established. These conditions are used to indicate activity to the MAC via PHY-CCA.indication primitive according to 16.4.8.5. A PHY-CCA.indication(BUSY) primitive shall be issued for energy detection (ED) and/or code lock prior to correct reception of the PLCP frame. The PMD primitives PMD_SQ and PMD_RSSI are issued to update the RSSI and SQ parameters reported to the MAC.

After a PHY-CCA.indication primitive is issued, the PHY entity shall begin searching for the SFD field. Once the SFD field is detected, CRC-16 processing shall be initiated and the PLCP IEEE 802.11 SIGNAL, IEEE 802.11 SERVICE and LENGTH fields are received. The CRC-16 FCS shall be processed. If the CRC-16 FCS check fails, the PHY receiver shall return to the RX IDLE state as depicted in Figure 16-9. Should the status of CCA return to the IDLE state during reception prior to completion of the full PLCP processing, the PHY receiver shall return to the RX IDLE state.

If the PLCP header reception is successful (and the SIGNAL field is completely recognizable and supported), a PHY-RXSTART.indication(RXVECTOR) primitive shall be issued. If dot11MgmtOptionTimingMsmtActivated is true, the PLCP shall do the following:

— Complete receiving the PLCP header and verify the validity of the PLCP Header.
— If the PLCP header reception is successful (and the SIGNAL field is completely recognizable and supported), a PHY-RXSTART.indication(RXVECTOR) shall be issued and RX_START_OF_FRAME_OFFSET parameter within the RXVECTOR shall be forwarded (see 16.4.4.3).

NOTE—The RX_START_OF_FRAME_OFFSET value is used as described in 6.3.57 in order to estimate when the start of the preamble for the incoming frame was detected on the medium at the receive antenna port.

The RXVECTOR associated with this primitive includes the SIGNAL field, the SERVICE field, the MPDU length in octets (calculated from the LENGTH field in microseconds), the antenna used for receive (RX_ANTENNA), RSSI, RCPI, and SQ.

The received MPDU bits are assembled into octets and presented to the MAC using a series of PHY-DATA.indication(DATA) primitive exchanges. The rate change indicated in the IEEE 802.11 SIGNAL field shall be initiated with the first symbol of the MPDU as described in 16.2.5. The PHY proceeds with MPDU reception. After the reception of the final bit of the last MPDU octet indicated by the PLCP preamble LENGTH field, the receiver shall be returned to the RX IDLE state as shown in Figure 16-9. A PHY-RXEND.indication(NoError) primitive shall be issued. A PHY-CCA.indication(IDLE) primitive shall be issued following a change in PHY carrier sense (PHYCS) and/or PHY energy detection (PHYED) according to the selected CCA method.

In the event that a change in PHYCS or PHYED would cause the status of CCA to return to the IDLE state before the complete reception of the MPDU as indicated by the PLCP LENGTH field, the error condition shall be reported to the MAC using a PHY-RXEND.indication(CarrierLost) primitive. The DSSS PHY shall ensure that the CCA indicates a busy medium for the intended duration of the transmitted packet.

If the PLCP header is successful, but the indicated rate in the SIGNAL field is not receivable, a PHY-RXSTART.indication primitive shall not be issued. The PHY shall indicate the error condition by issuing a PHY-RXEND.indication(UnsupportedRate) primitive. If the PLCP header is successful, but the SERVICE field is out of IEEE 802.11 DSSS specification, a PHY-RXSTART.indication primitive shall not be issued. The PHY shall indicate the error condition using a PHY-RXEND.indication(FormatViolation) primitive.

Also, in both cases, the DSSS PHY shall ensure that the CCA indicates a busy medium for the intended duration of the transmitted frame as indicated by the LENGTH field. The intended duration is indicated by the LENGTH field (length ×1 μs).

A typical state machine implementation of the receive PLCP is provided in Figure 16-9.

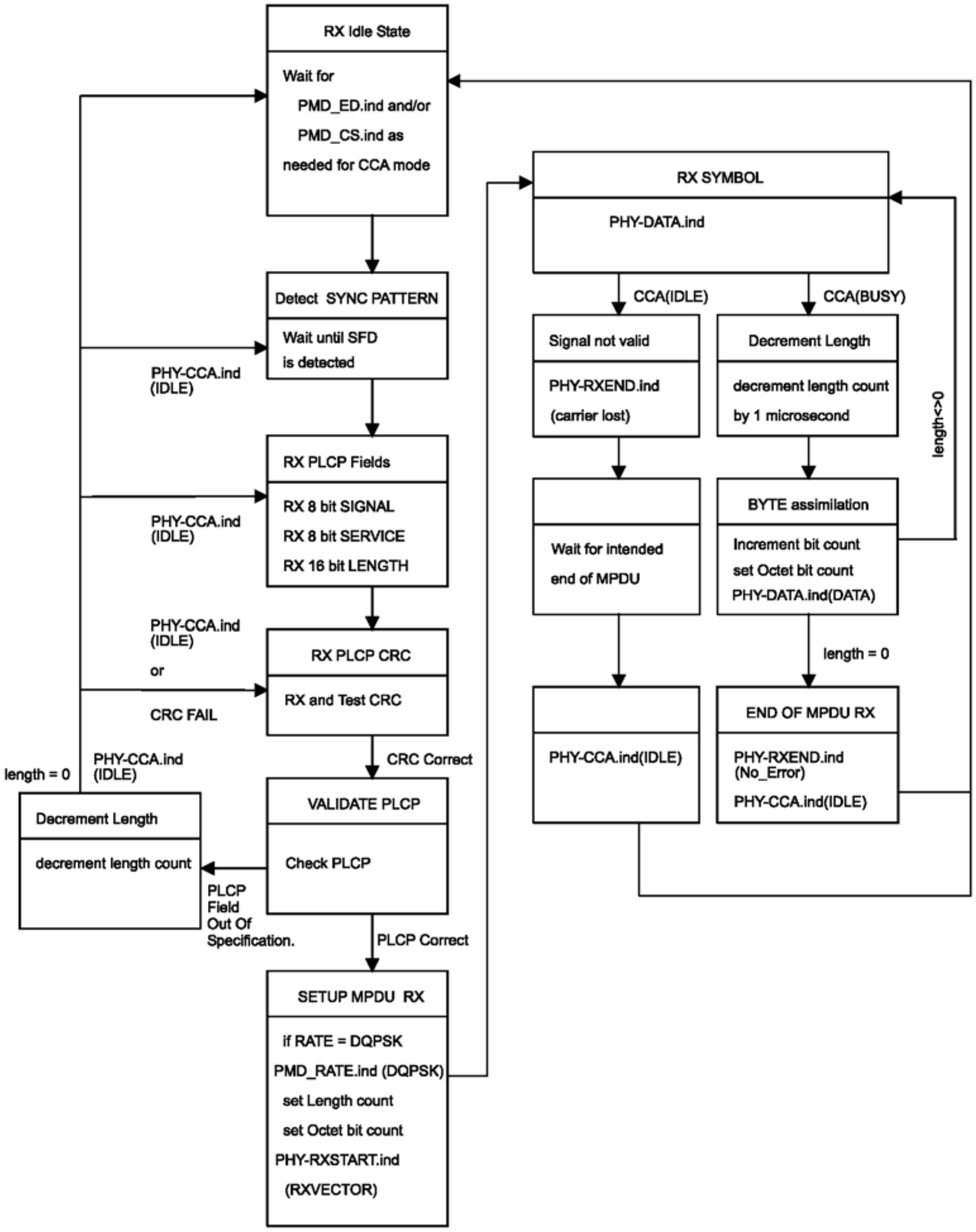


Figure 16-9—PLCP receive state machine

16.3 DSSS PLME

16.3.1 PLME_SAP sublayer management primitives

Table 16-1 lists the MIB attributes that may be accessed by the PHY entities and intralayer of higher level LMEs. These attributes are accessed via the PLME-GET, PLME-SET, and PLME-RESET primitives defined in Clause 6.

16.3.2 DSSS PHY MIB

All DSSS PHY MIB attributes are defined in Clause 7, with specific values defined in Table 16-1.

Table 16-1—MIB attribute default values/ranges

Managed object	Default value/range	Operational semantics
dot11PhyOperationComplianceGroup		
dot11PHYType	DSSS–2.4 (02)	Static
dot11RegDomainsSupported	Implementation dependent	Static
dot11CurrentRegDomain	Implementation dependent	Static
dot11PhyRateGroup		
dot11SupportedDataRatesTx	X'02', X'04'	Static
dot11SupportedDataRatesRx	X'02', X'04'	Static
dot11PhyAntennaComplianceGroup		
dot11CurrentTxAntenna	Implementation dependent	Dynamic
dot11DiversitySupportImplemented	Implementation dependent	Static
dot11CurrentRxAntenna	Implementation dependent	Dynamic
dot11PhyTxPowerComplianceGroup		
dot11NumberSupportedPowerLevelsImplemented	Implementation dependent	Static
dot11TxPowerLevel1	Implementation dependent	Static
dot11TxPowerLevel2	Implementation dependent	Static
dot11TxPowerLevel3	Implementation dependent	Static
dot11TxPowerLevel4	Implementation dependent	Static
dot11TxPowerLevel5	Implementation dependent	Static
dot11TxPowerLevel6	Implementation dependent	Static
dot11TxPowerLevel7	implementation dependent	Static
dot11TxPowerLevel8	Implementation dependent	Static
dot11CurrentTxPowerLevel	Implementation dependent	Dynamic

Table 16-1—MIB attribute default values/ranges *(continued)*

Managed object	Default value/range	Operational semantics
dot11PhyDSSSComplianceGroup		
dot11CurrentChannel	Implementation dependent	Dynamic
dot11CCAModeSupported	Implementation dependent	Static
dot11CurrentCCAMode	Implementation dependent	Dynamic
dot11EDThreshold	Implementation dependent	Dynamic
dot11AntennasListGroup		
dot11TxAntennaImplemented	Implementation dependent	Static
dot11RxAntennaImplemented	Implementation dependent	Static
dot11DiversitySelectionRxImplemented	Implementation dependent	Dynamic
NOTE—The column titled “Operational semantics” contains two types: static and dynamic. Static MIB attributes are fixed and cannot be modified for a given PHY implementation. MIB attributes defined as dynamic can be modified by some management entities.		

16.3.3 DS PHY characteristics

The static DS PHY characteristics, provided through the PLME-CHARACTERISTICS service primitive, are shown in Table 16-2. The definitions of these characteristics are in 6.5.4.

Table 16-2—DS PHY characteristics

Characteristic	Value
aSlotTime	20 µs
aSIFSTime	10 µs
aCCATime	≤ 15 µs
aPHY-RX-START-Delay	192 µs
aRxTxTurnaroundTime	≤ 5 µs
aTxPLCPDelay	Implementers may choose any value for this delay as long as the requirements of aRxTxTurnaroundTime are met.
aRxPLCPDelay	Implementers may choose any value for this delay as long as the requirements of aSIFSTime and aCCATime are met.
aRxTxSwitchTime	≤ 5 µs
aTxRampOnTime	Implementers may choose any value for this delay as long as the requirements of aRxTxTurnaroundTime are met.
aTxRampOffTime	Implementers may choose any value for this delay as long as the requirements of aSIFSTime are met.

Table 16-2—DS PHY characteristics *(continued)*

Characteristic	Value
aTxRFDelay	Implementers may choose any value for this delay as long as the requirements of aRxTxTurnaroundTime are met.
aRxRFDelay	Implementers may choose any value for this delay as long as the requirements of aSIFSTime and aCCATime are met.
aAirPropagationTime	1 µs
aMACProcessingDelay	≤ 2 µs
aPreambleLength	144 µs
aPLCPHeaderLength	48 µs
aMPDUDurationFactor	0
aMPDUMaxLength	$4 \le x \le (2^{13} - 1)$
aCWmin	31
aCWmax	1023

16.4 DSSS PMD sublayer

16.4.1 Scope and field of application

Subclause 16.4 describes the PMD services provided to the PLCP for the DSSS PHY. Also defined in 16.4 are the functional, electrical, and RF characteristics required for interoperability of implementations conforming to this standard. The relationship of this standard to the entire DSSS PHY is shown in Figure 16-10.

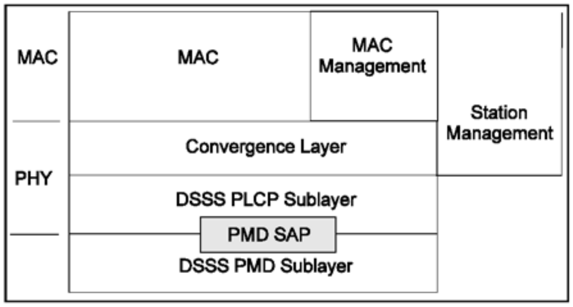


Figure 16-10—PMD layer reference model

16.4.2 Overview of service

The DSSS PMD sublayer accepts PLCP sublayer service primitives and provides the actual means by which data shall be transmitted or received from the medium. The combined function of DSSS PMD sublayer primitives and parameters for the receive function results in a data stream, timing information, and associated receive signal parameters being delivered to the PLCP sublayer. A similar functionality shall be provided for data transmission.

16.4.3 Overview of interactions

The primitives associated with the IEEE 802.11 PLCP sublayer to the DSSS PMD fall into two basic categories:

a) Service primitives that support PLCP peer-to-peer interactions, and
b) Service primitives that have local significance and that support sublayer-to-sublayer interactions.

16.4.4 Basic service and options

16.4.4.1 General

All of the service primitives described in 16.4.4 are considered mandatory unless otherwise specified.

16.4.4.2 PMD_SAP peer-to-peer service primitives

Table 16-3 indicates the primitives for peer-to-peer interactions.

Table 16-3—PMD_SAP peer-to-peer service primitives

Primitive	Request	Indicate	Confirm	Response
PHY-RXSTART		X		—
PHY-RXEND		X		—
PHY-CCA		X		—
PHY-TXSTART	X		X	—
PHY-TXEND	X		X	—
PHY-DATA	X	X	X	—

16.4.4.3 PMD_SAP peer-to-peer service primitive parameters

Several service primitives include a parameter vector. This vector shall be a list of parameters that may vary depending on PHY type. Table 16-4 indicates the parameters required by the MAC or DSSS PHY in each of the parameter vectors used for peer-to-peer interactions.

Table 16-4—DSSS PMD_SAP peer-to-peer service primitives

Parameter	Associated primitive	Value
LENGTH	PHY-RXSTART.indication (RXVECTOR), PHY-TXSTART.request (TXVECTOR)	0 to $2^{13} - 1$
DATARATE	PHY-RXSTART.indication (RXVECTOR), PHY-TXSTART.request (TXVECTOR)	1, 2 Mb/s

Table 16-4—DSSS PMD_SAP peer-to-peer service primitives *(continued)*

Parameter	Associated primitive	Value
SERVICE	PHY-RXSTART.indication (RXVECTOR), PHY-TXSTART.request (TXVECTOR)	1, 2 Mb/s
TXPWR_LEVEL	PHY-TXSTART.request (TXVECTOR)	Level1, Level2, Level3, Level4
TX_ANTENNA	PHY-TXSTART.request (TXVECTOR)	1–256
RSSI	PHY-RXSTART.indication (RXVECTOR)	0–255
SQ	PHY-RXSTART.indication (RXVECTOR)	0–255
RX_ANTENNA	PHY-RXSTART.indication (RXVECTOR)	1–256
RCPI (See NOTE)	PHY-RXSTART.indication (RXVECTOR)	0–255
TIME_OF_DEPARTURE _REQUESTED	PHY-TXSTART.request (TXVECTOR)	False, true. When true, the MAC entity requests that the PHY PLCP entity measures and reports time of departure parameters corresponding to the time when the first frame energy is sent by the transmitting port; when false, the MAC entity requests that the PHY PLCP entity neither measures nor reports time of departure parameters.
TIME_OF_DEPARTURE	PHY-TXSTART.confirm (TXSTATUS)	0 to 2^{32}–1. The locally measured time when the first frame energy is sent by the transmitting port, in units equal to 1/TIME_OF_DEPARTURE_ClockRate. This parameter is present only if TIME_OF_DEPARTURE_REQUESTED is true in the corresponding request.
TX_START_OF_FRAME _OFFSET	PHY-TXSTART.confirm (TXSTATUS)	0 to 2^{32}– 1. An estimate of the offset (in 10 ns units) from the point in time at which the start of the preamble corresponding to the frame was transmitted at the transmit antenna port to the point in time at which this primitive is issued to the MAC.
TIME_OF_DEPARTURE _ClockRate	PHY-TXSTART.confirm (TXSTATUS)	0 to 2^{16}–1. The clock rate, in units of MHz, is used to generate the TIME_OF_DEPARTURE value. This parameter is present only if TIME_OF_DEPARTURE_REQUESTED is true in the corresponding request.
RX_START_OF_FRAME _OFFSET	PHY-RXSTART.indication (RXVECTOR)	0 to 2^{32}–1. An estimate of the offset (in 10 ns units) from the point in time at which the start of the preamble corresponding to the incoming frame arrived at the receive antenna port to the point in time at which this primitive is issued to the MAC.
NOTE—RCPI is present only when dot11RadioMeasurementActivated is true.		

16.4.4.4 PMD_SAP sublayer-to-sublayer service primitives

Table 16-5 indicates the primitives for sublayer-to-sublayer interactions.

Table 16-5—PMD_SAP sublayer-to-sublayer service primitives

Primitive	Request	Indicate	Confirm	Response
PMD_TXSTART	X		—	—
PMD_TXEND	X		—	—
PMD_ANTSEL	X	X	—	—
PMD_TXPWRLVL	X		—	—
PMD_RATE	X	X	—	—
PMD_RSSI		X	—	—
PMD_SQ		X	—	—
PMD_CS		X	—	—
PMD_ED	X	X	—	—
PMD_RCPI		X	—	—

16.4.4.5 PMD_SAP service primitive parameters

Table 16-6 indicates the parameters for the PMD primitives.

Table 16-6—List of parameters for the PMD primitives

Parameter	Associated primitive	Value
DATA	PHY-DATA.request PHY-DATA.indication	Octet value: X'00'–X'FF'
TXVECTOR	PHY-DATA.request	A set of parameters
RXVECTOR	PHY-DATA.indication	A set of parameters
TXD_UNIT	PMD_DATA.request	1, 0: DBPSK dibit combinations 00,01,11,10: DQPSK
RXD_UNIT	PMD_DATA.indication	1, 0: DBPSK dibit combinations 00,01,11,10: DQPSK
RF_STATE	PMD_TXE.request	Receive, Transmit
ANT_STATE	PMD_ANTSEL.indication PMD_ANTSEL.request	1 to 256
TXPWR_LEVEL	PHY-TXSTART	0, 1, 2, 3 (max of 4 levels)
RATE	PMD_RATE.indication PMD_RATE.request	X'0A' for 1 Mb/s DBPSK X'14' for 2 Mb/s DQPSK
RSSI	PMD_RSSI.indication	0–8 bits of RSSI
SQ	PMD_SQ.indication	0–8 bits of SQ
RCPI	PMD-RCPI.indication	0–255

16.4.5 PMD_SAP detailed service specification

16.4.5.1 Introduction

The services provided by each PMD primitive are described in 16.4.5.2 to 16.4.5.16.

16.4.5.2 PMD_DATA.request

16.4.5.2.1 Function

This primitive defines the transfer of data from the PLCP sublayer to the PMD entity.

16.4.5.2.2 Semantics of the service primitive

The primitive shall provide the following parameter:

```
PMD_DATA.request(
                                  TXD_UNIT
                                  )
```

The TXD_UNIT parameter takes on the value of either 1 or 0 for DBPSK modulation or the dibit combination 00, 01, 11, or 10 for DQPSK modulation. This parameter represents a single block of data, which, in turn, shall be used by the PHY to be differentially encoded into a DBPSK or DQPSK transmitted symbol. The symbol itself shall be spread by the PN code prior to transmission.

16.4.5.2.3 When generated

This primitive shall be generated by the PLCP sublayer to request transmission of a symbol. The data clock for this primitive shall be supplied by the PMD layer based on the PN code repetition.

16.4.5.2.4 Effect of receipt

The PMD performs the differential encoding, PN code modulation, and transmission of the data.

16.4.5.3 PMD_DATA.indication

16.4.5.3.1 Function

This primitive defines the transfer of data from the PMD entity to the PLCP sublayer.

16.4.5.3.2 Semantics of the service primitive

The primitive shall provide the following parameter:

```
PMD_DATA.indication(
                                  RXD_UNIT
                                  )
```

The RXD_UNIT parameter takes on the value of 1 or 0 for DBPSK modulation or as the dibit 00, 01, 11, or 10 for DQPSK modulation. This parameter represents a single symbol that has been demodulated by the PMD entity.

16.4.5.3.3 When generated

This primitive, which is generated by the PMD entity, forwards received data to the PLCP sublayer. The data clock for this primitive shall be supplied by the PMD layer based on the PN code repetition.

16.4.5.3.4 Effect of receipt

The PLCP sublayer either interprets the bit or bits that are recovered as part of the PLCP or passes the data to the MAC sublayer as part of the MPDU.

16.4.5.4 PMD_TXSTART.request

16.4.5.4.1 Function

This primitive, which is generated by the PHY PLCP sublayer, initiates PPDU transmission by the PMD layer.

16.4.5.4.2 Semantics of the service primitive

The semantics of this primitive are as follows:
PMD_TXSTART.request

16.4.5.4.3 When generated

This primitive shall be generated by the PLCP sublayer to initiate the PMD layer transmission of the PPDU. The PHY-DATA.request primitive shall be provided to the PLCP sublayer prior to issuing the PMD_TXSTART command.

16.4.5.4.4 Effect of receipt

PMD_TXSTART initiates transmission of a PPDU by the PMD sublayer.

16.4.5.5 PMD_TXEND.request

16.4.5.5.1 Function

This primitive, which is generated by the PHY PLCP sublayer, ends PPDU transmission by the PMD layer.

16.4.5.5.2 Semantics of the service primitive

The semantics of the primitive are as follows:
PMD_TXEND.request

16.4.5.5.3 When generated

This primitive shall be generated by the PLCP sublayer to terminate the PMD layer transmission of the PPDU.

16.4.5.5.4 Effect of receipt

PMD_TXEND terminates transmission of a PPDU by the PMD sublayer.

16.4.5.6 PMD_ANTSEL.request

16.4.5.6.1 Function

This primitive, which is generated by the PHY PLCP sublayer, selects the antenna used by the PHY for transmission or reception (when diversity is disabled).

16.4.5.6.2 Semantics of the service primitive

The primitive shall provide the following parameter:

```
PMD_ANTSEL.request(
                              ANT_STATE
                              )
```

ANT_STATE selects which of the available antennas should be used for transmit. The number of available antennas shall be determined from the MIB table parameters aSuprtRxAntennas and aSuprtTxAntennas.

16.4.5.6.3 When generated

This primitive shall be generated by the PLCP sublayer to select a specific antenna for transmission or reception (when diversity is disabled).

16.4.5.6.4 Effect of receipt

PMD_ANTSEL immediately selects the antenna specified by ANT_STATE.

16.4.5.7 PMD_ANTSEL.indication

16.4.5.7.1 Function

This primitive, which is generated by the PHY PLCP sublayer, reports the antenna used by the PHY for reception of the most recent packet.

16.4.5.7.2 Semantics of the service primitive

The primitive shall provide the following parameter:

```
PMD_ANTSEL.indication(
                              ANT_STATE
                              )
```

ANT_STATE reports which of the available antennas was used for reception of the most recent packet.

16.4.5.7.3 When generated

This primitive shall be generated by the PLCP sublayer to report the antenna used for the most recent packet reception.

16.4.5.7.4 Effect of receipt

PMD_ANTSEL immediately reports the antenna specified by ANT_STATE.

16.4.5.8 PMD_TXPWRLVL.request

16.4.5.8.1 Function

This primitive, which is generated by the PHY PLCP sublayer, selects the power level used by the PHY for transmission.

16.4.5.8.2 Semantics of the service primitive

The primitive shall provide the following parameter:

```
PMD_TXPWRLVL.request(
                              TXPWR_LEVEL
                              )
```

TXPWR_LEVEL selects which of the optional transmit power levels should be used for the current packet transmission. The number of available power levels shall be determined by the MIB parameter dot11NumberSupportedPowerLevelsImplemented. See 16.4.7.4 for further information on the optional DSSS PHY power-level-control capabilities.

16.4.5.8.3 When generated

This primitive shall be generated by the PLCP sublayer to select a specific transmit power. This primitive shall be applied prior to setting PMD_TXSTART to the transmit state.

16.4.5.8.4 Effect of receipt

PMD_TXPWRLVL immediately sets the transmit power level given by TXPWR_LEVEL.

16.4.5.9 PMD_RATE.request

16.4.5.9.1 Function

This primitive, which is generated by the PHY PLCP sublayer, selects the modulation rate that shall be used by the DSSS PHY for transmission.

16.4.5.9.2 Semantics of the service primitive

The primitive shall provide the following parameter:

```
PMD_RATE.request(
                              RATE
                              )
```

The RATE parameter selects which of the DSSS PHY data rates shall be used for MPDU transmission. Subclause 16.4.6.5 provides further information on the DSSS PHY modulation rates. The DSSS PHY rate change capability is fully described in 16.2.

16.4.5.9.3 When generated

This primitive shall be generated by the PLCP sublayer to change or set the current DSSS PHY modulation rate used for the MPDU portion of a PPDU.

16.4.5.9.4 Effect of receipt

The receipt of PMD_RATE selects the rate that shall be used for all subsequent MPDU transmissions. This rate shall be used for transmission only. The DSSS PHY shall still be capable of receiving all the required DSSS PHY modulation rates.

16.4.5.10 PMD_RATE.indication

16.4.5.10.1 Function

This primitive, which is generated by the PMD sublayer, indicates which modulation rate was used to receive the MPDU portion of the PPDU. The modulation shall be indicated in the PSF.

16.4.5.10.2 Semantics of the service primitive

The primitive shall provide the following parameter:

```
PMD_RATE.indication(
                                    RATE
                                    )
```

In receive mode, the RATE parameter informs the PLCP layer which of the DSSS PHY data rates was used to process the MPDU portion of the PPDU. Subclause 16.4.6.5 provides further information on the DSSS PHY modulation rates. The DSSS PHY rate change capability is fully described in 16.2.

16.4.5.10.3 When generated

This primitive shall be generated by the PMD sublayer when the PSF has been properly detected.

16.4.5.10.4 Effect of receipt

This parameter shall be provided to the PLCP layer for information only.

16.4.5.11 PMD_RSSI.indication

16.4.5.11.1 Function

This optional primitive, which is generated by the PMD sublayer, provides to the PLCP and MAC entity the receive signal strength.

16.4.5.11.2 Semantics of the service primitive

The primitive shall provide the following parameter:

```
PMD_RSSI.indication(
                                    RSSI
                                    )
```

The RSSI shall be a measure of the RF energy received by the DSSS PHY. RSSI indications of up to 8 bits (256 levels) are supported.

16.4.5.11.3 When generated

This primitive shall be generated by the PMD when the DSSS PHY is in the receive state. It shall be continuously available to the PLCP, which, in turn, provides the parameter to the MAC entity.

16.4.5.11.4 Effect of receipt

This parameter shall be provided to the PLCP layer for information only. The RSSI may be used in conjunction with SQ as part of a CCA scheme.

16.4.5.12 PMD_SQ.indication

16.4.5.12.1 Function

This optional primitive, which is generated by the PMD sublayer, provides to the PLCP and MAC entity the SQ of the DSSS PHY PN code correlation. The SQ shall be sampled when the DSSS PHY achieves code lock and shall be held until the next code lock acquisition.

16.4.5.12.2 Semantics of the service primitive

The primitive shall provide the following parameter:

```
PMD_SQ.indication(
                                    SQ
                                    )
```

The SQ shall be a measure of the PN code correlation quality received by the DSSS PHY. SQ indications of up to 8 bits (256 levels) are supported.

16.4.5.12.3 When generated

This primitive shall be generated by the PMD when the DSSS PHY is in the receive state and code lock is achieved. It shall be continuously available to the PLCP, which, in turn, provides the parameter to the MAC entity.

16.4.5.12.4 Effect of receipt

This parameter shall be provided to the PLCP layer for information only. The SQ may be used in conjunction with RSSI as part of a CCA scheme.

16.4.5.13 PMD_CS.indication

16.4.5.13.1 Function

This primitive, which is generated by the PMD, shall indicate to the PLCP layer that the receiver has acquired (locked) the PN code and data are being demodulated.

16.4.5.13.2 Semantics of the service primitive

The PMD_CS (carrier sense) primitive in conjunction with PMD_ED provides CCA status through the PLCP layer PHY-CCA primitive. PMD_CS indicates a binary status of ENABLED or DISABLED. PMD_CS shall be ENABLED when the correlator SQ indicated in PMD_SQ is greater than the CS_THRESHOLD parameter. PMD_CS shall be DISABLED when the PMD_SQ falls below the correlation threshold.

16.4.5.13.3 When generated

This primitive shall be generated by the PHY when the DSSS PHY is receiving a PPDU and the PN code has been acquired.

16.4.5.13.4 Effect of receipt

This indicator shall be provided to the PLCP for forwarding to the MAC entity for information purposes through the PHY-CCA indicator. This parameter shall indicate that the RF medium is busy and occupied by a DSSS PHY signal. The DSSS PHY should not be placed into the transmit state when PMD_CS is ENABLED.

16.4.5.14 PMD_ED.indication

16.4.5.14.1 Function

This optional primitive, which is generated by the PMD, shall indicate to the PLCP layer that the receiver has detected RF energy indicated by the PMD_RSSI primitive that is above a predefined threshold.

16.4.5.14.2 Semantics of the service primitive

The PMD_ED (energy detect) primitive, along with the PMD_SQ, provides CCA status at the PLCP layer through the PHY-CCA primitive. PMD_ED indicates a binary status of ENABLED or DISABLED. PMD_ED shall be ENABLED when the RSSI indicated in PMD_RSSI is greater than the ED_THRESHOLD parameter. PMD_ED shall be DISABLED when the PMD_RSSI falls below the energy detect threshold.

16.4.5.14.3 When generated

This primitive shall be generated by the PHY when the PHY is receiving RF energy from any source that exceeds the ED_THRESHOLD parameter.

16.4.5.14.4 Effect of receipt

This indicator shall be provided to the PLCP for forwarding to the MAC entity for information purposes through the PMD_ED indicator. This parameter shall indicate that the RF medium may be busy with an RF energy source that is not DSSS PHY compliant. If a DSSS PHY source is being received, the PMD_CS function shall be enabled shortly after the PMD_ED function is enabled.

16.4.5.15 PMD_ED.request

16.4.5.15.1 Function

This optional primitive, which is generated by the PHY PLCP, sets the energy detect ED_THRESHOLD value.

16.4.5.15.2 Semantics of the service primitive

The primitive shall provide the following parameter:

```
PMD_ED.request(
                                    ED_THRESHOLD
                                    )
```

ED_THRESHOLD is the value that the RSSI indicated shall exceed for PMD_ED to be enabled.

16.4.5.15.3 When generated

This primitive shall be generated by the PLCP sublayer to change or set the current DSSS PHY energy detect threshold.

16.4.5.15.4 Effect of receipt

The receipt of PMD_ED immediately changes the ED threshold as set by the ED_THRESHOLD parameter.

16.4.5.16 PHY-CCA.indication

16.4.5.16.1 Function

This primitive, which is generated by the PMD, indicates to the PLCP layer that the receiver has detected RF energy that adheres to the CCA algorithm.

16.4.5.16.2 Semantics of the service primitive

The PHY-CCA primitive provides CCA status at the PLCP layer to the MAC.

16.4.5.16.3 When generated

This primitive shall be generated by the PHY when the PHY is receiving RF energy from any source that exceeds the ED_THRESHOLD parameter (PMD_ED is active), and optionally is a valid correlated DSSS PHY signal whereby PMD_CS would also be active.

16.4.5.16.4 Effect of receipt

This indicator shall be provided to the PLCP for forwarding to the MAC entity for information purposes through the PHY-CCA indicator. This parameter indicates that the RF medium may be busy with an RF energy source that may or may not be DSSS PHY compliant. If a DSSS PHY source is being received, the PMD_CS function shall be enabled shortly after the PMD_ED function is enabled.

16.4.5.17 PMD_RCPI.indication

16.4.5.17.1 Function

This optional primitive, generated by the PMD sublayer, provides the received channel power indicator (RCPI) to the PLCP and MAC.

16.4.5.17.2 Semantics of the service primitive

The primitive shall provide the following parameter:

```
  PMD_RCPI.indication(
                                         RCPI
                                         )
```

The RCPI shall be a measure of the channel power received by the DSSS PHY. RCPI indications are supported as defined in 16.4.8.6.

16.4.5.17.3 When generated

This primitive shall be generated by the PMD when the DSSS PHY is in the receive state. It shall be continuously available to the PLCP, which in turn provides the parameter to the MAC entity.

16.4.5.17.4 Effect of receipt

This parameter shall be provided to the PLCP layer for information only. The RCPI may be used in conjunction with RSNI to measure input signal quality.

16.4.6 PMD operating specifications, general

16.4.6.1 General

Subclause 16.4.6 provides general specifications for the DSSS PMD sublayer. These specifications apply to both the Receive and the Transmit functions and general operation of a DSSS PHY.

16.4.6.2 Operating frequency range

The DSSS PHY shall operate in the frequency range of 2.4 GHz to 2.4835 GHz as allocated by regulatory bodies in the China, United States and Europe or in the 2.471 GHz to 2.497 GHz frequency band as allocated by regulatory authority in Japan.

16.4.6.3 Channel Numbering of operating channels

When dot11OperatingClassesRequired is true, channel numbering shall be as specified in 18.3.8.4.2 and channelization shall be as specified in 18.3.8.4.3. When dot11OperatingClassesDefined is false or not defined, the channel center frequencies and CHNL_ID numbers shall be as shown in Table 16-7. See the applicable regulations for the countries in which the implementation operates.

Table 16-7—DSSS PHY frequency channel plan

CHNL_ID	Frequency (MHz)
1	2412
2	2417
3	2422
4	2427
5	2432
6	2437
7	2442
8	2447
9	2452
10	2457
11	2462
12	2467
13	2472
14	2484

In a multiple cell network topology, overlapping and/or adjacent cells using different channels can operate simultaneously without interference if the distance between the center frequencies is at least 30 MHz. Channel 14 shall be designated specifically for operation in Japan.

16.4.6.4 Spreading sequence

The following 11-chip Barker sequence shall be used as the PN code sequence:
+1, –1, +1, +1, –1, +1, +1, +1, –1, –1, –1

The leftmost chip shall be output first in time. The first chip shall be aligned at the start of a transmitted symbol. The symbol duration shall be exactly 11 chips long.

16.4.6.5 Modulation and channel data rates

Two modulation formats and data rates are specified for the DSSS PHY: a *basic access rate* and an *enhanced access rate*. The basic access rate shall be based on 1 Mb/s DBPSK modulation. The DBPSK encoder is specified in Table 16-8. The enhanced access rate shall be based on 2 Mb/s DQPSK. The DQPSK encoder is specified in Table 16-9. (In the tables, +jω shall be defined as counterclockwise rotation.)

Table 16-8—1 Mb/s DBPSK encoding table

Bit input	Phase change (+jω)
0	0
1	π

Table 16-9—2 Mb/s DQPSK encoding table

Dibit pattern (d0,d1) d0 is first in time	Phase change (+jω)
00	0
01	π/2
11	π
10	3π/2 (–π/2)

16.4.6.6 Transmit and receive in-band and out-of-band spurious emissions

The DSSS PHY shall conform with in-band and out-of-band spurious emissions as set by the appropriate regulatory bodies.

16.4.6.7 TX-to-RX turnaround time

The TX-to-RX turnaround time shall be less than 10 µs, including the power-down ramp specified in 16.4.7.8.

The TX-to-RX turnaround time shall be measured at the air interface from the trailing edge of the last transmitted symbol to valid CCA detection of the incoming signal. The CCA should occur within 25 µs (10 µs for turnaround time plus 15 µs for energy detect) or by the next slot boundary occurring after 25 µs has elapsed (refer to 16.4.8.5). A receiver input signal 3 dB above the ED threshold described in 16.4.8.5 shall be present at the receiver.

16.4.6.8 RX-to-TX turnaround time

The RX-to-TX turnaround time shall be measured at the MAC/PHY interface, using the PHY-TXSTART.request primitive and shall be ≤ 5 µs. This includes the transmit power-on ramp described in 16.4.7.8.

16.4.6.9 Slot time

The slot time for the DSSS PHY shall be the sum of the RX-to-TX turnaround time (5 µs) and the energy detect time (15 µs specified in 16.4.8.5). The propagation delay shall be regarded as being included in the energy detect time.

16.4.6.10 Transmit and receive antenna port impedance

The impedance of the transmit and receive antenna port(s) shall be 50 Ω if the port is exposed.

16.4.7 PMD transmit specifications

16.4.7.1 Introduction

The transmit functions and parameters associated with the PMD sublayer are described in 16.4.7.2 to 16.4.7.10.

16.4.7.2 Transmit power levels

The maximum allowable output power is measured in accordance with practices specified by the appropriate regulatory bodies.

16.4.7.3 Minimum transmitted power level

The minimum transmitted power shall be no less than 1 mW.

16.4.7.4 Transmit power level control

Power control shall be provided for transmitted power greater than 100 mW. A maximum of four power levels may be provided. At a minimum, a radio capable of transmission greater than 100 mW shall be capable of switching power back to 100 mW or less.

16.4.7.5 Transmit spectrum mask

The transmitted spectral products shall be less than –30 dBr (decibel relative to the SINx/x peak) for f_c – 22 MHz < f < f_c –11 MHz, f_c +11 MHz < f < f_c + 22 MHz, –50 dBr for f < f_c –22 MHz, and f > f_c + 22 MHz, where f_c is the channel center frequency. The transmit spectral mask is shown in Figure 16-11. The measurements shall be made using 100 kHz resolution bandwidth and a 30 kHz video bandwidth.

Channel 14 is unique. The Japanese standard ARIB RCR-STD 33 (5.0) [B7] states that B90/2pi normalized to the 'transmission speed of modulation signal' shall be > 10. Therefore, for channel 14, B90/2pi > 13.75 MHz for CCK spreading and >10.0 MHz for Barker spreading.

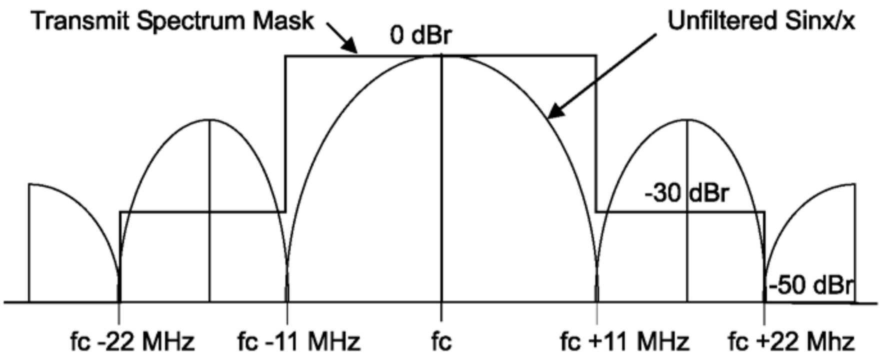


Figure 16-11—Transmit spectrum mask

16.4.7.6 Transmit center frequency tolerance

The transmitted center frequency tolerance shall be ±25 ppm maximum.

16.4.7.7 Chip clock frequency tolerance

The PN code chip clock frequency tolerance shall be better than ±25 ppm maximum.

16.4.7.8 Transmit power-on and power-down ramp

The transmit power-on ramp for 10% to 90% of maximum power shall be no greater than 2 µs. The transmit power-on ramp is shown in Figure 16-12.

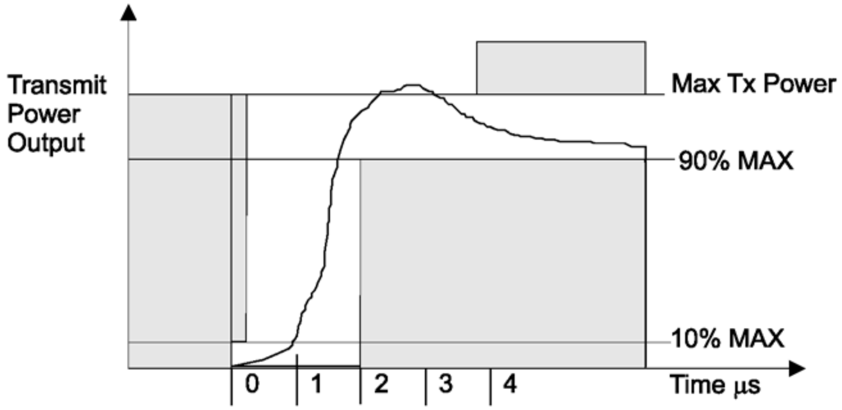


Figure 16-12—Transmit power-on ramp

The transmit power-down ramp for 90% to 10% maximum power shall be no greater than 2 µs. The transmit power-down ramp is shown in Figure 16-13.

The transmit power ramps shall be constructed such that the DSSS PHY emissions conform with the spurious frequency product specification defined in 16.4.6.6.

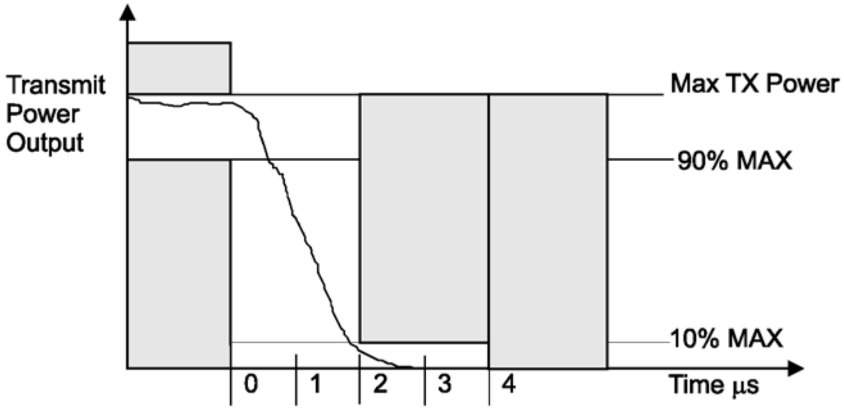


Figure 16-13—Transmit power-down ramp

16.4.7.9 RF carrier suppression

The RF carrier suppression, measured at the channel center frequency, shall be at least 15 dB below the peak SIN(x)/x power spectrum. The RF carrier suppression shall be measured while transmitting a repetitive 01 data sequence with the scrambler disabled using DQPSK modulation. A 100 kHz resolution bandwidth shall be used to perform this measurement.

16.4.7.10 Transmit modulation accuracy

The transmit modulation accuracy requirement for the DSSS PHY shall be based on the difference between the actual transmitted waveform and the ideal signal waveform. Modulation accuracy shall be determined by measuring the peak vector error magnitude measured during each chip period. Worst-case vector error magnitude shall not exceed 0.35 for the normalized sampled chip data. The ideal complex I and Q constellation points associated with DQPSK modulation (0.707, 0.707), (0.707, –0.707), (–0.707, 0.707), (–0.707, –0.707) shall be used as the reference. These measurements shall be from baseband I and Q sampled data after recovery through a reference receiver system.

Figure 16-14 illustrates the ideal DQPSK constellation points and range of worst-case error specified for modulation accuracy.

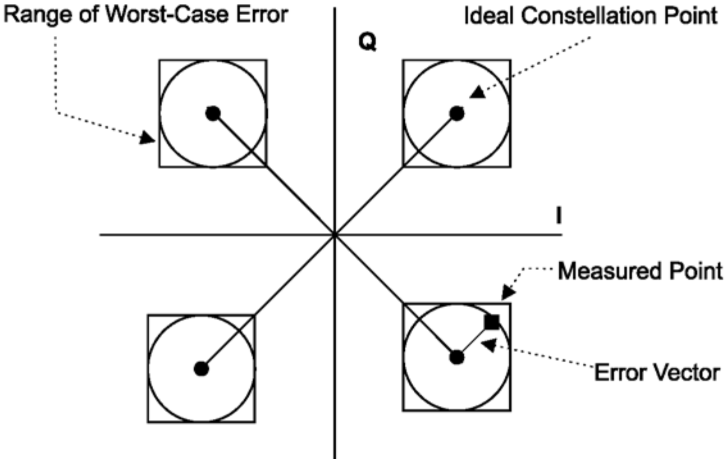


Figure 16-14—Modulation accuracy measurement example

Error vector measurement requires a reference receiver capable of carrier lock. All measurements shall be made under carrier lock conditions. The distortion induced in the constellation by the reference receiver shall be calibrated and measured. The test data error vectors described below shall be corrected to compensate for the reference receiver distortion.

The IEEE 802.11 vendor compatible radio shall provide an exposed TX chip clock, which shall be used to sample the I and Q outputs of the reference receiver.

The measurement shall be made under the conditions of continuous DQPSK transmission using scrambled all ones.

The eye pattern of the I channel shall be used to determine the *I* and *Q* sampling point. The chip clock provided by the vendor radio shall be time delayed such that the samples fall at a 1/2 chip period offset from the mean of the zero crossing positions of the eye (see Figure 16-15). This is the ideal center of the eye and may not be the point of maximum eye opening.

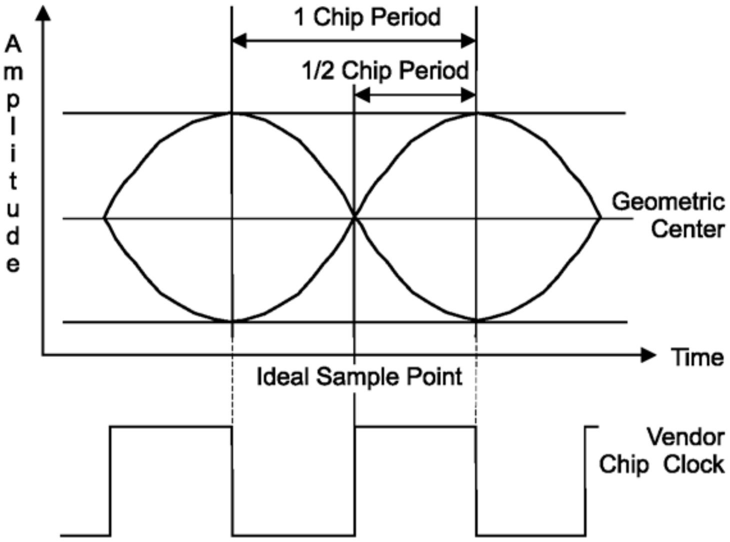


Figure 16-15—Chip clock alignment with baseband eye pattern

Using the aligned chip clock, 1000 samples of the *I* and *Q* baseband outputs from the reference receiver are captured. The vector error magnitudes shall be calculated as follows:

Calculate the dc offsets for *I* and *Q* samples.

$$I_{\text{mean}} = \sum_{n=1}^{1000} I(n)/1000$$

$$Q_{\text{mean}} = \sum_{n=1}^{1000} Q(n)/1000$$

Calculate the dc corrected *I* and *Q* samples for all *n* =1000 sample pairs.

$$I_{\text{dc}}(n) = I(n) - I_{\text{mean}}$$

$Q_{dc}(n) = Q(n) - Q_{mean}$

Calculate the average magnitude of *I* and *Q* samples.

$$I_{mag} = \sum_{n=1}^{1000} |I_{dc}(n)| / 1000$$

$$Q_{mag} = \sum_{n=1}^{1000} |Q_{dc}(n)| / 1000$$

Calculate the normalized error vector magnitude for the $I_{dc}(n)/Q_{dc}(n)$ pairs.

$$V_{err}(n) = [\{|I_{dc}(n)|/I_{mag} - 1\}^2 + \{|Q_{dc}(n)|/Q_{mag} - 1\}^2]^{\frac{1}{2}} - V_{correction}$$

with $V_{correction}$ = error induced by the reference receiver system.

A vendor DSSS PHY implementation shall be compliant if for all n =1000 samples the following condition is met:

$V_{err}(n) < 0.35$

16.4.7.11 Time of Departure accuracy

The Time of Departure accuracy test evaluates TIME_OF_DEPARTURE against aTxPmdTxStartRMS and aTxPmdTxStartRMS against TIME_OF_DEPARTURE_ACCURACY_TEST_THRESH as defined Annex T with the following test parameters:

— MULTICHANNEL_SAMPLING_RATE is $22 \times 10^6\left(1 + \left\lceil \frac{f_H - f_L}{22\text{ MHz}} \right\rceil\right)$ sample/s

where

f_H is the nominal center frequency in Hz of the highest channel in the channel set

f_L is the nominal center frequency in Hz of the lowest channel in the channel set, the channel set is the set of channels upon which frames providing measurements are transmitted, the channel set comprises channels uniformly spaced across $f_H - f_L \geq 50$ MHz

$\lceil x \rceil$ equals the smallest integer equal to or larger than x

— FIRST_TRANSITION_FIELD is the SYNC field.

— SECOND_TRANSITION_FIELD is the SFD field.

— TRAINING_FIELD is the concatenation of the SYNC and SFD fields, using a chip pulse that should approximate a rectangular pulse of duration 1/ 11 MHz convolved with a brick-wall low pass filter of bandwidth 11 MHz.

— TIME_OF_DEPARTURE_ACCURACY_TEST_THRESH is 80 ns.

NOTE—The indicated chip pulse applies to the time of departure accuracy test equipment, and not the transmitter or receiver.

16.4.8 PMD receiver specifications

16.4.8.1 Introduction

The receive functions and parameters associated with the PMD sublayer are described in 16.4.8.2 to 16.4.8.5.

16.4.8.2 Receiver minimum input level sensitivity

The FER shall be less than 8×10^{-2} at an MPDU length of 1024 octets for an input level of –80 dBm measured at the antenna connector. This FER shall be specified for 2 Mb/s DQPSK modulation. The test for the minimum input level sensitivity shall be conducted with the ED threshold set ≤ –80 dBm.

16.4.8.3 Receiver maximum input level

The receiver shall provide a maximum FER of 8×10^{-2} at an MPDU length of 1024 octets for a maximum input level of –4 dBm measured at the antenna. This FER shall be specified for 2 Mb/s DQPSK modulation.

16.4.8.4 Receiver adjacent channel rejection

Adjacent channel rejection is defined between any two channels with ≥ 30 MHz separation in each channel group defined in 16.4.6.3.

The adjacent channel rejection shall be ≥ 35 dB with an FER of 8×10^{-2} using 2 Mb/s DQPSK modulation described in 16.4.6.5 and an MPDU length of 1024 octets.

The adjacent channel rejection shall be measured using the following method:

Input a 2 Mb/s DQPSK modulated signal at a level 6 dB greater than specified in 16.4.8.2. In an adjacent channel (≥ 30 MHz separation as defined by the channel numbering), input a signal modulated in a similar fashion that adheres to the transmit mask specified in 16.4.7.5 to a level 41 dB above the level specified in 16.4.8.2. The adjacent channel signal shall be derived from a separate signal source. It shall not be a frequency shifted version of the reference channel. Under these conditions, the FER shall be no worse than 8×10^{-2}.

16.4.8.5 CCA

The DSSS PHY shall provide the capability to perform CCA according to at least one of the following three methods:

- *CCA Mode 1:* Energy above threshold. CCA shall report a busy medium upon detection of any energy above the ED threshold.
- *CCA Mode 2:* CS only. CCA shall report a busy medium only upon detection of a DSSS signal. This signal may be above or below the ED threshold.
- *CCA Mode 3:* CS with energy above threshold. CCA shall report a busy medium upon detection of a DSSS signal with energy above the ED threshold.

The ED status shall be given by the PMD primitive, PMD_ED. The CS status shall be given by PMD_CS. The status of PMD_ED and PMD_CS is used in the PLCP to indicate activity to the MAC through the PHY PHY-CCA.indication primitive.

A busy channel shall be indicated by a PHY-CCA.indication primitive of class BUSY.

A clear channel shall be indicated by a PHY-CCA.indication primitive of class IDLE.

The dot11CCAModeSupported shall indicate the appropriate operation modes. The PHY shall be configured through dot11CurrentCCAMode.

The CCA shall be true if there is no energy detect or CS. The CCA parameters are subject to the following criteria:

a) The ED threshold shall be ≤ –80 dBm for TX power > 100 mW, –76 dBm for 50 mW < TX power ≤ 100 mW, and –70 dBm for TX power ≤ 50 mW.
b) With a valid signal (according to the CCA mode of operation) present at the receiver antenna within 5 µs of the start of a MAC slot boundary, the CCA indicator shall report channel busy before the end of the slot time. This implies that the CCA signal is available as an exposed test point. Refer to Figure 9-14 (in 9.3.7) for a definition of slot time boundary.
c) In the event that a correct PLCP header is received, the DSSS PHY shall hold the CCA signal inactive (channel busy) for the full duration as indicated by the PLCP LENGTH field. Should a loss of CS occur in the middle of reception, the CCA shall indicate a busy medium for the intended duration of the transmitted packet.

Conformance to DSSS PHY CCA shall be demonstrated by applying a DSSS-compliant signal, above the appropriate ED threshold (item a), so that all conditions described in item b and item c are demonstrated.

16.4.8.6 Received Channel Power Indicator Measurement

The RCPI indicator is a measure of the received RF power in the selected channel for a received frame. This parameter shall be a measure by the PHY sublayer of the received RF power in the channel measured over the entire received frame or by other equivalent means that meet the specified accuracy. RCPI shall be a monotonically increasing, logarithmic function of the received power level defined in dBm. The allowed values for the RCPI parameter shall be an 8-bit value in the range from 0 to 220, with indicated values rounded to the nearest 0.5 dB as follows:

0: Power ≤ – 110 dBm

1: Power = – 109.5 dBm

2: Power = – 109.0 dBm

and so on where

$$\text{RCPI} = \text{Int}\{(\text{Power in dBm} + 110) \times 2\} \text{ for } 0 \text{ dBm} > \text{Power} > -110 \text{ dBm}$$

220: Power ≥ – 0 dBm

221–254: Reserved

255: Measurement not available

RCPI shall equal the received RF power within an accuracy of ±5 dB (95% confidence interval) within the specified dynamic range of the receiver. The received RF power shall be determined assuming a receiver noise equivalent bandwidth equal to the channel bandwidth multiplied by 1.1.

17. High Rate direct sequence spread spectrum (HR/DSSS) PHY specification

17.1 Overview

17.1.1 General

This clause specifies the High Rate extension of the PHY for the DSSS system (see Clause 16), hereinafter known as the High Rate PHY for the 2.4 GHz band designated for ISM applications.

This extension of the DSSS system builds on the data rate capabilities, as described in Clause 16, to provide 5.5 Mb/s and 11 Mb/s payload data rates in addition to the 1 Mb/s and 2 Mb/s rates. To provide the higher rates, 8-chip complementary code keying (CCK) is employed as the modulation scheme. The chipping rate is 11 MHz, which is the same as the DSSS system described in Clause 16, thus providing the same occupied channel bandwidth. The basic new capability described in this clause is called HR/DSSS. The basic High Rate PHY uses the same PLCP preamble and header as the DSSS PHY, so both PHYs can co-exist in the same BSS and can use the rate switching mechanism as provided.

In addition to providing higher speed extensions to the DSSS system, a number of optional features allow the performance of the RF LAN system to be improved as technology allows the implementation of these options to become cost effective.

An optional mode replacing the CCK modulation with HR/DSSS/PBCC is provided.

The PBCC option is obsolete. Consequently, this option may be removed in a later revision of the standard.

Another optional mode is provided that allows data throughput at the higher rates (2, 5.5, and 11 Mb/s) to be significantly increased by using a shorter PLCP preamble. This mode is called HR/DSSS/short, or HR/DSSS/PBCC/short. This short preamble mode can coexist with DSSS, HR/DSSS, or HR/DSSS/PBCC under limited circumstances, such as on different channels or with appropriate CCA mechanisms.

An optional capability for Channel Agility is also provided. This option allows an implementation to overcome some inherent difficulty with static channel assignments (a tone jammer), without burdening all implementations with the added cost of this capability. This option can also be used to implement IEEE 802.11-compliant systems that are interoperable with both FH and DS modulations. See Annex K for more details.

17.1.2 Scope

This clause specifies the PHY entity for the HR/DSSS extension and explains how this standard accommodates the High Rate PHY.

The High Rate PHY consists of the following two protocol functions:

a) A PHY convergence function, which adapts the capabilities of the PMD system to the PHY service. This function is supported by the PLCP, which defines a method for mapping the MPDUs into a framing format suitable for sending and receiving user data and management information between two or more STAs using the associated PMD system. The PHY exchanges PPDUs that contain PSDUs. The MAC uses the PHY service, so each MPDU corresponds to a PSDU that is carried in a PPDU.
b) A PMD system, whose function defines the characteristics of, and method of transmitting and receiving data through, a WM between two or more STAs, each using the High Rate PHY system.

17.1.3 High Rate PHY functions

17.1.3.1 General

The 2.4 GHz High Rate PHY architecture is depicted in the ISO/IEC basic reference model shown in Figure 17-10 (in 17.4.1). The High Rate PHY contains three functional entities: the PMD function, the PHY convergence function, and the layer management function. Each of these functions is described in detail in 17.1.3.2, 17.1.3.3, and 17.1.3.4. For the purposes of MAC and MAC management, when Channel Agility is both present and enabled (see 17.3.2 and Annex J), the High Rate PHY shall be interpreted to be both a High Rate and an FH PHY.

The High Rate PHY service shall be provided to the MAC through the PHY service primitives described in Clause 7.

17.1.3.2 PLCP sublayer

To allow the MAC to operate with minimum dependence on the PMD sublayer, a PLCP sublayer is defined. This function simplifies the PHY service interface to the MAC services.

17.1.3.3 PMD sublayer

The PMD sublayer provides a means and method of transmitting and receiving data through a WM between two or more STAs, each using the High Rate system.

17.1.3.4 PLME

The PLME performs management of the local PHY functions in conjunction with the MLME.

17.1.4 Service specification method and notation

The models represented by figures and state diagrams are intended to be illustrations of functions provided. It is important to distinguish between a model and a real implementation. The models are optimized for simplicity and clarity of presentation; the actual method of implementation is left to the discretion of the High-Rate-PHY-compliant developer.

The service of a layer or sublayer is a set of capabilities that it offers to a user in the next-higher layer (or sublayer). Abstract services are specified here by describing the service primitives and parameters that characterize each service. This definition is independent of any particular implementation.

17.2 High Rate PLCP sublayer

17.2.1 Overview

Subclause 17.2 provides a convergence procedure for the 2 Mb/s, 5.5 Mb/s, and 11 Mb/s specification, in which PSDUs are converted to and from PPDUs. During transmission, the PSDU shall be appended to a PLCP preamble and header to create the PPDU. Two different preambles and headers are defined: the mandatory supported long preamble and header, which interoperates with the current 1 Mb/s and 2 Mb/s DSSS specification, and an optional short preamble and header. At the receiver, the PLCP preamble and header are processed to aid in demodulation and delivery of the PSDU.

The optional short preamble and header is intended for applications where maximum throughput is desired and interoperability with legacy and nonshort-preamble-capable equipment is not a consideration. That is, it is expected to be used only in networks of like equipment, which can all handle the optional mode.

17.2.2 PPDU format

17.2.2.1 General

Two different preambles and headers are defined: the mandatory supported long preamble and header, which is interoperable with the current 1 Mb/s and 2 Mb/s DSSS specification, and an optional short preamble and header.

17.2.2.2 Long PPDU format

Figure 17-1 shows the format for the interoperable (long) PPDU, including the High Rate PLCP preamble, the High Rate PLCP header, and the PSDU. The PLCP preamble contains the following fields: SYNC and SFD. The PLCP header contains the following fields: signaling (SIGNAL), service (SERVICE), length (LENGTH), and CRC-16. Each of these fields is described in detail in 17.2.3. The format for the PPDU, including the long High Rate PLCP preamble, the long High Rate PLCP header, and the PSDU, does not differ from the format for 1 Mb/s and 2 Mb/s. The only exceptions are:

a) The encoding of the rate in the SIGNAL field;
b) The use of a bit in the SERVICE field to resolve an ambiguity in PSDU length in octets, when the length is expressed in whole microseconds;
c) The use of a bit in the SERVICE field to indicate if the optional PBCC mode is being used;
d) The use of a bit in the SERVICE field to indicate that the transit frequency and bit clocks are locked.

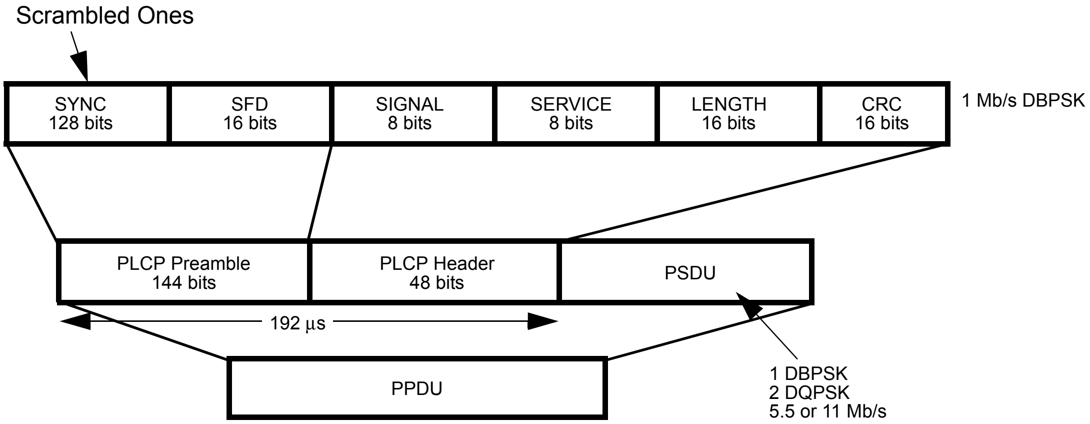


Figure 17-1—Long PPDU format

17.2.2.3 Short PPDU format

The short PLCP preamble and header (HR/DSSS/short) is defined as optional for HR/DSSS. The short preamble and header may be used to minimize overhead and, thus, maximize the network data throughput. The format of the PPDU, with HR/DSSS/short, is depicted in Figure 17-2. For Clause 19 STAs support of this preamble type is mandatory.

A transmitter using the short PLCP is only interoperable with another receiver that is also capable of receiving this short PLCP. To interoperate with a receiver that is not capable of receiving a short preamble and header, the transmitter shall use the long PLCP preamble and header. The short PLCP preamble uses the 1 Mb/s Barker code spreading with DBPSK modulation. The short PLCP header uses the 2 Mb/s Barker code spreading with DQPSK modulation, and the PSDU is transmitted at 2 Mb/s, 5.5 Mb/s, or 11 Mb/s.

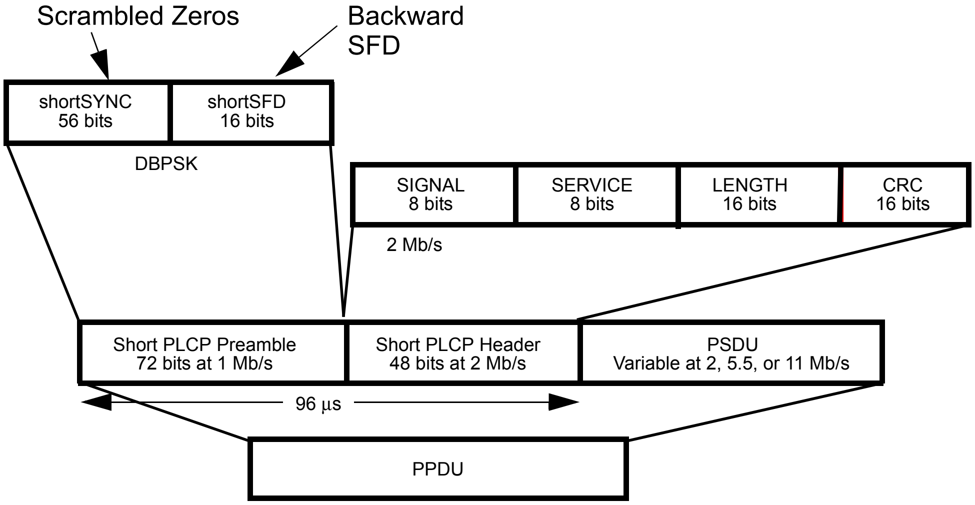


Figure 17-2—Short PPDU format

17.2.3 PPDU field definitions

17.2.3.1 General

In the PLCP field definition subclauses (17.2.3.2 to 17.2.3.15), the definitions of the long (Clause 16) PLCP fields are given first, followed by the definitions of the short PLCP. The names for the short PLCP fields are preceded by the term *short*.

17.2.3.2 Long PLCP SYNC field

The SYNC field shall consist of 128 bits of scrambled 1 bits. This field is provided so the receiver can perform the necessary synchronization operations. The initial state of the scrambler (seed) shall be [1101100], where the leftmost bit specifies the value to put in the first delay element (Z^1) in Figure 17-5 (in 17.2.4), and the rightmost bit specifies the value to put in the last delay element in the scrambler.

To support the reception of DSSS signals generated with implementations based on Clause 16, the receiver shall also be capable of synchronization on a SYNC field derived from any nonzero scrambler initial state.

17.2.3.3 Long PLCP SFD

The SFD shall be provided to indicate the start of PHY-dependent parameters within the PLCP preamble. The SFD shall be a 16-bit field, [1111 0011 1010 0000], where the rightmost bit shall be transmitted first in time.

17.2.3.4 Long PLCP SIGNAL field

The 8-bit SIGNAL field indicates to the PHY the modulation that shall be used for transmission (and reception) of the PSDU. The data rate shall be equal to the SIGNAL field value multiplied by 100 kbit/s. The High Rate PHY supports four mandatory rates given by the following 8-bit words, which represent the rate in units of 100 kbit/s, where the LSB shall be transmitted first in time:

a) X’0A’ (MSB to LSB) for 1 Mb/s;

b) X’14’ (MSB to LSB) for 2 Mb/s;

c) X’37’ (MSB to LSB) for 5.5 Mb/s;

d) X’6E’ (MSB to LSB) for 11 Mb/s.

The High Rate PHY rate change capability is described in 17.2.3.15. This field shall be protected by the CRC-16 FCS described in 17.2.3.7.

17.2.3.5 Long PLCP SERVICE field

Three bits have been defined in the SERVICE field to support the High Rate extension. The rightmost bit (bit 7) shall be used to supplement the LENGTH field described in 17.2.3.6. Bit 3 shall be used to indicate whether the modulation method is CCK <0> or PBCC <1>, as shown in Table 17-1. Bit 2 shall be used to indicate that the transmit frequency and symbol clocks are derived from the same oscillator. This locked clocks bit shall be set by the PHY based on its implementation configuration. The SERVICE field shall be transmitted b0 first in time, and shall be protected by the CRC-16 FCS described in 17.2.3.7. An IEEE 802.11-compliant device shall set the values of the bits b0, b1, b4, b5, and b6 to 0.

Table 17-1—SERVICE field definitions

b0	b1	b2	b3	b4	b5	b6	b7
Reserved	Reserved	Locked clocks bit 0 = not 1 = locked	Mod. selection bit 0 = XXK 1 = PBCC	Reserved	Reserved	Reserved	Length extension bit

17.2.3.6 Long PLCP LENGTH field

The PLCP LENGTH field shall be an unsigned 16-bit integer that indicates the number of microseconds required to transmit the PSDU. The transmitted value shall be determined from the LENGTH and DATARATE parameters in the TXVECTOR issued with the PHY-TXSTART.request primitive described in 17.4.4.3.

The LENGTH field provided in the TXVECTOR is in octets and is converted to microseconds for inclusion in the PLCP LENGTH field. The LENGTH field is calculated as follows. Because there is an ambiguity in the number of octets that is described by a length in integer microseconds for any data rate over 8 Mb/s, a length extension bit shall be placed at bit position b7 in the SERVICE field to indicate when the smaller potential number of octets is correct as follows:

a) 5.5 Mb/s CCK Length = number of octets × 8/5.5, rounded up to the next integer.

b) 11 Mb/s CCK Length = number of octets × 8/11, rounded up to the next integer; the service field (b7) bit shall indicate a 0 if the rounding took less than 8/11 or a 1 if the rounding took more than or equal to 8/11.

c) 5.5 Mb/s PBCC Length = (number of octets + 1) × 8/5.5, rounded up to the next integer.

d) 11 Mb/s PBCC Length = (number of octets + 1) × 8/11, rounded up to the next integer; the service field (b7) bit shall indicate a 0 if the rounding took less than 8/11 or a 1 if the rounding took more than or equal to 8/11.

At the receiver, the number of octets in the MPDU is calculated as follows:

a) 5.5 Mb/s CCK Number of octets = Length × 5.5/8, rounded down to the next integer.

b) 11 Mb/s CCK Number of octets = Length × 11/8, rounded down to the next integer, minus 1 if the service field (b7) bit is a 1.

c) 5.5 Mb/s PBCC Number of octets = (Length × 5.5/8) –1, rounded down to the next integer.

d) 11 Mb/s PBCC Number of octets = $(\text{Length} \times 11/8) - 1$, rounded down to the next integer, minus 1 if the service field (b7) bit is a 1.

An example for an 11 Mb/s calculation described in psuedocode form is shown below. At the transmitter, the values of the LENGTH field and length extension bit are calculated as follows:

```
    $\text{LENGTH}' = ((\text{number of octets} + P) \times 8) / R$

    $\text{LENGTH} = \text{Ceiling}(\text{LENGTH}')$

If

    $(R = 11)$ and $((\text{LENGTH} - \text{LENGTH}') \geq 8/11)$

then

    Length Extension = 1

else

    Length Extension = 0
```

where
R is the data rate (Mb/s)
P is 0 for CCK
P is 1 for PBCC
Ceiling (X) returns the smallest integer value greater than or equal to X

At the receiver, the number of octets in the MPDU is calculated as follows:

$$\text{Number of octets} = \text{Floor}(((\text{Length} \times R) / 8) - P) - \text{Length Extension}$$

where
R is the data rate (Mb/s)
P is 0 for CCK
P is 1 for PBCC
Floor (X) returns the largest integer value less than or equal to X

Table 17-2 shows an example calculation for several packet lengths of CCK at 11 Mb/s.

Table 17-2—Example of LENGTH calculations for CCK

TX octets	Octets (× 8/11)	LENGTH	Length extension bit	LENGTH (× 11/8)	Floor (X)	RX octets
1023	744	744	0	1023	1023	1023
1024	744.7273	745	0	1024.375	1024	1024
1025	745.4545	746	0	1025.75	1025	1025
1026	746.1818	747	1	1027.125	1027	1026

Table 17-3 shows an example calculation for several packet lengths of PBCC at 11 Mb/s.

Table 17-3—Example of LENGTH calculations for PBCC

TX octets	(Octets × 8/11) + 1	LENGTH	Length extension bit	(LENGTH × 11/8) – 1	Floor (X)	RX octets
1023	744.7273	745	0	1023.375	1023	1023
1024	745.4545	746	0	1024.750	1024	1024
1025	746.1818	747	1	1026.125	1026	1025
1026	746.9091	747	0	1026.125	1026	1026

This example illustrates why normal rounding or truncation of the number does produce the right result. The LENGTH field is defined in units of microseconds and corresponds to the actual length, and the number of octets is exact.

The LSB shall be transmitted first in time. This field shall be protected by the CRC-16 FCS described in 17.2.3.7.

17.2.3.7 PLCP CRC (CRC-16) field

The SIGNAL, SERVICE, and LENGTH fields shall be protected with a CRC-16 FCS. The CRC-16 FCS shall be the ones complement of the remainder generated by the modulo 2 division of the protected PLCP fields by the polynomial

$$x^{16} + x^{12} + x^{5} + 1$$

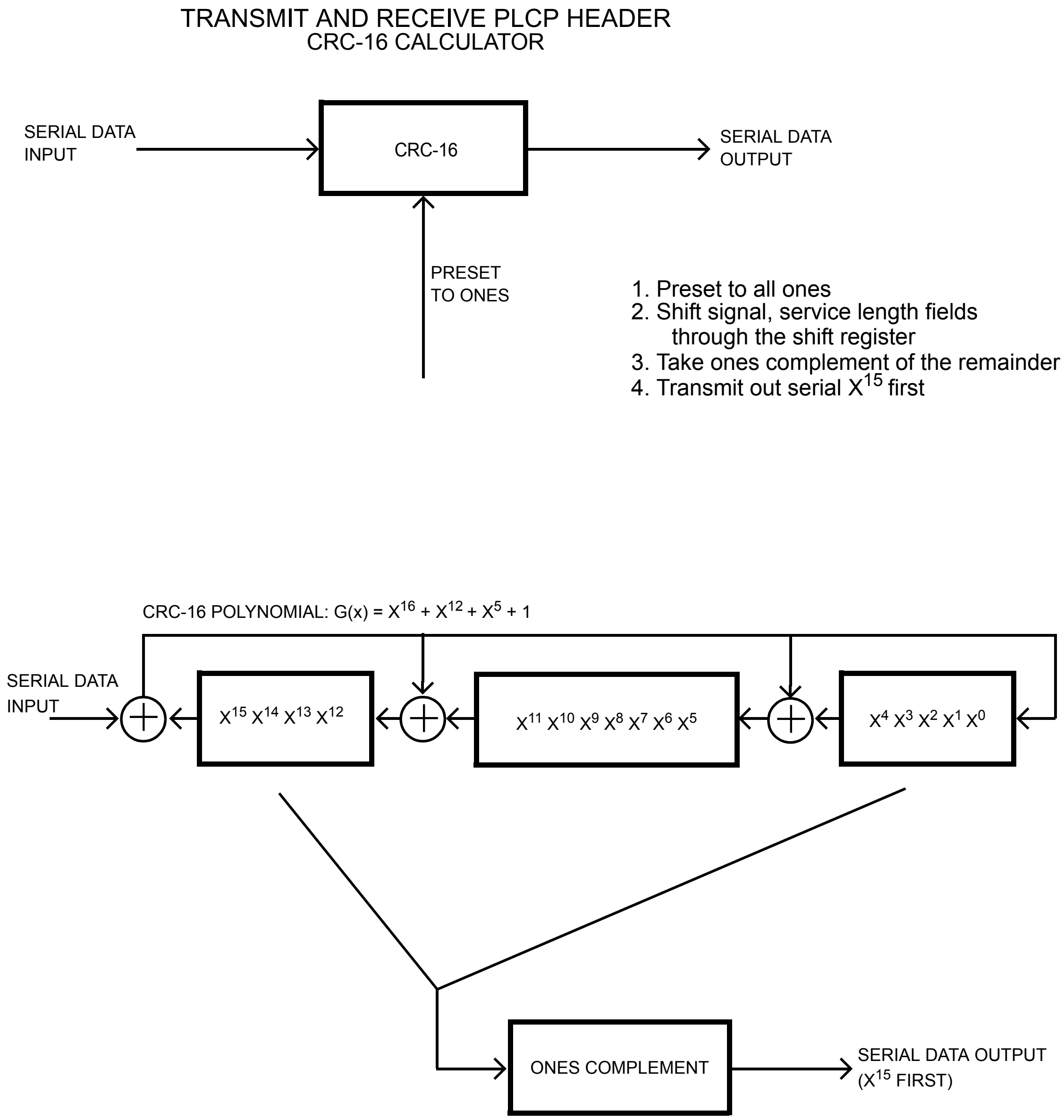
The protected bits shall be processed in transmit order. All FCS calculations shall be made prior to data scrambling. A schematic of the processing is shown in Figure 17-3.

As an example, the SIGNAL, SERVICE, and LENGTH fields for a DBPSK signal with a PPDU length of 192 µs (24 octets) would be given by the following:
0101 0000 0000 0000 0000 0011 0000 0000 [leftmost bit (b0) transmitted first in time]
b0...b48

The ones complement FCS for these protected PLCP preamble bits would be the following:
0101 1011 0101 0111 [leftmost bit (b0) transmitted first in time]
b0.........................b16

Figure 17-3 depicts this example.

An illustrative example of the CRC-16 FCS using the information from Figure 17-3 is shown in Figure 17-4.



1. Preset to all ones
2. Shift signal, service length fields through the shift register
3. Take ones complement of the remainder
4. Transmit out serial X^{15} first

Figure 17-3—CRC-16 implementation

```
Data    CRC Registers
        MSB          LSB

        1111111111111111     ; Initialize preset to ones
 0      1110111111011111
 1      1101111110111110
 0      1010111101011101
 1      0101111010111010
 0      1011110101110100
 0      0110101011001001
 0      1101010110010010
 0      1011101100000101
 0      0110011000101011
 0      1100110001010110
 0      1000100010001101
 0      0000000100111011
 0      0000001001110110
 0      0000010011101100
 0      0000100111011000
 0      0001001110110000
 0      0010011101100000
 0      0100111011000000
 0      1001110110000000
 0      0010101100100001
 0      0101011001000010
 0      1010110010000100
 1      0101100100001000
 1      1010001000110001
 0      0101010001000011
 0      1010100010000110
 0      0100000100101101
 0      1000001001011010
 0      0001010010010101
 0      0010100100101010
 0      0101001001010100
 0      1010010010101000
        0101101101010111     ; ones complement, result = CRC FCS parity
```

Figure 17-4—Example of CRC calculation

17.2.3.8 Long PLCP data modulation and modulation rate change

The long PLCP preamble and header shall be transmitted using the 1 Mb/s DBPSK modulation. The SIGNAL and SERVICE fields combined shall indicate the modulation that shall be used to transmit the PSDU. The SIGNAL field indicates the rate, and the SERVICE field indicates the modulation. The transmitter and receiver shall initiate the modulation and rate indicated by the SIGNAL and SERVICE fields, starting with the first octet of the PSDU. The PSDU transmission rate shall be set by the DATARATE parameter in the TXVECTOR, issued with the PHY-TXSTART.request primitive described in 17.4.4.2.

17.2.3.9 Short PLCP synchronization (shortSYNC)

The shortSYNC field shall consist of 56 bits of scrambled 0 bits. This field is provided so the receiver can perform the necessary synchronization operations. The initial state of the scrambler (seed) shall be [001 1011], where the left end bit specifies the value to place in the first delay element (Z^1) in Figure 17-5 (in 17.2.4), and the right end bit specifies the value to place in the last delay element (Z^7).

17.2.3.10 Short PLCP SFD field (shortSFD)

The shortSFD shall be a 16-bit field and be the time reverse of the field of the SFD in the long PLCP preamble (17.2.3.3). The field is the bit pattern 0000 0101 1100 1111. The right end bit shall be transmitted first in time. A receiver not configured to use the short header option does not detect this SFD.

17.2.3.11 Short PLCP SIGNAL field (shortSIGNAL)

The 8-bit SIGNAL field of the short header indicates to the PHY the data rate that shall be used for transmission (and reception) of the PSDU. A PHY operating with the HR/DSSS/short option supports three mandatory rates given by the following 8-bit words, where the LSB shall be transmitted first in time and the number represents the rate in units of 100 kBit/s:

a) X'14' (MSB to LSB) for 2 Mb/s;
b) X'37'(MSB to LSB) for 5.5 Mb/s;
c) X'6E' (MSB to LSB) for 11 Mb/s.

17.2.3.12 Short PLCP SERVICE field (shortSERVICE)

The SERVICE field in the short header shall be the same as the SERVICE field described in 17.2.3.5.

17.2.3.13 Short PLCP LENGTH field (shortLENGTH)

The LENGTH field in the short header shall be the same as the LENGTH field described in 17.2.3.6.

17.2.3.14 Short CRC-16 field (shortCRC)

The CRC in the short header shall be the same as the CRC field defined in 17.2.3.7. The CRC-16 is calculated over the shortSIGNAL, shortSERVICE, and shortLENGTH fields.

17.2.3.15 Short PLCP data modulation and modulation rate change

The short PLCP preamble shall be transmitted using the 1 Mb/s DBPSK modulation. The short PLCP header shall be transmitted using the 2 Mb/s modulation. The SIGNAL and SERVICE fields combined shall indicate the modulation that shall be used to transmit the PSDU. The SIGNAL field indicates the rate, and the SERVICE field indicates the modulation. The transmitter and receiver shall initiate the modulation and rate indicated by the SIGNAL and SERVICE fields, starting with the first octet of the PSDU. The PSDU transmission rate shall be set by the DATARATE parameter in the TXVECTOR, issued with the PHY-TXSTART.request primitive described in 17.4.4.2.

17.2.4 PLCP/High Rate PHY data scrambler and descrambler

The polynomial $G(z) = z^{-7} + z^{-4} + 1$ shall be used to scramble all bits transmitted. The feedthrough configuration of the scrambler and descrambler is self-synchronizing, which requires no prior knowledge of the transmitter initialization of the scrambler for receive processing. Figure 17-5 and Figure 17-6 show typical implementations of the data scrambler and descrambler, but other implementations are possible.

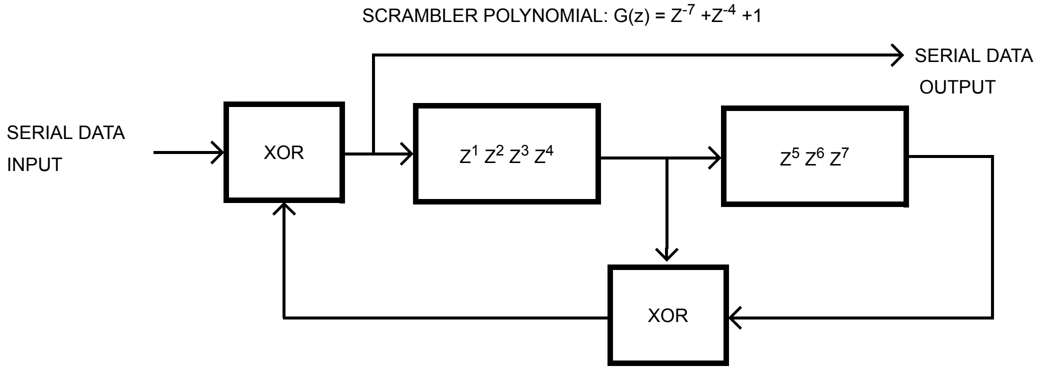


Figure 17-5—Data scrambler

The scrambler shall be initialized as specified in 17.2.3.9 for the short PLCP and 17.2.3.2 for the long PLCP. For a long preamble, this shall result in the scrambler registers Z^1 to Z^7 in Figure 17-5 having the data pattern [1101100] (i.e., Z^1= 1... Z^7= 0) when the scrambler is first started. The scrambler shall be initialized with the reverse pattern [0011011] when transmitting the optional short preamble.

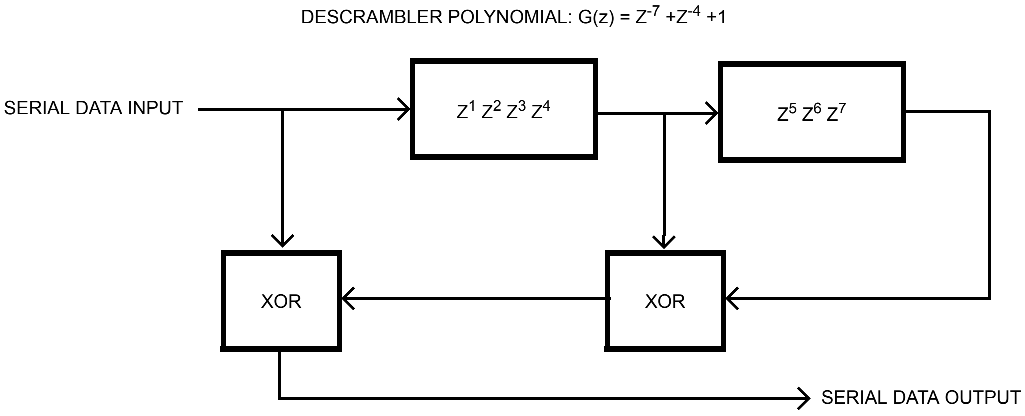


Figure 17-6—Data descrambler

17.2.5 Transmit PLCP

The transmit procedures for a High Rate PHY using the long PLCP preamble and header are the same as the transmit procedures described in 16.2.6 and 16.2.7 and do not change apart from the ability to transmit 5.5 Mb/s and 11 Mb/s.

The procedures for a transmitter employing HR/DSSS/short and HR/DSSS/PBCC/short are the same except for length and rate changes. The decision to use a long or short PLCP is beyond the scope of this standard.

The transmit PLCP is shown in Figure 17-7.

A PHY-TXSTART.request(TXVECTOR) primitive is issued by the MAC to start the transmission of a PPDU. In addition to parameters DATARATE and LENGTH, other transmit parameters such as PREAMBLE_TYPE and MODULATION are set via the PHY-SAP with the PHY-TXSTART.request(TXVECTOR) primitive, as described in 17.3.5. The SIGNAL, SERVICE, and LENGTH fields of the PLCP header are calculated as described in 17.2.3.

The PLCP shall issue PMD_ANTSEL, PMD_RATE, and PMD_TXPWRLVL primitives to configure the PHY. The PLCP shall then issue a PMD_TXSTART.request primitive, and the PHY entity shall immediately initiate data scrambling and transmission of the PLCP preamble based on the parameters passed in the PHY-TXSTART.request primitive. The time required for transmit power-on ramp, described in 17.4.7.7, shall be included in the PLCP SYNC field. If dot11MgmtOptionTODImplemented and dot11MgmtOptionTODActivated are true or if dot11MgmtOptionTimingMsmtActivated is true and the TXVECTOR parameter TIME_OF_DEPARTURE_REQUESTED is true, then the PLCP shall issue a PHY_TXSTART.confirm(TXSTATUS) primitive to the MAC, forwarding the TIME_OF_DEPARTURE corresponding to the time when the first frame energy is sent by the transmitting port and TIME_OF_DEPARTURE_ClockRate parameter within the TXSTATUS vector. If dot11MgmtOptionTimingMsmtActivated is true, then the PLCP shall forward the value of TX_START_OF_FRAME_OFFSET in TXSTATUS vector. Once the PLCP preamble transmission is complete, data shall be exchanged between the MAC and the PHY by a series of PHY-DATA.request(DATA) primitives issued by the MAC and PHY-DATA.confirm primitives issued by the PHY. The modulation and rate change, if any, shall be initiated with the first data symbol of the PSDU, as described in 17.2.3.8 and 17.2.3.15. The PHY proceeds with PSDU transmission through a series of data octet transfers from the MAC. At the PMD layer, the data octets are sent in LSB-to-MSB order and presented to the PHY through PMD_DATA.request primitives. Transmission can be prematurely terminated by the MAC through the PHY-TXEND.request primitive. PHY-TXSTART shall be disabled by the issuance of the PHY-TXEND.request primitive. Normal termination occurs after the transmission of the final bit of the last PSDU octet, calculated from the number supplied in the PHY preamble LENGTH and SERVICE fields using the equations specified in 17.2.3.6. The PPDU transmission shall be completed and the PHY entity shall enter the receive state (i.e., PHY-TXSTART shall be disabled). It is recommended that modulation continue during power-down to prevent radiating a continuous wave carrier. Each PHY-TXEND.request primitive is acknowledged with a PHY-TXEND.confirm primitive from the PHY.

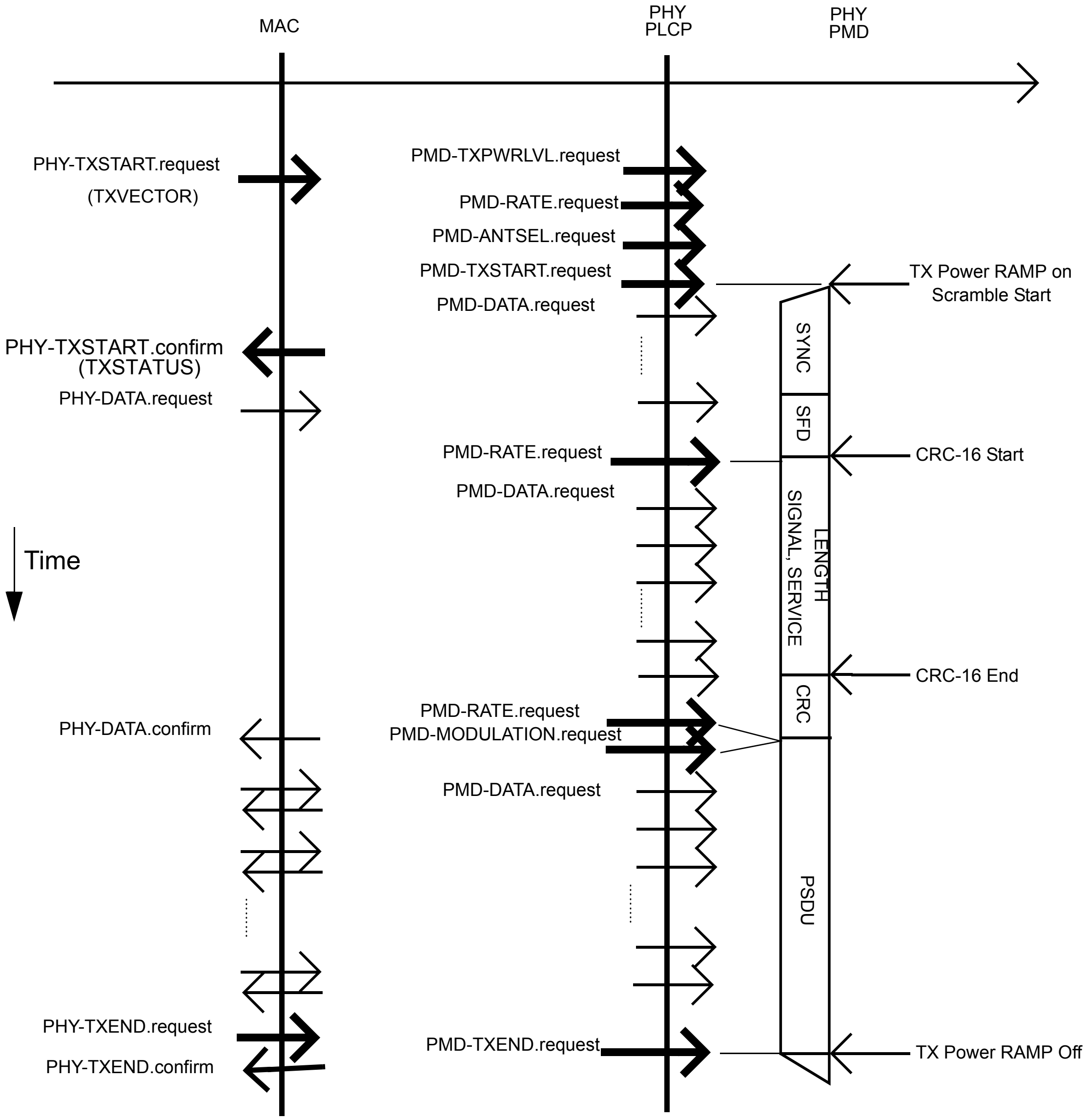


Figure 17-7—Transmit PLCP

17.2.6 Receive PLCP

The receive procedures for receivers configured to receive the mandatory and optional PLCPs, rates, and modulations are described in this subclause. A receiver that supports this High Rate extension of the standard is capable of receiving 5.5 Mb/s and 11 Mb/s, in addition to 1 Mb/s and 2 Mb/s. If the PHY implements the short preamble option, it shall detect both short and long preamble formats and indicate which type of preamble was received in the RXVECTOR. If the PHY implements the PBCC modulation option, it shall detect either CCK or PBCC modulations, as indicated in the SIGNAL field, and shall report the type of modulation used in the RXVECTOR.

The receiver shall implement the CCA procedure as defined in 17.4.8.5. Upon receiving a PPDU, the receiver shall distinguish between a long and short header format by the value of the SFD, as specified in 17.2.2. The receiver shall demodulate a long PLCP header using BPSK at 1 Mb/s. The receiver shall demodulate a short PLCP header using QPSK at 2 Mb/s. The receiver shall use the SIGNAL and SERVICE fields of the PLCP header to determine the data rate and modulation of the PSDU.

The receive PLCP is shown in Figure 17-8. In order to receive data, the PHY-TXSTART.request primitive shall be disabled so that the PHY entity is in the receive state. Further, through station management via the PLME, the PHY shall be set to the appropriate channel and the CCA method chosen. Other receive parameters, such as RSSI, RCPI, SQ, and indicated DATARATE, may be accessed via the PHY-SAP.

Upon receiving the transmitted energy, according to the selected CCA mode, the PMD_ED shall be enabled (according to 17.4.8.5) as the RSSI reaches the ED_THRESHOLD, and/or PMD_CS shall be enabled after code lock is established. These conditions are used to indicate activity to the MAC via the PHY-CCA.indication primitive, according to 17.4.8.5. A PHY-CCA.indication(BUSY) primitive shall be issued for ED and/or code lock prior to correct reception of the PLCP header. The PMD primitives, PMD_SQ, PMD_RSSI, and PMD_RCPI are issued to update the SQ, RSSI, and RCPI parameters reported to the MAC.

After a PHY-CCA.indication primitive is issued, the PHY entity shall begin searching for the SFD field. Once the SFD field is detected, CRC-16 processing shall be initiated and the PLCP SIGNAL, SERVICE, and LENGTH fields shall be received. The CRC-16 FCS shall be processed. If the CRC-16 FCS check fails, the PHY receiver shall return to the RX IDLE state, as depicted in Figure 17-9. Should the status of CCA return to the IDLE state during reception prior to completion of the full PLCP processing, the PHY receiver shall return to the RX IDLE state.

If the PLCP header reception is successful (and the SIGNAL field is completely recognizable and supported), a PHY-RXSTART.indication(RXVECTOR) primitive shall be issued. The RXVECTOR associated with this primitive includes:

a) The SIGNAL field
b) The SERVICE field
c) The PSDU length in octets (calculated from the LENGTH field in microseconds and the DATARATE in Mb/s, in accordance with the formula in 17.2.3.6)
d) RXPREAMBLE_TYPE (which is an enumerated type taking on values SHORTPREAMBLE or LONGPREAMBLE)
e) ANT_STATE (the antenna used for receive), RSSI, RCPI, and SQ

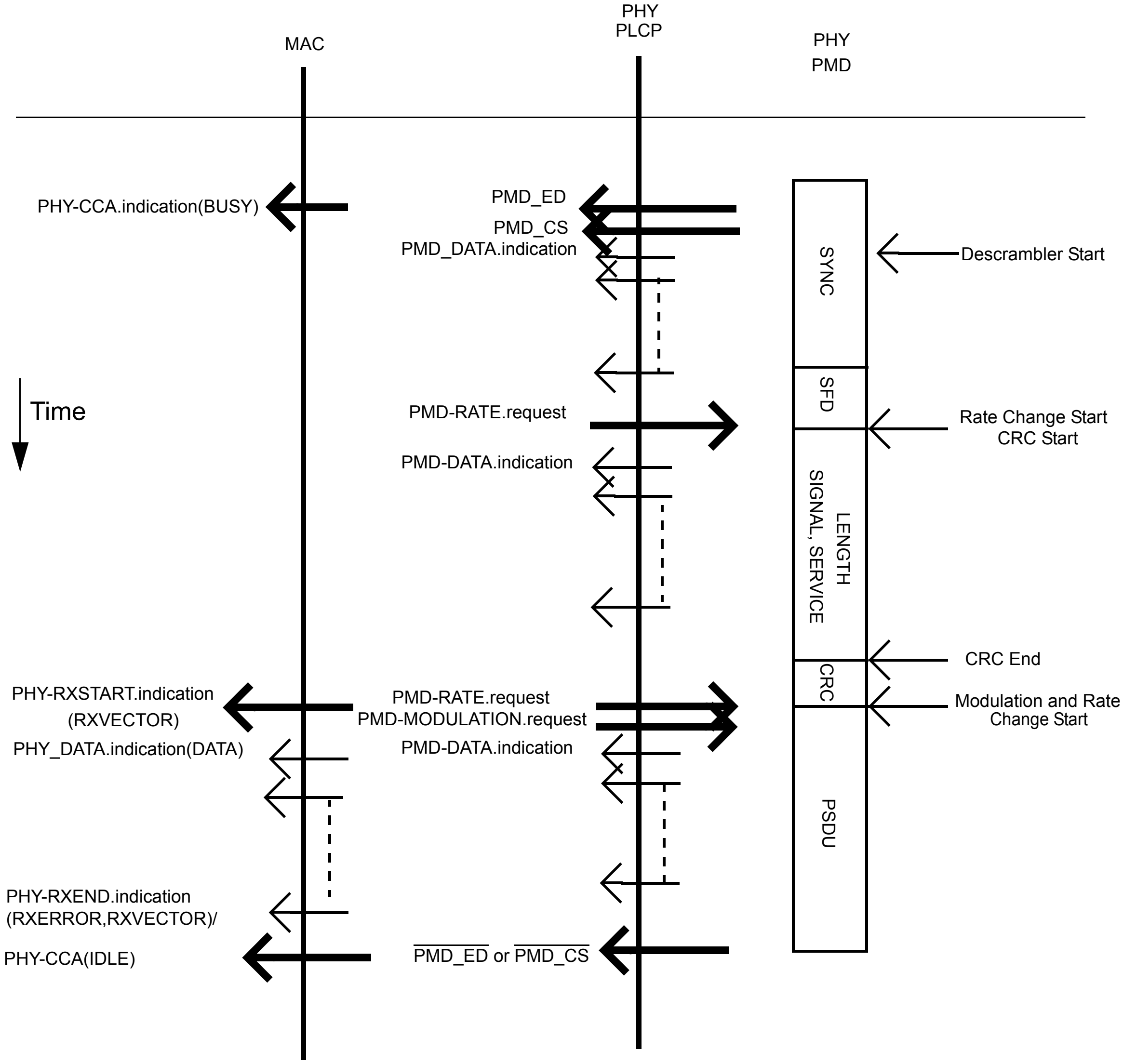


Figure 17-8—Receive PLCP

If dot11MgmtOptionTimingMsmtActivated is true, the PLCP shall do the following:

— Complete receiving the PLCP header and verify the validity of the PLCP Header.

— If the PLCP header reception is successful (and the SIGNAL field is completely recognizable and supported), a PHY-RXSTART.indication(RXVECTOR) shall be issued and RX_START_OF_FRAME_OFFSET parameter within the RXVECTOR shall be forwarded (see 17.3.5).

NOTE—The RX_START_OF_FRAME_OFFSET value is used as described in 6.3.57 in order to estimate when the start of the preamble for the incoming frame was detected on the medium at the receive antenna port.

The received PSDU bits are assembled into octets and presented to the MAC using a series of PHY-DATA.indication(DATA) primitive exchanges. The rate and modulation change indicated in the SIGNAL field shall be initiated with the first symbol of the PSDU, as described in 17.2.5. The PHY proceeds with PSDU reception. After reception of the final bit of the last PSDU octet, indicated by the PLCP preamble LENGTH field, the receiver shall be returned to the RX IDLE state shown in Figure 17-9.

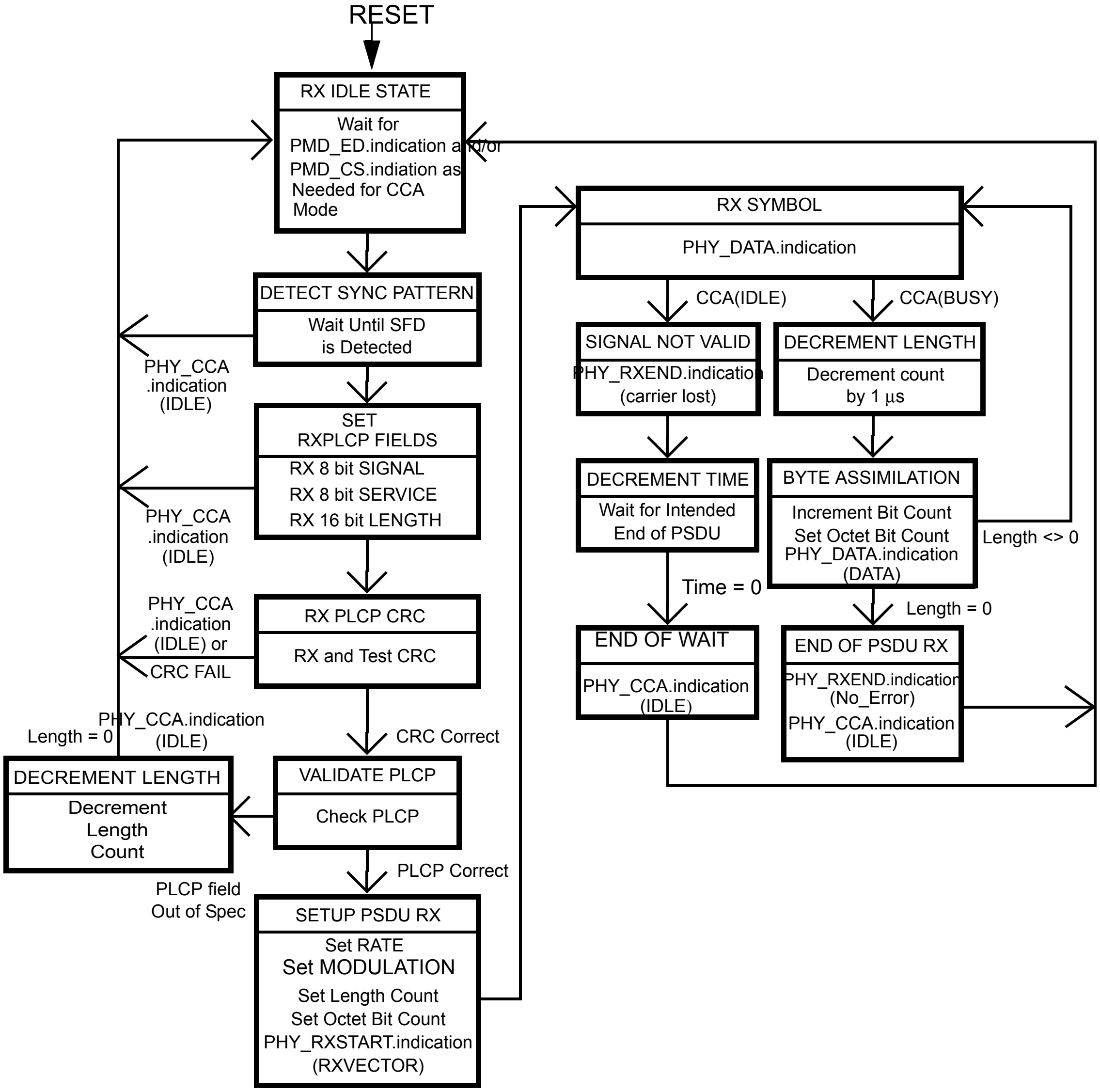


Figure 17-9—PLCP receive state machine

A PHY-RXEND.indication(NoError) primitive shall be issued. A PHY-CCA.indication(IDLE) primitive shall be issued following a change in PHYCS and/or PHYED according to the selected CCA method.

In the event that a change in PHYCS or PHYED would cause the status of CCA to return to the IDLE state before the complete reception of the PSDU, as indicated by the PLCP LENGTH field, the error condition shall be reported to the MAC using a PHY-RXEND.indication(CarrierLost) primitive. The High Rate PHY shall ensure that the CCA indicates a busy medium for the intended duration of the transmitted PPDU.

If the PLCP header is successful, but the indicated rate or modulation in the SIGNAL and SERVICE fields is not within the capabilities of the receiver, a PHY-RXSTART.indication primitive shall not be issued. The PHY shall indicate the error condition using a PHY-RXEND.indication(UnsupportedRate) primitive. If the PLCP header is invalid, a PHY-RXSTART.indication primitive shall not be issued, and the PHY shall indicate the error condition using a PHY-RXEND.indication(FormatViolation) primitive. Also, in both cases, the High Rate PHY shall ensure that the CCA indicates a busy medium for the intended duration of the transmitted PSDU, as indicated by the LENGTH field. The intended duration is indicated by the LENGTH field (LENGTH × 1 μs).

A typical state machine implementation of the receive PLCP is shown in Figure 17-9.

17.3 High Rate PLME

17.3.1 PLME_SAP sublayer management primitives

Table 17-4 lists the MIB attributes that may be accessed by the PHY entities and intralayer or higher level LMEs. These attributes are accessed via the PLME-GET, PLME-SET, and PLME-RESET primitives defined in Clause 6.

17.3.2 High Rate PHY MIB

All High Rate PHY MIB attributes are defined in Annex C, with specific values defined in Table 17-4.

Table 17-4—MIB attribute default values/ranges

Managed object	Default value/range	Operational semantics
dot11PhyOperationTable		
dot11PHYType	High Rate–2.4 (X‘05’)	Static
dot11CurrentRegDomain	Implementation dependent	Static
dot11PhyHRDSSSEntryTable		
dot11ShortPreambleOptionImplemented	Implementation dependent	Static
dot11PBCCOptionImplemented	Implementation dependent	Static
dot11ChannelAgility Present	Implementation dependent	Static
dot11ChannelAgilityActivated	false/Boolean	Dynamic
dot11PhyAntennaTable		
dot11CurrentTxAntenna	Implementation dependent	Dynamic
dot11DiversitySupportImplemented	Implementation dependent	Static
dot11CurrentRxAntenna	Implementation dependent	Dynamic
dot11PhyTxPowerTable		
dot11NumberSupportedPowerLevelsImplemented	Implementation dependent	Static
dot11TxPowerLevel1	Implementation dependent	Static
dot11TxPowerLevel2	Implementation dependent	Static
dot11TxPowerLevel3	Implementation dependent	Static
dot11TxPowerLevel4	Implementation dependent	Static
dot11TxPowerLevel5	Implementation dependent	Static
dot11TxPowerLevel6	Implementation dependent	Static
dot11TxPowerLevel7	Implementation dependent	Static
dot11TxPowerLevel8	Implementation dependent	Static
dot11CurrentTxPowerLevel	Implementation dependent	Dynamic

Table 17-4—MIB attribute default values/ranges *(continued)*

Managed object	Default value/range	Operational semantics
dot11PhyDSSSTable		
dot11CurrentChannel	Implementation dependent	Dynamic
dot11CCAModeSupported	Implementation dependent	Static
dot11CurrentCCAMode	Implementation dependent	Dynamic
dot11EDThreshold	Implementation dependent	Dynamic
dot11AntennasListTable		
dot11SupportTxAntenna	Implementation dependent	Static
dot11SupportRxAntenna	Implementation dependent	Static
dot11DiversitySelectionRxImplemented	Implementation dependent	Dynamic
dot11RegDomainsSupportedTable		
dot11RegDomainsImplementedValue	Implementation dependent	Static
dot11SupportedDataRatesTx	Table Tx X’02’, X’04’, X’0B’, X’16’	Static
dot11SupportedDataRatesRx	Table Rx X’02’, X’04’, 'X’0B’, X’16’	Static
NOTE—The column titled “Operational semantics” contains two types: static and dynamic. Static MIB attributes are fixed and cannot be modified for a given PHY implementation. Dynamic MIB attributes can be modified by some management entities.		

17.3.3 DS PHY characteristics

The static DS PHY characteristics, provided through the PLME-CHARACTERISTICS service primitive, are shown in Table 17-5. The definitions of these characteristics are in 6.5.4.

Table 17-5—High Rate PHY characteristics

Characteristic	Value
aSlotTime	20 µs
aSIFSTime	10 µs
aCCATime	≤ 15 µs
aPHY-RX-START-Delay	192 µs for long preamble and 96 µs for short preamble
aRxTxTurnaroundTime	≤ 5 µs
aTxPLCPDelay	Implementers may choose any value for this delay as long as the requirements of aRxTxTurnaroundTime are met.
aRxPLCPDelay	Implementers may choose any value for this delay as long as the requirements of aSIFSTime and aCCATime are met.
aRxTxSwitchTime	≤ 5 µs
aTxRampOnTime	Implementers may choose any value for this delay as long as the requirements of aRxTxTurnaroundTime are met.
aTxRampOffTime	Implementers may choose any value for this delay as long as the requirements of aSIFSTime are met.
aTxRFDelay	Implementers may choose any value for this delay as long as the requirements of aRxTxTurnaroundTime are met.
aRxRFDelay	Implementers may choose any value for this delay as long as the requirements of aSIFSTime and aCCATime are met.

Table 17-5—High Rate PHY characteristics *(continued)*

Characteristic	Value
aAirPropagationTime	1 µs
aMACProcessingDelay	≤ 2 µs
aPreambleLength	144 µs
aPLCPHeaderLength	48 µs
aMPUMaxLength	$14 \le x \le (2^{12} - 1)$
aCWmin	31
aCWmax	1023

17.3.4 High Rate TXTIME calculation

The value of the TXTIME parameter returned by the PLME-TXTIME.confirm primitive shall be calculated according to the following equation:

$$\text{TXTIME} = \text{PreambleLength} + \text{PLCPHeaderTime} + \text{Ceiling}(((\text{LENGTH}+\text{PBCC}) \times 8) / \text{DATARATE})$$

where

LENGTH and DATARATE are values from the TXVECTOR parameter of the corresponding PLME-TXTIME.request primitive

LENGTH is in units of octets

DATARATE is in units of Mb/s

Ceiling is a function that returns the smallest integer value greater than or equal to its argument value

PBCC has a value of 1 if the SIGNAL value from the TXVECTOR parameter specifies PBCC and has a value of 0 otherwise

The value of PreambleLength is 144 µs if the TXPREAMBLE_TYPE value from the TXVECTOR parameter indicates “LONGPREAMBLE,” or 72 µs if the TXPREAMBLE_TYPE value from the TXVECTOR parameter indicates “SHORTPREAMBLE”

The value of PLCPHeaderTime is 48 µs if the TXPREAMBLE_TYPE value from the TXVECTOR parameter indicates “LONGPREAMBLE,” or 24 µs if the TXPREAMBLE_TYPE value from the TXVECTOR parameter indicates “SHORTPREAMBLE”

17.3.5 Vector descriptions

Several service primitives include a parameter vector. These vectors are a list of parameters as described in Table 17-6. DATARATE and LENGTH are described in 7.3.4.5. The remaining parameters are considered to be management parameters and are specific to this PHY.

Table 17-6—Parameter vectors

Parameter	Associated vector	Value
DATARATE	RXVECTOR, TXVECTOR	The rate used to transmit the PSDU in Mb/s.
LENGTH	RXVECTOR, TXVECTOR	The length of the PSDU in octets.

Table 17-6—Parameter vectors *(continued)*

Parameter	Associated vector	Value
PREAMBLE_TYPE	RXVECTOR, TXVECTOR	The preamble used for the transmission of this PPDU. This is an enumerated type that takes the value SHORTPREAMBLE or LONGPREAMBLE.
MODULATION	RXVECTOR, TXVECTOR	The modulation used for the transmission of this PSDU. This is an integer where 0 means CCK and 1 means PBCC.
ANT_STATE	RXVECTOR	1–256
RSSI	RXVECTOR	0–8 bits of RSSI
RCPI (see NOTE)	RXVECTOR	0–255
SQ	RXVECTOR	0–8 bits of SQ
TIME_OF_DEPARTURE_REQUESTED	TXVECTOR	false, true. When true, the MAC entity requests that the PHY PLCP entity measures and reports time of departure parameters corresponding to the time when the first frame energy is sent by the transmitting port; when false, the MAC entity requests that the PHY PLCP entity neither measures nor reports time of departure parameters.
TIME_OF_DEPARTURE	TXSTATUS	0 to 2^{32}– 1. The time when the first frame energy is sent by the transmitting port, measured by the local PHY entity, in units equal to 1/TIME_OF_DEPARTURE_ClockRate. This parameter is present only if TIME_OF_DEPARTURE_REQUESTED is true in the corresponding request.
TIME_OF_DEPARTURE_ClockRate	TXSTATUS	0 to 2^{16}– 1. The clock rate, in units of MHz, is used to generate the TIME_OF_DEPARTURE value. This parameter is present only if TIME_OF_DEPARTURE_REQUESTED is true in the corresponding request.
TX_START_OF_FRAME_OFFSET	TXSTATUS	0 to 2^{32}– 1. An estimate of the offset (in 10 ns units) from the point in time at which the start of the preamble corresponding to the frame was transmitted at the transmit antenna port to the point in time at which this primitive is issued to the MAC.
RX_START_OF_FRAME_OFFSET	RXVECTOR	0 to 2^{32}– 1. An estimate of the offset (in 10 ns units) from the point in time at which the start of the preamble corresponding to the incoming frame arrived at the receive antenna port to the point in time at which this primitive is issued to the MAC.
NOTE—RCPI is present only when dot11RadioMeasurementActivated is true.		

17.4 High Rate PMD sublayer

17.4.1 Scope and field of application

The PMD services provided to the PLCP for the High Rate PHY are described in 17.4. Also defined in 17.4 are the functional, electrical, and RF characteristics required for interoperability of implementations conforming to this specification. The relationship of this specification to the entire High Rate PHY is shown in Figure 17-10.

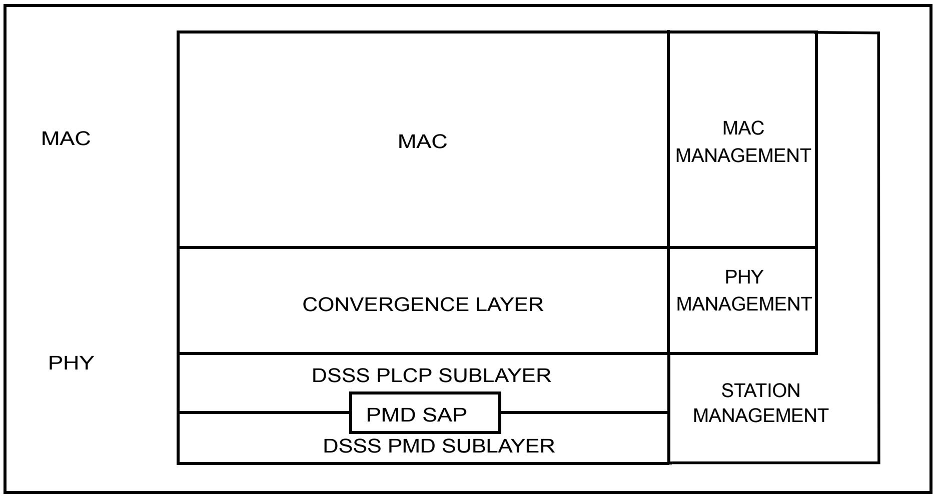


Figure 17-10—Layer reference model

17.4.2 Overview of service

The High Rate PMD sublayer accepts PLCP sublayer service primitives and provides the actual means by which data are transmitted or received from the medium. The combined functions of the High Rate PMD sublayer primitives and parameters for the receive function result in a data stream, timing information, and associated receive signal parameters being delivered to the PLCP sublayer. A similar functionality is provided for data transmission.

17.4.3 Overview of interactions

The primitives associated with the PLCP sublayer to the High Rate PMD fall into two basic categories:

a) Service primitives that support PLCP peer-to-peer interactions;

b) Service primitives that have local significance and that support sublayer-to-sublayer interactions.

17.4.4 Basic service and options

17.4.4.1 General

All of the service primitives described in 17.4.4 are considered mandatory, unless otherwise specified.

17.4.4.2 PMD_SAP peer-to-peer service primitives

Table 17-7 indicates the primitives for peer-to-peer interactions.

Table 17-7—PMD_SAP peer-to-peer service primitives

Primitive	Request	Indicate	Confirm	Response
PMD_DATA	X	X	—	—

17.4.4.3 PMD_SAP sublayer-to-sublayer service primitives

Table 17-8 indicates the primitives for sublayer-to-sublayer interactions.

Table 17-8—PMD_SAP sublayer-to-sublayer service primitives

Primitive	Request	Indicate	Confirm	Response
PMD_TXSTART	X	—	—	—
PMD_TXEND	X	—	—	—
PMD_ANTSEL	X	X	—	—
PMD_TXPWRLVL	X	—	—	—
PMD_MODULATION	X	X	—	—
PMD_PREAMBLE	X	X	—	—
PMD_RATE	X	X	—	—
PMD_RSSI	—	X	—	—
PMD_SQ	—	X	—	—
PMD_CS	—	X	—	—
PMD_ED	X	X	—	—
PMD_RCPI	—	X	—	—

17.4.5 PMD_SAP detailed service specification

17.4.5.1 Introduction

The services provided by each PMD primitive are described in 17.4.5.2 to 17.4.5.16.

17.4.5.2 PMD_DATA.request

17.4.5.2.1 Function

This primitive defines the transfer of data from the PLCP sublayer to the PMD entity.

17.4.5.2.2 Semantics of the service primitive

This primitive provides the following parameter:

Parameter	Associated primitive	Value (Mb/s)	Description
TXD_UNIT	PMD_DATA.request	0,1: 1 00,01,11,10:2 X'0'–X'F': 5.5 X'00'–X'FF': 11	This parameter represents a single block of data, which, in turn, is used by the PMD to be differentially encoded into a transmitted symbol. The symbol itself is spread by the PN code prior to transmission.

17.4.5.2.3 When generated

This primitive is generated by the PLCP sublayer to request transmission of a symbol. The data clock for this primitive is supplied by the PMD layer based on the PN code repetition.

17.4.5.2.4 Effect of receipt

The PMD performs the differential encoding, PN code modulation, and transmission of data.

17.4.5.3 PMD_DATA.indication

17.4.5.3.1 Function

This primitive defines the transfer of data from the PMD entity to the PLCP sublayer.

17.4.5.3.2 Semantics of the service primitive

This primitive provides the following parameter:

Parameter	Associated primitive	Value (Mb/s)	Description
RXD_UNIT	PMD_DATA.indication	0,1: 1 00,01,11,10:2 X'0' - X'F': 5.5 X'00' - X'FF': 11	This parameter represents a single symbol that has been demodulated by the PMD entity.

17.4.5.3.3 When generated

This primitive, which is generated by the PMD entity, forwards received data to the PLCP sublayer. The data clock for this primitive is supplied by the PMD layer based on the PN code repetition.

17.4.5.3.4 Effect of receipt

The PLCP sublayer either interprets the bit or bits that are recovered as part of the PLCP or passes the data to the MAC sublayer as part of the PSDU.

17.4.5.4 PMD_MODULATION.request

17.4.5.4.1 Function

This primitive, which is generated by the PHY PLCP sublayer, selects the modulation code that is used by the High Rate PHY for transmission.

17.4.5.4.2 Semantics of the service primitive

This primitive provides the following parameter:

Parameter	Associated primitive	Value	Description
MODULATION	PMD_MODULATION. request PMD_MODULATION. indication	1 MbBarker, 2 MbBarker, 5.5 CCK, 11 CCK, 5.5 PBCC, or 11 PBCC	In Receive mode, the MODULATION parameter informs the PLCP layer which PHY data modulation was used to process the PSDU portion of the PPDU. See 17.4.6.4 for further information on the High Rate PHY modulation codes.

17.4.5.4.3 When generated

This primitive is generated by the PLCP sublayer to change or set the current High Rate PHY modulation code used for the PSDU portion of a PPDU. The PMD_MODULATION.request primitive is normally issued prior to issuing the PMD_TXSTART command.

17.4.5.4.4 Effect of receipt

The receipt of PMD_MODULATION selects the modulation that is used for all subsequent PSDU transmissions. This code is used for transmission only. The High Rate PHY shall still be capable of receiving all the required High Rate PHY modulations. This primitive, which is generated by the PMD entity, sets the state of the PHY for demodulation of the appropriate modulation.

17.4.5.5 PMD_PREAMBLE.request

17.4.5.5.1 Function

This primitive, which is generated by the PHY PLCP sublayer, selects the preamble mode that is used by the High Rate PHY for transmission.

17.4.5.5.2 Semantics of the service primitive

This primitive provides the following parameter:

Parameter	Associated primitive	Value	Description
PREAMBLE	PMD_PREAMBLE.request	0 for long 1 for short	PREAMBLE selects which of the High Rate PHY preamble types is used for PLCP transmission. See 17.2.2 for further information on the High Rate PHY preamble modes.

17.4.5.5.3 When generated

This primitive is generated by the PLCP sublayer to change or set the current High Rate PHY preamble mode used for the PLCP portion of a PPDU. The PMD_PREAMBLE.request primitive is normally issued prior to issuing the PMD_TXSTART command.

17.4.5.5.4 Effect of receipt

The receipt of PMD_PREAMBLE selects the preamble mode that is used for all subsequent PSDU transmissions. This mode is used for transmission only. The High Rate PHY shall still be capable of receiving all the required High Rate PHY preambles. This primitive sets the state of the PHY for modulation of the appropriate mode.

17.4.5.6 PMD_PREAMBLE.indication

17.4.5.6.1 Function

This primitive, which is generated by the PMD sublayer, indicates which preamble mode was used to receive the PLCP portion of the PPDU.

17.4.5.6.2 Semantics of the service primitive

This primitive provides the following parameter:

Parameter	Associated primitive	Value	Description
PREAMBLE	PMD_PREAMBLE. indication	0 for long 1 for short	In RECEIVE mode, the PREAMBLE parameter informs the PLCP layer which of the High Rate PHY preamble modes was used to send the PLCP portion of the PPDU.

17.4.5.6.3 When generated

This primitive is generated by the PMD sublayer when the PLCP preamble has been properly detected.

17.4.5.6.4 Effect of receipt

This parameter is provided to the PLCP layer for information only.

17.4.5.7 PMD_TXSTART.request

17.4.5.7.1 Function

As a result of receiving a PHY_DATA.request primitive from the MAC, the PLCP issues this primitive, which initiates PPDU transmission by the PMD layer.

17.4.5.7.2 Semantics of the service primitive

This primitive has no parameters.

17.4.5.7.3 When generated

This primitive is generated by the PLCP sublayer to initiate the PMD layer transmission of the PPDU. The PHY-DATA.request primitive is provided to the PLCP sublayer prior to issuing the PMD_TXSTART command.

17.4.5.7.4 Effect of receipt

PMD_TXSTART initiates transmission of a PPDU by the PMD sublayer.

17.4.5.8 PMD_TXEND.request

17.4.5.8.1 Function

This primitive, which is generated by the PHY PLCP sublayer, ends PPDU transmission by the PMD layer.

17.4.5.8.2 Semantics of the service primitive

This primitive has no parameters.

17.4.5.8.3 When generated

This primitive is generated by the PLCP sublayer to terminate the PMD layer transmission of the PPDU.

17.4.5.8.4 Effect of receipt

PMD_TXEND terminates transmission of a PPDU by the PMD sublayer.

17.4.5.9 PMD_ANTSEL.request

17.4.5.9.1 Function

This primitive, which is generated by the PHY PLCP sublayer, selects the antenna used by the PHY for transmission or reception (when diversity is disabled).

17.4.5.9.2 Semantics of the service primitive

This primitive provides the following parameter:

Parameter	Associated primitive	Value	Description
ANT_STATE	PMD_ANTSEL.request PMD_ANTSEL.indication	1 to 256	ANT_STATE selects which of the available antennas should be used for transmit. The number of available antennas is determined from the MIB table parameters, aSuprtRxAntennas and aSuprtTxAntennas.

17.4.5.9.3 When generated

This primitive is generated by the PLCP sublayer to select a specific antenna for transmission (or reception when diversity is disabled).

17.4.5.9.4 Effect of receipt

PMD_ANTSEL immediately selects the antenna specified by ANT_STATE.

17.4.5.10 PMD_TXPWRLVL.request

17.4.5.10.1 Function

This primitive, which is generated by the PHY PLCP sublayer, selects the power level used by the PHY for transmission.

17.4.5.10.2 Semantics of the service primitive

This primitive provides the following parameter:

Parameter	Associated primitive	Value	Description
TXPWR_LEVEL	PMD_TXPWRLVL.request	0, 1, 2, 3 (maximum of 4 levels)	TXPWR_LEVEL selects which of the optional transmit power levels should be used for the current PPDU transmission. The number of available power levels is determined by the MIB parameter dot11NumberSupportedPowerLevels. See 17.4.7.3 for further information on the optional High Rate PHY power-level control capabilities.

17.4.5.10.3 When generated

This primitive is generated by the PLCP sublayer to select a specific transmit power. This primitive is applied prior to setting PMD_TXSTART to the transmit state.

17.4.5.10.4 Effect of receipt

PMD_TXPWRLVL immediately sets the transmit power level given by TXPWR_LEVEL.

17.4.5.11 PMD_RATE.request

17.4.5.11.1 Function

This primitive, which is generated by the PHY PLCP sublayer, selects the data rate that shall be used by the High Rate PHY for transmission.

17.4.5.11.2 Semantics of the service primitive

This primitive provides the following parameter:

Parameter	Associated primitive	Value (Mb/s)	Description
RATE	PMD_RATE.indication PMD_RATE.request	X'0A' or 1 X'14' for 2 X'37' for 5.5 X'6E' for 11	RATE selects which of the High Rate PHY data rates is used for PSDU transmission. See 17.4.6.4 for further information on the High Rate PHY data rates. The High Rate PHY rate change capability is described in 17.2.

17.4.5.11.3 When generated

This primitive is generated by the PLCP sublayer to change or set the current High Rate PHY data rate used for the PSDU portion of a PPDU.

17.4.5.11.4 Effect of receipt

The receipt of PMD_RATE selects the rate that is used for all subsequent PSDU transmissions. This rate is used for transmission only. The High Rate PHY shall still be capable of receiving all the required High Rate PHY data rates.

17.4.5.12 PMD_RSSI.indication

17.4.5.12.1 Function

This optional primitive may be generated by the PMD to provide the receive signal strength to the PLCP.

17.4.5.12.2 Semantics of the service primitive

This primitive provides the following parameter:

Parameter	Associated primitive	Value	Description
RSSI	PMD_RSSI.indication	0–8 bits of RSSI	The RSSI is a measure of the RF energy received by the High Rate PHY.

17.4.5.12.3 When generated

This primitive is generated by the PMD when the High Rate PHY is in the receive state. It is continuously available to the PLCP, which, in turn, provides the parameter to the MAC entity.

17.4.5.12.4 Effect of receipt

This parameter is provided to the PLCP layer for information only. The RSSI may be used in conjunction with SQ as part of a CCA scheme.

17.4.5.13 PMD_SQ.indication

17.4.5.13.1 Function

This optional primitive may be generated by the PMD to provide an indication of the SQ of the High Rate PHY PN code correlation to the PLCP. SQ is a measure of the quality of BARKER code lock, providing an effective measure during the full reception of a PLCP preamble and header.

17.4.5.13.2 Semantics of the service primitive

This primitive provides the following parameter:

Parameter	Associated primitive	Value	Description
SQ	PMD_SQ.indication	0–8 bits of SQ	This primitive is a measure of the SQ received by the HR/DSSS PHY.

17.4.5.13.3 When generated

This primitive is generated by the PMD when the High Rate PHY is in the receive state and Barker code lock is achieved. It is continuously available to the PLCP, which, in turn, provides the parameter to the MAC entity.

17.4.5.13.4 Effect of receipt

This parameter is provided to the PLCP layer for information only. The SQ may be used in conjunction with RSSI as part of a CCA scheme.

17.4.5.14 PMD_CS.indication

17.4.5.14.1 General

This primitive, which is generated by the PMD, shall indicate to the PLCP layer that the receiver has acquired (locked) the Barker code and data are being demodulated.

17.4.5.14.2 Function

This primitive, which is generated by the PMD, shall indicate to the PLCP layer that the receiver has acquired (locked) the Barker code and data are being demodulated.

17.4.5.14.3 Semantics of the service primitive

This primitive provides the following parameter:

Parameter	Associated primitive	Value	Description
PMD_CS	PMD_CS.indication	0 for DISABLED 1 for ENABLED	The PMD_CS primitive, in conjunction with PMD_ED, provides CCA status through the PLCP layer PHYCCA primitive. PMD_CS indicates a binary status of ENABLED or DISABLED. PMD_CS is ENABLED when the correlator SQ indicated in PMD_SQ is greater than the correlation threshold. PMD_CS is DISABLED when the PMD_SQ falls below the correlation threshold.

17.4.5.14.4 When generated

This primitive is generated by the PMD sublayer when the High Rate PHY is receiving a PPDU and the PN code has been acquired.

17.4.5.14.5 Effect of receipt

This indicator is provided to the PLCP for forwarding to the MAC entity for information purposes through the PHY-CCA indicator. This parameter shall indicate that the RF medium is busy and occupied by a High Rate PHY signal. The High Rate PHY should not be placed into the transmit state when PMD_CS is ENABLED.

17.4.5.15 PMD_ED.indication

17.4.5.15.1 Function

This optional primitive may be generated by the PMD to provide an indication that the receiver has detected RF energy indicated by the PMD_RSSI primitive that is above a predefined threshold.

17.4.5.15.2 Semantics of the service primitive

This primitive provides the following parameter:

Parameter	Associated primitive	Value	Description
PMD_ED	PMD_ED.indication	0 for DISABLED 1 for ENABLED	The PMD_ED primitive, along with the PMD_SQ, provides CCA status at the PLCP layer through the PHY-CCA primitive. PMD_ED indicates a binary status of ENABLED or DISABLED. PMD_ED is ENABLED when the RSSI in PMD_RSSI is greater than the ED_THRESHOLD parameter. PMD_ED is DISABLED when the PMD_RSSI falls below the energy detect threshold.

17.4.5.15.3 When generated

This primitive is generated by the PHY when the PHY is receiving RF energy from any source that exceeds the ED_THRESHOLD parameter.

17.4.5.15.4 Effect of receipt

This indicator is provided to the PLCP for forwarding to the MAC entity for information purposes through the PMD_ED indicator. This parameter shall indicate that the RF medium may be busy with an RF energy source that is not High Rate PHY compliant. If a High Rate PHY source is being received, the PMD_CS function is enabled shortly after the PMD_ED function is enabled.

17.4.5.16 PMD_ED.request

17.4.5.16.1 Function

This optional primitive may be generated by the PLCP to set a value for the energy detect ED_THRESHOLD.

17.4.5.16.2 Semantics of the service primitive

This primitive provides the following parameter:

Parameter	Associated primitive	Value	Description
PMD_ED	PMD_ED.request	ED_THRESHOLD	ED_THRESHOLD is the threshold that the RSSI should be greater than in order for PMD_ED to be enabled. PMD_ED is DISABLED when the PMD_RSSI falls below the energy detect threshold.

17.4.5.16.3 When generated

This primitive is generated by the PLCP sublayer to change or set the current High Rate PHY energy detect threshold.

17.4.5.16.4 Effect of receipt

The receipt of PMD_ED immediately changes the energy detect threshold as set by the ED_THRESHOLD parameter.

17.4.5.17 PMD_RCPI.indication

17.4.5.17.1 Function

This optional primitive, generated by the PMD sublayer, provides the RCPI to the PLCP and MAC.

17.4.5.17.2 Semantics of the service primitive

The primitive shall provide the following parameter:

Parameter	Associated primitive	Value	Description
RCPI	PMD_RCPI.indication	0–255	The RCPI is a measure of the received power by the High Rate PHY as defined in 17.4.8.6.

17.4.5.17.3 When generated

This primitive shall be generated by the PMD when the High Rate PHY is in the receive state when dot11RadioMeasurementActivated is true. It is continuously available to the PLCP, which in turn provides the parameter to the MAC entity.

17.4.5.17.4 Effect of receipt

This parameter shall be provided to the PLCP layer for information only. The RCPI may be used in conjunction with RSNI to measure input signal quality.

17.4.6 PMD operating specifications, general

17.4.6.1 General

General specifications for the High Rate PMD sublayer are provided in 17.4.6.2 to 17.4.6.14. These specifications apply to both the receive and transmit functions and general operation of a High Rate PHY.

WLANs implemented in accordance with this standard are subject to equipment certification and operation requirements established by regional and national regulatory administrations. The PMD specification establishes minimum technical requirements for interoperability, based upon established regulations at the time this standard was issued. These regulations are subject to revision, or may be superseded. Requirements that are subject to local geographic regulations are annotated within the PMD specification. Regulatory requirements that do not affect interoperability are not addressed in this standard. Implementers are referred to the following regulatory sources for further information. Operation in countries within defined regulatory domains may be subject to additional regulations.

17.4.6.2 Operating frequency range

The High Rate PHY shall operate in the 2.4–2.4835 GHz frequency range, as allocated by regulatory bodies in the China, United States, Europe, and Japan, or in the 2.471–2.497 GHz frequency range, as allocated by regulatory authority in Japan.

17.4.6.3 Channel Numbering of operating channels

When dot11OperatingClassesRequired is true, channel numbering shall be as specified in 18.3.8.4.2 and channelization shall be as specified in 18.3.8.4.3. When dot11OperatingClassesDefined is false or not defined, the channel center frequencies and CHNL_ID numbers shall be as shown in Table 17-9. See the applicable regulations for the countries in which the implementation operates.

Table 17-9—High Rate PHY frequency channel plan

CHNL_ID	Frequency (MHz)
1	2412
2	2417
3	2422
4	2427
5	2432
6	2437
7	2442
8	2447
9	2452
10	2457
11	2462
12	2467
13	2472
14	2484

In a multiple cell network topology, overlapping and/or adjacent cells using different channels can operate simultaneously without interference if the distance between the center frequencies is at least 25 MHz. Channel 14 shall be designated specifically for operation in Japan.

17.4.6.4 Modulation and channel data rates

Four modulation formats and data rates are specified for the High Rate PHY. The basic access rate shall be based on 1 Mb/s DBPSK modulation. The enhanced access rate shall be based on 2 Mb/s DQPSK. The extended direct sequence specification defines two additional data rates. The High Rate access rates shall be based on the CCK modulation scheme for 5.5 Mb/s and 11 Mb/s. An optional PBCC mode is also provided for potentially enhanced performance.

17.4.6.5 Spreading sequence and modulation for 1 Mb/s and 2 Mb/s

The following 11-chip Barker sequence shall be used as the PN code sequence for the 1 Mb/s and 2 Mb/s modulation:

+1, –1, +1, +1, –1, +1, +1, +1, –1, –1, –1

The leftmost chip shall be output first in time. The first chip shall be aligned at the start of a transmitted symbol. The symbol duration shall be exactly 11 chips long.

The DBPSK encoder for the basic access rate is specified in Table 17-10. The DQPSK encoder is specified in Table 17-11. (In these tables, +jω shall be defined as counterclockwise rotation.)

Table 17-10—1 Mb/s DBPSK encoding table

Bit input	Phase change (+jω)
0	0
1	π

Table 17-11—2 Mb/s DQPSK encoding table

Dibit pattern (d0,d1) (d0 is first in time)	Phase change (+jω)
00	0
01	π/2
11	π
10	3π/2 (–π/2)

17.4.6.6 Spreading sequences and modulation for CCK modulation at 5.5 Mb/s and 11 Mb/s

17.4.6.6.1 General

For the CCK modulation modes, the spreading code length is 8 and is based on complementary codes. The chipping rate is 11 Mchip/s. The symbol duration shall be exactly 8 complex chips long.

The following formula shall be used to derive the CCK codewords that shall be used for spreading both 5.5 Mb/s and 11 Mb/s

$$\begin{aligned} c = \{ & e^{j(\varphi_1 + \varphi_2 + \varphi_3 + \varphi_4)}, e^{j(\varphi_1 + \varphi_3 + \varphi_4)}, e^{j(\varphi_1 + \varphi_2 + \varphi_4)}, \\ & -e^{j(\varphi_1 + \varphi_4)}, e^{j(\varphi_1 + \varphi_2 + \varphi_3)}, e^{j(\varphi_1 + \varphi_3)}, -e^{j(\varphi_1 + \varphi_2)}, e^{j\varphi_1} \} \end{aligned} \tag{17-1}$$

where C is the codeword

C = {c0 to c7}

The terms φ1, φ2, φ3, and φ4 are defined in 17.4.6.6.3 for 5.5 Mb/s and 17.4.6.6.4 for 11 Mb/s.

This formula creates 8 complex chips (c0 to c7), where c0 is transmitted first in time.

This is a form of the generalized Hadamard transform encoding, where φ1 is added to all code chips, φ2 is added to all odd code chips, φ3 is added to all odd pairs of code chips, and φ4 is added to all odd groups of four code chips.

The term φ1 modifies the phase of all code chips of the sequence and shall be DQPSK encoded for 5.5 Mb/s and 11 Mb/s. This shall take the form of rotating the whole symbol by the appropriate amount relative to the phase of the preceding symbol. Note that the chip c7 of the symbol defined above is the chip that indicates the symbol's phase and is transmitted last.

17.4.6.6.2 Cover code for CCK

The fourth and seventh chips are rotated 180° by a cover sequence to optimize the sequence correlation properties and minimize dc offsets in the codes. This explains the minus sign on the fourth and seventh terms in Equation (17-1).

17.4.6.6.3 CCK 5.5 Mb/s modulation

At 5.5 Mb/s, 4 bits (d0 to d3; d0 first in time) are transmitted per symbol.

The data bits d0 and d1 encode φ1 based on DQPSK. The DQPSK encoder is specified in Table 17-11. (In the table, +jω shall be defined as counterclockwise rotation.) The phase change for φ1 is relative to the phase φ1 of the preceding symbol. For the header to PSDU transition, the phase change for φ1 is relative to the phase of the preceding DQPSK (2 Mb/s) symbol. That is, the phase of the last symbol of the CRC-16 is the reference phase for the first symbol generated from the PSDU octets. (See the definition in 17.4.6.5 for the reference phase of this Barker coded symbol.) A "+1" chip in the Barker code shall represent the same carrier phase as a "+1" chip in the CCK code.

All odd-numbered symbols generated from the PSDU octets shall be given an extra 180 degree (π) rotation, in addition to the standard DQPSK modulation as shown in Table 17-12. The symbols of the PSDU shall be numbered starting with 0 for the first symbol, for the purposes of determining odd and even symbols. That is, the PSDU transmission starts on an even-numbered symbol.

Table 17-12—DQPSK encoding table

Dibit pattern (d0, d1) (d0 is first in time)	Even symbols phase change (+jω)	Odd symbols phase change (+jω)
00	0	π
01	π/2	3π/2 (–π/2)
11	π	0
10	3π/2 (–π/2)	π/2

The data dibits d2 and d3 CCK encode the basic symbol, as specified in Table 17-13. This table is derived from the formula above by setting φ2 = (d2 × π) + π /2, φ3 = 0, and φ4 = d3 × π. In this table, d2 and d3 are in the order shown, and the complex chips are shown c0 to c7 (left to right), with c0 transmitted first in time.

Table 17-13—5.5 Mb/s CCK encoding table

d2, d3	c1	c2	c3	c4	c5	c6	c7	c8
00	1j	1	1j	–1	1j	1	–1j	1
01	–1j	–1	–1j	1	1j	1	–1j	1
10	–1j	1	–1j	–1	–1j	1	1j	1
11	1j	–1	1j	1	–1j	1	1j	1

17.4.6.6.4 CCK 11 Mb/s modulation

At 11 Mb/s, 8 bits (d0 to d7; d0 first in time) are transmitted per symbol.

The first dibit (d0, d1) encodes φ1 based on DQPSK. The DQPSK encoder is specified in Table 17-11. The phase change for φ1 is relative to the phase φ1 of the preceding symbol. In the case of header to PSDU transition, the phase change for φ1 is relative to the phase of the preceding DQPSK symbol. All odd-numbered symbols of the PSDU are given an extra 180 degree (π) rotation, in accordance with the DQPSK modulation shown in Table 17-11. Symbol numbering starts with 0 for the first symbol of the PSDU.

The data dibits (d2, d3), (d4, d5), and (d6, d7) encode φ2, φ3, and φ4, respectively, based on QPSK as specified in Table 17-14. Note that this table is binary (not Gray) coded.

Table 17-14—QPSK encoding table

Dibit pattern [di, d(i+1)] (di is first in time)	Phase
00	0
01	π/2
10	π
11	3π/2 (–π/2)

17.4.6.7 DSSS/PBCC data modulation and modulation rate (optional)

This optional coding scheme uses a binary convolutional coding with a 64-state binary convolutional code (BCC) and a cover sequence. The output of the BCC is encoded jointly onto the I and Q channels, as described in this subclause.

The encoder for this scheme is shown in Figure 17-11. Incoming data are first encoded with a binary convolutional code. A cover code is applied to the encoded data prior to transmission through the channel.

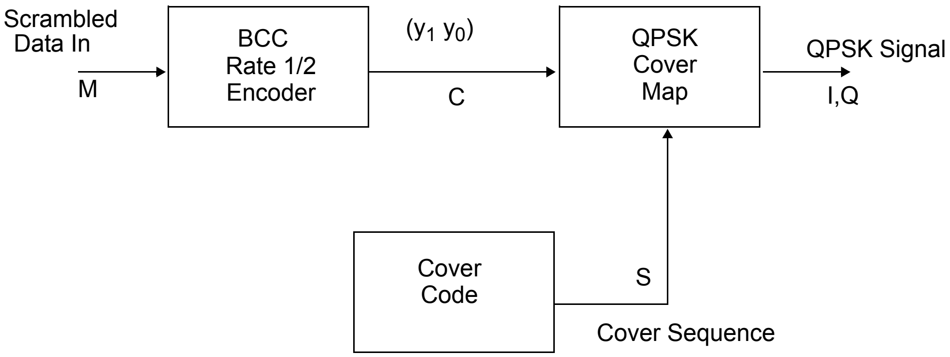


Figure 17-11—PBCC modulator scheme

The BCC that is used is a 64-state, rate ½ code. The generator matrix for the code is given as

$$G = [D^6 + D^4 + D^3 + D + 1, \quad D^6 + D^5 + D^4 + D^3 + D^2 + 1]$$

or in octal notation, it is given by

$$G = [133, \quad 175]$$

Because the system is frame (PPDU) based, the encoder shall be in state zero (i.e., all memory elements contain 0 at the beginning of each PPDU). The encoder is also be placed in a known state at the end of each PPDU to prevent the data bits near the end of the PPDU from being substantially less reliable than those early on in the PPDU. To place the encoder in a known state at the end of a PPDU, at least six deterministic bits shall be input immediately following the last data bit input to the convolutional encoder. This is achieved by appending 1 octet containing all zeros to the end of the PPDU prior to transmission, and discarding the final octet of each received PPDU. In this manner, the decoding process can be completed reliably on the last data bits.

An encoder block diagram is shown in Figure 17-12. It consists of six memory elements. For every data bit input, two output bits are generated.

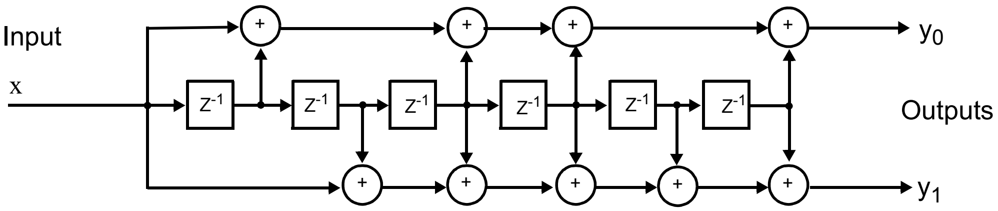


Figure 17-12—PBCC convolutional encoder

The output of the binary convolutional code described above is mapped to a constellation using one of two possible rates. The 5.5 Mb/s rate uses BPSK, and the 11 Mb/s rate uses QPSK. In QPSK mode, each pair of output bits from the binary convolutional code is used to produce one symbol; in BPSK mode, each pair of bits from the BCC is taken serially (y_0 first) and used to produce two BPSK symbols. This yields a throughput of one bit per symbol in QPSK mode and one-half a bit per symbol in BPSK mode.

The phase of the first complex chip of the PSDU shall be defined with respect to the phase of the last chip of the PCLP header (i.e., the last chip of the CRC check). The bits (y_1 y_0) = (0, 0) shall indicate the same phase as the last chip of the CRC check. The other three combinations of (y_1 y_0) shall be defined with respect to this reference phase, as shown in Figure 17-13.

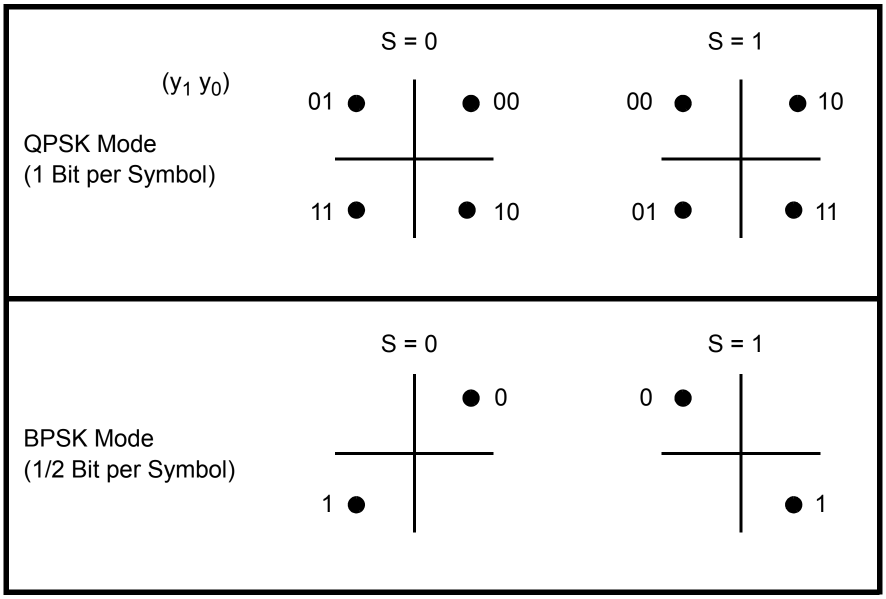


Figure 17-13—Cover code mapping

The mapping from BCC outputs to PSK constellation points in BPSK and QPSK modes is determined by a pseudorandom cover sequence. This is shown for both modes in Figure 17-13. Note that this is an absolute phase table, not differential as in CCK.

The pseudorandom cover sequence is generated from a seed sequence. The 16-bit seed sequence is 0011001110001011, where the first bit of the sequence in time is the leftmost bit. This sequence in octal notation is given as 150714, where the LSB is the first in time. This seed sequence is used to generate the 256-bit pseudorandom cover sequence, which is used in the mapping of the current PSK symbol. It is the current binary value of this sequence at every given point in time that is taken as S in Figure 17-13.

This sequence of 256 bits is produced by taking the first sixteen bits of the sequence as the seed sequence, the second sixteen bits as the seed sequence cyclically left rotated by three, the third sixteen bits as the seed sequence cyclically left rotated by six, etc. If c*i* is the *i*th bit of the seed sequence, where 0 ≤ I ≤ 15, then the sequence that is used to cover the data is given row-wise as follows:

c0 c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15
c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c0 c1 c2
c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c0 c1 c2 c3 c4 c5
c9 c10 c11 c12 c13 c14 c15 c0 c1 c2 c3 c4 c5 c6 c7 c8
c12 c13 c14 c15 c0 c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11
c15 c0 c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14
c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c0 c1
c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c0 c1 c2 c3 c4
c8 c9 c10 c11 c12 c13 c14 c15 c0 c1 c2 c3 c4 c5 c6 c7
c11 c12 c13 c14 c15 c0 c1 c2 c3 c4 c5 c6 c7 c8 c9 c10
c14 c15 c0 c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13

c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c0
c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c0 c1 c2 c3
c7 c8 c9 c10 c11 c12 c13 c14 c15 c0 c1 c2 c3 c4 c5 c6
c10 c11 c12c13 c14 c15 c0 c1 c2 c3 c4 c5 c6 c7 c8 c9
c13 c14 c15 c0 c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12

For PPDUs with more than 256 data bits, this sequence of 256 bits is simply repeated.

17.4.6.8 Channel Agility (optional)

17.4.6.8.1 General

This Channel Agility option allows an implementation to overcome some inherent difficulty with static channel assignments (a tone jammer), without burdening all implementations with the added cost of this capability. When the Channel Agility option is enabled, the PHY shall meet the requirements on channel switching and settling time, as described in 17.4.6.13, and the hop sequences described below. This option can also be used to implement IEEE 802.11-compliant systems that are interoperable between both FH and DS modulations. Annex K contains a description of the expected behavior when such networks are employed.

17.4.6.8.2 Hop sequences

The hop sequences for each of the specified geographical areas are defined with two sets. High Rate frequency channels referred to in this subclause are defined in Table 17-9.

The first set (Figure 17-14 and Figure 17-16) uses nonoverlapping frequency channels to allow the High Rate systems to minimize interference degradation. The synchronization of FH is performed by the MLME, as defined in 10.1.6 for the FH PHY. The PLME SAP service primitives used to command a new frequency channel are defined in 6.5.

The second set (Figure 17-15 and Figure 17-17) uses half overlapping frequency channels, with 10 MHz center frequency spacing, to enable interoperability with 1 Mb/s and 2 Mb/s FH systems hopping with the approved IEEE 802.11 hop sequences. The High Rate hop frequency is calculated from the specific 1 MHz channel chosen for a given hop by picking the closest High Rate channel within the set. Where there is a choice of two DSSS channels, the lower one shall be the one chosen. Therefore, the chosen channel shall be no more than ±5 MHz of the channel center of the FH channel. When operating on the FH channels beyond ±5 MHz of the closest High Rate channel specified in the set, the High Rate mode shall not be used and all FH transmissions shall occur at the 1 Mb/s or 2 Mb/s rate.

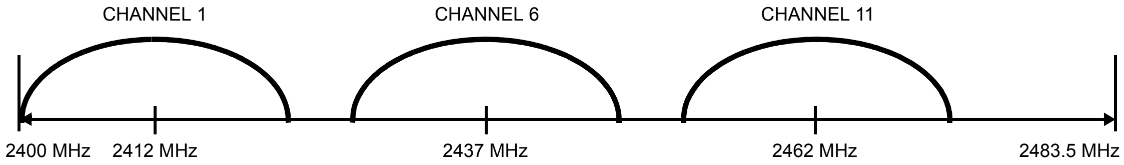


Figure 17-14—China and North American channel selection—nonoverlapping

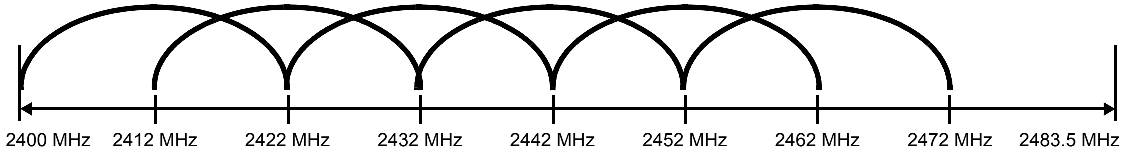


Figure 17-15—China and North American channel selection—overlapping

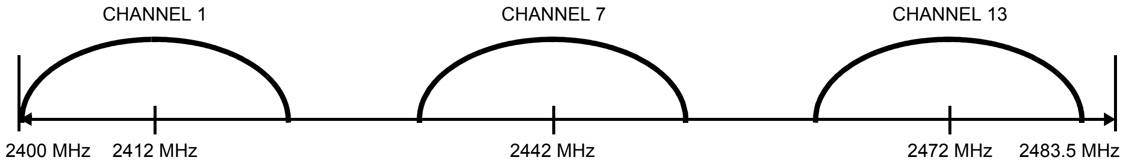


Figure 17-16—European channel selection—nonoverlapping

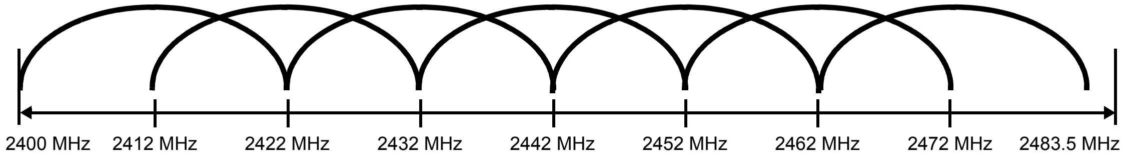


Figure 17-17—European channel selection—overlapping

17.4.6.8.3 Operating channels

The operating channels for specified geographical areas are defined in Table 17-15 and Table 17-16.

Table 17-15—China and North American operating channels

Set	Number of channels	HR/DSSS channel numbers
1	3	1, 6, 11
2	6	1, 3, 5, 7, 9, 11

Table 17-16—European operating channels (except France and Spain)

Set	Number of channels	HR/DSS channel numbers
1	3	1, 7, 13
2	7	1, 3, 5, 7, 9, 11, 13

17.4.6.8.4 Hop patterns

An FH pattern, Fx, consists of a permutation of all frequency channels defined in Table 17-15 and Table 17-16. For a given pattern number *x*, the hopping sequence is as follows:

$Fx = \{fx\ (1), fx\ (2), ..., fx\ (p)\}$

where

$fx\ (i)$ is the channel number (as defined in 14.7.4) for i^{th} frequency in the x^{th} hopping pattern

p is the number of hops in pseudorandom hopping pattern before repeating sequence (79 for North America and most of Europe)

The FH patterns for Set 1 of each geographic area are based on the hop patterns in Table 17-17 and Table 17-18.

Table 17-17—China and North American Set 1 hop patterns

Index	Pattern 1	Pattern 2
1	1	1
2	6	11
3	11	6

Table 17-18—European Set 1 hop patterns (except France and Spain)

Index	Pattern 1	Pattern 2
1	1	1
2	7	13
3	13	7

The FH patterns for Set 2 of each geographic area are defined by the 1/2 Mb/s FH PHY hop sequences, as described in the FH PHY (14.7.8). Given the hopping pattern number *x*, and the index for the next frequency, *i* (in the range 1 to p), the DS channel number (as defined in 17.4.6.3) shall be selected with the following algorithm:

China and North America

$f'x\ (i) = f'x\ (i)$ for $1 \leq f'x\ (i) \leq 11$;

$f'x\ (i) =$ null for $f'x\ (i) < 1$ and $f'x\ (i) > 11$;

$f'x\ (i) = 2 \times \mathrm{Int}\ [(\{[b(i) + x] \bmod (79) + 2\} - 6) / 10] - 1$;

with $b(i)$ defined in Table 14-11 (in 14.7.8).

Most of Europe

$f'x\ (i) = f'x\ (i)$ for $1 \leq f'x\ (i) \leq 13$;

$f'x\ (i) =$ null for $f'x\ (i) < 1$ and $f'x\ (i) > 13$;

$f'x\ (i) = 2 \times \mathrm{Int}\ [(\{[b(i) + x] \bmod (79) + 2\} - 6) / 10] - 1$;

with $b(i)$ defined in Table 14-11 (in 14.7.8).

17.4.6.9 Transmit and receive in-band and out-of-band spurious emissions

The High Rate PHY conforms with in-band and out-of-band spurious emissions as set by the appropriate regulatory bodies.

17.4.6.10 TX-to-RX turnaround time

The TX-to-RX turnaround time shall be less than 10 µs, including the power-down ramp specified in 17.4.7.7.

The TX-to-RX turnaround time shall be measured at the air interface from the trailing edge of the last transmitted symbol to the valid CCA detection of the incoming signal. The CCA should occur within 25 µs (10 µs for turnaround time, plus 15 µs for energy detect), or by the next slot boundary occurring after the 25 µs has elapsed (see 17.4.8.5). A receiver input signal 3 dB above the ED threshold described in 17.4.8.5 shall be present at the receiver.

17.4.6.11 RX-to-TX turnaround time

The RX-to-TX turnaround time shall be measured at the MAC/PHY interface using the PHY-TXSTART.request primitive, and shall be 5 µs. This includes the transmit power-on ramp described in 17.4.7.7.

17.4.6.12 Slot time

The slot time for the High Rate PHY shall be the sum of the RX-to-TX turnaround time (5 µs) and the energy detect time (15 µs specified in 17.4.8.5). The propagation delay shall be regarded as being included in the energy detect time.

17.4.6.13 Channel switching/settling time

When the Channel Agility option is enabled, the time to change from one operating channel frequency to another, as specified in 17.4.6.3, is 224 µs. A conformant PMD meets this switching time specification when the operating channel center frequency has settled to within ±60 kHz of the nominal channel center. STAs shall not transmit until after the channel change settling time.

17.4.6.14 Transmit and receive antenna port impedance

The impedance of the transmit and receive antenna port(s) shall be 50 Ω if the port is exposed.

17.4.7 PMD transmit specifications

17.4.7.1 Introduction

The transmit functions and parameters associated with the PMD sublayer are described in 17.4.7.2 to 17.4.7.9.

17.4.7.2 Transmit power levels

The maximum allowable output power is measured in accordance with practices specified by the appropriate regulatory bodies.

17.4.7.3 Transmit power level control

Power control shall be provided for transmitted power greater than 100 mW. A maximum of four power levels may be provided. As a minimum, a radio capable of transmission greater than 100 mW shall be capable of switching power back to 100 mW or less.

17.4.7.4 Transmit spectrum mask

The transmitted spectral products shall be less than –30 dBr (decibel relative to the SINx/x peak) for
f_c – 22 MHz < f < f_c –11 MHz; and
f_c + 11 MHz < f < f_c + 22 MHz;

and shall be less than –50 dBr for
f < f_c – 22 MHz; and
f > f_c + 22 MHz.

where
f_c is the channel center frequency

The transmit spectral mask is shown in Figure 17-18. The measurements shall be made using a 100 kHz resolution bandwidth and a 100 kHz video bandwidth.

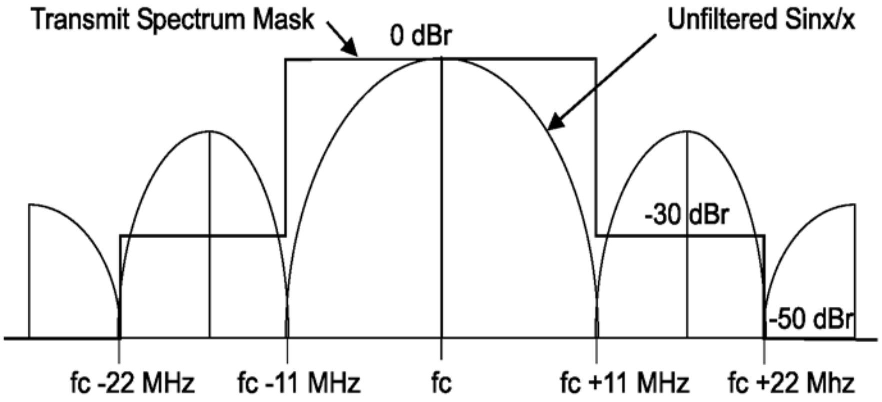


Figure 17-18—Transmit spectrum mask

17.4.7.5 Transmit center frequency tolerance

The transmitted center frequency tolerance shall be ±25 ppm maximum.

17.4.7.6 Chip clock frequency tolerance

The PN code chip clock frequency tolerance shall be better than ±25 ppm maximum. It is highly recommended that the chip clock and the transmit frequency be locked (coupled) for optimum demodulation performance. If these clocks are locked, it is recommended that bit 2 of the SERVICE field be set to 1, as indicated in 17.2.3.5.

17.4.7.7 Transmit power-on and power-down ramp

The transmit power-on ramp for 10% to 90% of maximum power shall be no greater than 2 µs. The transmit power-on ramp is shown in Figure 17-19.

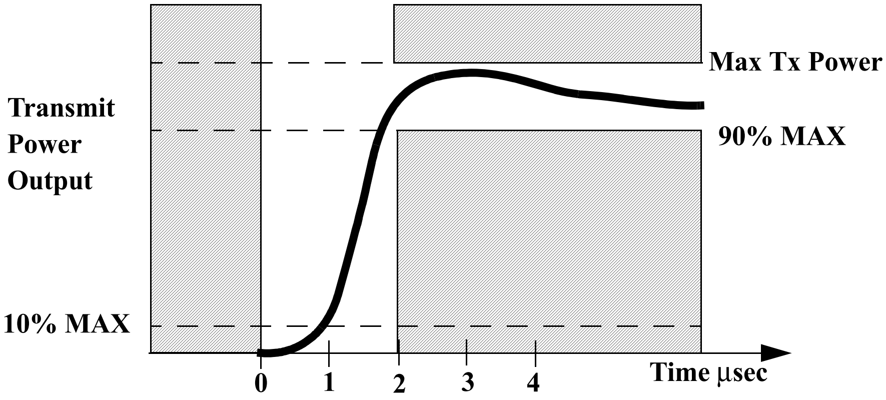


Figure 17-19—Transmit power-on ramp

The transmit power-down ramp for 90% to 10% maximum power shall be no greater than 2 µs. The transmit power-down ramp is shown in Figure 17-20.

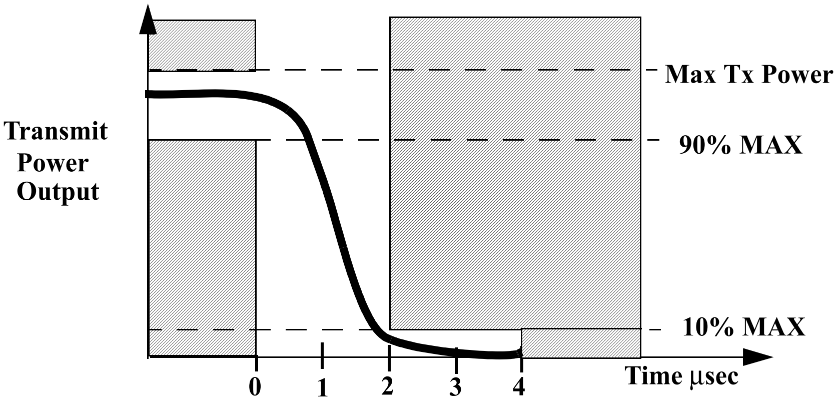


Figure 17-20—Transmit power-down ramp

The transmit power ramps shall be constructed such that the High Rate PHY emissions conform with spurious frequency product specification defined in 17.4.6.9.

17.4.7.8 RF carrier suppression

The RF carrier suppression, measured at the channel center frequency, shall be at least 15 dB below the peak SIN(x)/x power spectrum. The RF carrier suppression shall be measured while transmitting a repetitive 01 data sequence with the scrambler disabled using DQPSK modulation. A 100 kHz resolution bandwidth shall be used to perform this measurement.

17.4.7.9 Transmit modulation accuracy

The transmit modulation accuracy requirement for the High Rate PHY shall be based on the difference between the actual transmitted waveform and the ideal signal waveform. Modulation accuracy shall be determined by measuring the peak vector error magnitude during each chip period. Worst-case vector error magnitude shall not exceeded 0.35 for the normalized sampled chip data. The ideal complex I and Q constellation points associated with DQPSK modulation, (0.707, 0.707), (0.707, –0.707), (–0.707, 0.707), (–0.707, –0.707), shall be used as the reference. These measurements shall be from baseband I and Q sampled data after recovery through a reference receiver system.

Figure 17-21 illustrates the ideal DQPSK constellation points and range of worst-case error specified for modulation accuracy.

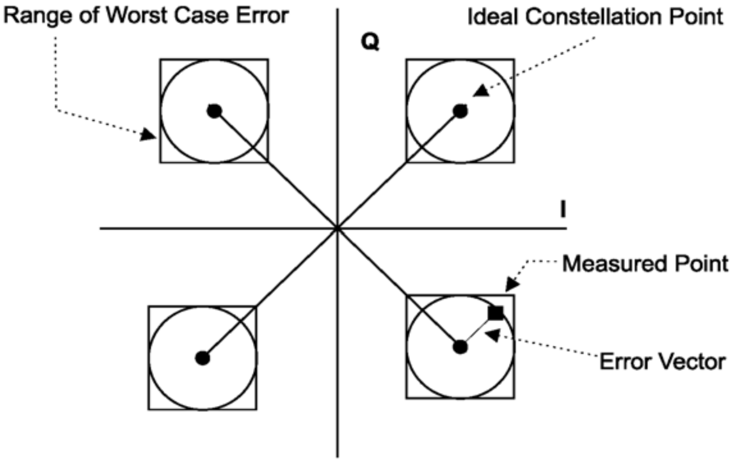


Figure 17-21—Modulation accuracy measurement example

Error vector measurement requires a reference receiver capable of carrier lock. All measurements shall be made under carrier lock conditions. The distortion induced in the constellation by the reference receiver shall be calibrated and measured. The test data error vectors described below shall be corrected to compensate for the reference receiver distortion.

The IEEE 802.11-compatible radio shall provide an exposed TX chip clock, which shall be used to sample the I and Q outputs of the reference receiver.

The measurement shall be made under the conditions of continuous DQPSK transmission using scrambled all ones.

The eye pattern of the I channel shall be used to determine the I and Q sampling point. The chip clock provided by the vendor radio shall be time delayed, such that the samples fall at a 1/2 chip period offset from the mean of the zero crossing positions of the eye (see Figure 17-22). This is the ideal center of the eye and may not be the point of maximum eye opening.

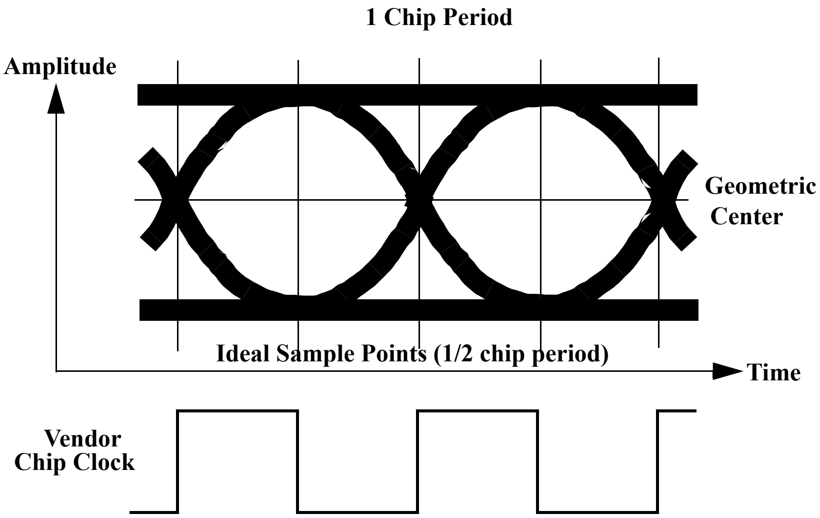


Figure 17-22—Chip clock alignment with baseband eye pattern

Using the aligned chip clock, 1000 samples of the I and Q baseband outputs from the reference receiver are captured. The vector error magnitudes shall be calculated as follows:

Calculate the dc offsets for I and Q samples

$$I_{\text{mean}} = \sum_{n=0}^{999} I(n)/1000$$

$$Q_{\text{mean}} = \sum_{n=0}^{999} Q(n)/1000$$

Calculate the dc corrected I and Q samples for all n = 1000 sample pairs

$\mathrm{I_{dc}(n) = I(n)} - I_{\text{mean}}$

$\mathrm{Q_{dc}(n) = Q(n)} - Q_{\text{mean}}$

Calculate the average magnitude of I and Q samples

$$I_{\text{mag}} = \sum_{n=0}^{999} |I_{\text{dc}}(n)|/1000$$

$$Q_{\text{mag}} = \sum_{n=0}^{999} |Q_{\text{dc}}(n)|/1000$$

Calculate the normalized error vector magnitude for the $\mathrm{I_{dc}(n)/Q_{dc}(n)}$ pairs

$$V_{\mathrm{err}}(n) = [\{|I_{\mathrm{dc}}(n)|/I_{\mathrm{mag}} - 1\}^2 + \{|Q_{\mathrm{dc}}(n)|/Q_{\mathrm{mag}} - 1\}^2]^{\frac{1}{2}} - V_{\mathrm{correction}}$$

where

$V_{\mathrm{correction}}$ is the error induced by the reference receiver system

A vendor High Rate PHY implementation shall be compliant if for all n = 1000 samples, the following condition is met:

$V_{\mathrm{err}}(n) < 0.35$

17.4.7.10 Time of Departure accuracy

The Time of Departure accuracy test evaluates TIME_OF_DEPARTURE against aTxPmdTxStartRMS and aTxPmdTxStartRMS against TIME_OF_DEPARTURE_ACCURACY_TEST_THRESH as defined in Annex T with the following test parameters:

— MULTICHANNEL_SAMPLING_RATE is $22 \times 10^6\left(1 + \left\lceil \frac{f_H - f_L}{22\ \mathrm{MHz}} \right\rceil\right)$ sample/s

where

f_H is the nominal center frequency in Hz of the highest channel in the channel set

f_L is the nominal center frequency in Hz of the lowest channel in the channel set, the channel set is the set of channels upon which frames providing measurements are transmitted, the channel set comprises channels uniformly spaced across $f_H - f_L \geq 50$ MHz

$\lceil x \rceil$ equals the smallest integer equal to or larger than x

— FIRST_TRANSITION_FIELD is the SYNC field.

— SECOND_TRANSITION_FIELD is the SFD field.

— TRAINING_FIELD is the concatenation of the appropriate short or long SYNC and SFD fields, using a chip pulse which should approximate a rectangular pulse of duration 1/ 11 MHz convolved with a brick-wall low pass filter of bandwidth 11 MHz.

— TIME_OF_DEPARTURE_ACCURACY_TEST_THRESH is 80 ns.

NOTE—The indicated chip pulse applies to the time of departure accuracy test equipment, and not the transmitter or receiver.

17.4.8 PMD receiver specifications

17.4.8.1 Introduction

The receive functions and parameters associated with the PMD sublayer are described in 17.4.8.2 to 17.4.8.5.

17.4.8.2 Receiver minimum input level sensitivity

The FER shall be less than 8×10^{-2} at a PSDU length of 1024 octets for an input level of –76 dBm measured at the antenna connector. This FER shall be specified for 11 Mb/s CCK modulation. The test for the minimum input level sensitivity shall be conducted with the ED threshold set less than or equal to –76 dBm.

17.4.8.3 Receiver maximum input level

The receiver shall provide a maximum FER of 8×10^{-2} at a PSDU length of 1024 octets for a maximum input level of –10 dBm measured at the antenna. This FER shall be specified for 11 Mb/s CCK modulation.

17.4.8.4 Receiver adjacent channel rejection

Adjacent channel rejection is defined between any two channels with ≥ 25 MHz separation in each channel group, as defined in 17.4.6.3.

The adjacent channel rejection shall be equal to or better than 35 dB, with an FER of 8×10^{-2} using 11 Mbit/s CCK modulation described in 17.4.6.4 and a PSDU length of 1024 octets.

The adjacent channel rejection shall be measured using the following method.

Input an 11 Mb/s CCK modulated signal at a level 6 dB greater than specified in 17.4.8.2. In an adjacent channel (≥ 25 MHz separation as defined by the channel numbering), input a signal modulated in a similar fashion, which adheres to the transmit mask specified in 17.4.7.4, to a level 41 dB above the level specified in 17.4.8.2. The adjacent channel signal shall be derived from a separate signal source. It shall not be a frequency shifted version of the reference channel. Under these conditions, the FER shall be no worse than 8×10^{-2}.

17.4.8.5 CCA

The High Rate PHY shall provide the capability to perform CCA according to at least one of the following three methods:

- CCA Mode 1: Energy above threshold. CCA shall report a busy medium upon detecting any energy above the ED threshold.
- CCA Mode 4: CS with timer. CCA shall start a timer whose duration is 3.65 ms and report a busy medium only upon the detection of a High Rate PHY signal. CCA shall report an IDLE medium after the timer expires and no High Rate PHY signal is detected. The 3.65 ms timeout is the duration of the longest possible 5.5 Mb/s PSDU.
- CCA Mode 5: A combination of CS and energy above threshold. CCA shall report busy at least while a High Rate PPDU with energy above the ED threshold is being received at the antenna.

The ED status shall be given by the PMD primitive, PMD_ED. The CS status shall be given by PMD_CS. The status of PMD_ED and PMD_CS is used in the PLCP to indicate activity to the MAC through the PHY-CCA.indication primitive.

A busy channel shall be indicated by PHY-CCA.indication primitive of class BUSY. A clear channel shall be indicated by PHY-CCA.indication primitive of class IDLE.

dot11CCAModeSupported shall indicate the appropriate operation modes. The PHY shall be configured through dot11CurrentCCAMode.

The CCA shall indicate true if there is no energy detect or CS. The CCA parameters are subject to the following criteria:

a) If a valid High Rate signal is detected during its preamble within the CCA window, the ED threshold shall be less than or equal to –76 dBm for TX power > 100 mW; –73 dBm for 50 mW < TX power ≤ 100 mW; and –70 dBm for TX power ≤ 50 mW.
b) With a valid signal (according to the CCA mode of operation) present at the receiver antenna within 5 µs of the start of a MAC slot boundary, the CCA indicator shall report channel busy before the end of the slot time. This implies that the CCA signal is available as an exposed test point. Refer to Figure 9-14 (in 9.3.7) for a slot time boundary definition.
c) In the event that a correct PLCP header is received, the High Rate PHY shall hold the CCA signal inactive (channel busy) for the full duration, as indicated by the PLCP LENGTH field. Should a loss of CS occur in the middle of reception, the CCA shall indicate a busy medium for the intended

duration of the transmitted PPDU. Upon reception of a correct PLCP header, the timer of CCA Mode 2 shall be overridden by this requirement.

Conformance to the High Rate PHY CCA shall be demonstrated by applying an equivalent High-Rate-compliant signal above the appropriate ED threshold (item a) so that all conditions described in item b and item c are demonstrated.

17.4.8.6 Received Channel Power Indicator Measurement

The RCPI indicator is a measure of the received RF power in the selected channel for a received frame. This parameter shall be a measure by the PHY sublayer of the received RF power in the channel measured over the entire received frame or by other equivalent means that meet the specified accuracy. RCPI shall be a monotonically increasing, logarithmic function of the received power level defined in dBm. The allowed values for the RCPI parameter shall be an 8-bit value in the range from 0 to 220, with indicated values rounded to the nearest 0.5 dB as follows:

0: Power ≤ – 110 dBm

1: Power = – 109.5 dBm

2: Power = – 109.0 dBm

and so on where

RCPI = Int{(Power in dBm +110) × 2} for 0 dbm > Power > – 110 dBm

220: Power ≥ – 0 dBm

221–254: Reserved

255: Measurement not available

RCPI shall equal the received RF power within an accuracy of ±5 dB (95% confidence interval) within the specified dynamic range of the receiver. The received RF power shall be determined assuming a receiver noise equivalent bandwidth equal to the channel bandwidth multiplied by 1.1.

18. Orthogonal frequency division multiplexing (OFDM) PHY specification

18.1 Introduction

18.1.1 General

This clause specifies the PHY entity for an orthogonal frequency division multiplexing (OFDM) system. The OFDM system provides a WLAN with data payload communication capabilities of 6, 9, 12, 18, 24, 36, 48, and 54 Mb/s. The support of transmitting and receiving at data rates of 6, 12, and 24 Mb/s is mandatory. The system uses 52 subcarriers that are modulated using binary or quadrature phase shift keying (BPSK or QPSK) or using 16- or 64-quadrature amplitude modulation (16-QAM or 64-QAM). Forward error correction coding (convolutional coding) is used with a coding rate of 1/2, 2/3, or 3/4.

The OFDM system also provides a "half-clocked" operation using 10 MHz channel spacings with data communications capabilities of 3, 4.5, 6, 9, 12, 18, 24, and 27 Mb/s. The support of transmitting and receiving at data rates of 3, 6, and 12 Mb/s is mandatory when using 10 MHz channel spacing. The half-clocked operation doubles symbol times and clear channel assessment (CCA) times when using 10 MHz channel spacing. The regulatory requirements and information regarding use of this OFDM PHY are in Annex D and Annex E.

The OFDM system also provides a "quarter-clocked" operation using 5 MHz channel spacing with data communication capabilities of 1.5, 2.25, 3, 4.5, 6, 9, 12, and 13.5 Mb/s. The support of transmitting and receiving at data rates of 1.5, 3, and 6 Mb/s is mandatory when using 5 MHz channel spacing. The quarter-clocked operation quadruples symbol times and CCA times when using 5 MHz channel spacing. The regulatory requirements and information regarding use of this OFDM PHY are in Annex D and Annex E.

18.1.2 Scope

Subclause 18.1 describes the PHY services provided to the IEEE 802.11 WLAN MAC by the OFDM PHY. The OFDM PHY consists of two protocol functions, as follows:

a) A PHY convergence function, which adapts the capabilities of the PMD system to the PHY service. This function is supported by the PLCP, which defines a method of mapping the IEEE 802.11 PSDUs into a framing format suitable for sending and receiving user data and management information between two or more STAs using the associated PMD system.

b) A PMD system whose function defines the characteristics and method of transmitting and receiving data through a WM between two or more STAs, each using the OFDM system.

18.1.3 OFDM PHY functions

18.1.3.1 General

The OFDM PHY architecture is depicted in the reference model shown in Figure 4-14 (in 4.9). The OFDM PHY contains three functional entities: the PMD function, the PHY convergence function, and the layer management function. Each of these functions is described in detail in 18.1.3.2 to 18.1.3.5.

The OFDM PHY service is provided to the MAC through the PHY service primitives described in Clause 7.

18.1.3.2 PLCP sublayer

In order to allow the IEEE 802.11 MAC to operate with minimum dependence on the PMD sublayer, a PHY convergence sublayer is defined. This function simplifies the PHY service interface to the IEEE 802.11 MAC services.

18.1.3.3 PMD sublayer

The PMD sublayer provides a means to send and receive data between two or more STAs. This clause is concerned with PHYs using OFDM modulation.

18.1.3.4 PLME

The PLME performs management of the local PHY functions in conjunction with the MLME.

18.1.3.5 Service specification method

The models represented by figures and state diagrams are intended to be illustrations of the functions provided. It is important to distinguish between a model and a real implementation. The models are optimized for simplicity and clarity of presentation; the actual method of implementation is left to the discretion of the IEEE 802.11 OFDM-PHY-compliant developer.

The service of a layer or sublayer is the set of capabilities that it offers to a user in the next higher layer (or sublayer). Abstract services are specified here by describing the service primitives and parameters that characterize each service. This definition is independent of any particular implementation.

18.2 OFDM PHY specific service parameter list

18.2.1 Introduction

The architecture of the IEEE 802.11 MAC is intended to be PHY independent. Some PHY implementations require medium management state machines running in the MAC sublayer in order to meet certain PMD requirements. These PHY-dependent MAC state machines reside in a sublayer defined as the MLME. In certain PMD implementations, the MLME may need to interact with the PLME as part of the normal PHY-SAP primitives. These interactions are defined by the PLME parameter list currently defined in the PHY service primitives as TXVECTOR and RXVECTOR. The list of these parameters, and the values they may represent, are defined in the specific PHY specifications for each PMD. Subclause 18.2 addresses the TXVECTOR and RXVECTOR for the OFDM PHY.

18.2.2 TXVECTOR parameters

18.2.2.1 General

The parameters in Table 18-1 are defined as part of the TXVECTOR parameter list in the PHY-TXSTART.request primitive.

18.2.2.2 TXVECTOR LENGTH

The allowed values for the LENGTH parameter are in the range of 1 to 4095. This parameter is used to indicate the number of octets in the MPDU which the MAC is currently requesting the PHY to transmit. This value is used by the PHY to determine the number of octet transfers that will occur between the MAC and the PHY after receiving a request to start the transmission.

18.2.2.3 TXVECTOR DATARATE

The DATARATE parameter describes the bit rate at which the PLCP shall transmit the PSDU. Its value takes any of the rates defined in Table 18-1. Data rates of 6, 12, and 24 Mb/s shall be supported for 20 MHz channel spacing, data rates of 3, 6, and 12 Mb/s shall be supported for 10 MHz channel spacing, and data rates of 1.5, 3, and 6 Mb/s shall be supported for 5 MHz channel spacing; other rates may also be supported.

Table 18-1—TXVECTOR parameters

Parameter	Associated primitive	Value
LENGTH	PHY-TXSTART.request (TXVECTOR)	1–4095
DATATRATE	PHY-TXSTART.request (TXVECTOR)	6, 9, 12, 18, 24, 36, 48, and 54 Mb/s for 20 MHz channel spacing (Support of 6, 12, and 24 Mb/s data rates is mandatory.) 3, 4.5, 6, 9, 12, 18, 24, and 27 Mb/s for 10 MHz channel spacing (Support of 3, 6, and 12 Mb/s data rates is mandatory.) 1.5, 2.25, 3, 4.5, 6, 9, 12, and 13.5 Mb/s for 5 MHz channel spacing (Support of 1.5, 3, and 6 Mb/s data rates is mandatory.)
SERVICE	PHY-TXSTART.request (TXVECTOR)	Scrambler initialization; 7 null bits + 9 reserved null bits
TXPWR_LEVEL	PHY-TXSTART.request (TXVECTOR)	1–8
TIME_OF_ DEPARTURE_ REQUESTED	PHY-TXSTART.request (TXVECTOR)	False, true. When true, the MAC entity requests that the PHY PLCP entity measures and reports time of departure parameters corresponding to the time when the first frame energy is sent by the transmitting port; when false, the MAC entity requests that the PHY PLCP entity neither measures nor reports time of departure parameters.

18.2.2.4 TXVECTOR SERVICE

The SERVICE parameter consists of 7 null bits used for the scrambler initialization and 9 null bits reserved for future use.

18.2.2.5 TXVECTOR TXPWR_LEVEL

The allowed values for the TXPWR_LEVEL parameter are in the range from 1 to 8. This parameter is used to indicate which of the available TxPowerLevel attributes defined in the MIB shall be used for the current transmission.

18.2.2.6 TIME_OF_DEPARTURE_REQUESTED

The allowed values are false or true. A parameter value of true indicates that the MAC sublayer is requesting that the PLCP entity provides measurement of when the first frame energy is sent by the transmitting port and reporting within the PHY-TXSTART.confirm(TXSTATUS) primitive. A parameter value of false indicates that the MAC sublayer is requesting that the PLCP entity not provide time of departure measurement nor reporting in the PHY-TXSTART.confirm(TXSTATUS) primitive.

18.2.3 RXVECTOR parameters

18.2.3.1 General

The parameters listed in Table 18-2 are defined as part of the RXVECTOR parameter list in the PHY-RXSTART.indication primitive.

Table 18-2—RXVECTOR parameters

Parameter	Associated primitive	Value
LENGTH	PHY-RXSTART.indication	1–4095
RSSI	PHY-RXSTART.indication (RXVECTOR)	0–RSSI maximum
DATARATE	PHY-RXSTART.request (RXVECTOR)	6, 9, 12, 18, 24, 36, 48, and 54 Mb/s for 20 MHz channel spacing (Support of 6, 12, and 24 Mb/s data rates is mandatory.) 3, 4.5, 6, 9, 12, 18, 24, and 27 Mb/s for 10 MHz channel spacing (Support of 3, 6, and 12 Mb/s data rates is mandatory.) 1.5, 2.25, 3, 4.5, 6, 9, 12, and 13.5 Mb/s for 5 MHz channel spacing (Support of 1.5, 3, and 6 Mb/s data rates is mandatory.)
SERVICE	PHY-RXSTART.request (RXVECTOR)	Null
RCPI (see NOTE)	PHY-RXSTART.indication (RXVECTOR) PHY-RXEND.indication (RXVECTOR)	0–255
ANT_STATE (see NOTE)	PHY-RXSTART.indication (RXVECTOR) PHY-RXEND.indication (RXVECTOR)	0–255
RX_START_OF_FRAME_OFFSET	PHY-RXSTART.indication (RXVECTOR)	0 to 2^{32}– 1. An estimate of the offset (in 10 ns units) from the point in time at which the start of the preamble corresponding to the incoming frame arrived at the receive antenna port to the point in time at which this primitive is issued to the MAC.
NOTE—Parameter is present only when dot11RadioMeasurementActivated is true.		

18.2.3.2 RXVECTOR LENGTH

The allowed values for the LENGTH parameter are in the range from 1–4095. This parameter is used to indicate the value contained in the LENGTH field which the PLCP has received in the PLCP header. The MAC and PLCP use this value to determine the number of octet transfers that will occur between the two sublayers during the transfer of the received PSDU.

18.2.3.3 RXVECTOR RSSI

The allowed values for the RSSI parameter are in the range from 0 to RSSI maximum. This parameter is a measure by the PHY of the energy observed at the antenna used to receive the current PPDU. RSSI shall be measured during the reception of the PLCP preamble. RSSI is intended to be used in a relative manner, and it shall be a monotonically increasing function of the received power.

18.2.3.4 DATARATE

DATARATE shall represent the data rate at which the current PPDU was received. The allowed values of the DATARATE are 6, 9, 12, 18, 24, 36, 48, or 54 Mb/s for 20 MHz channel spacing; 3, 4.5, 6, 9, 12, 18, 24, or 27 Mb/s for 10 MHz channel spacing; and 1.5, 2.25, 3, 4.5, 6, 9, 12, or 13.5 Mb/s for 5 MHz channel spacing.

18.2.3.5 SERVICE

The SERVICE field shall be null.

18.2.3.6 RXVECTOR RCPI

The allowed values for the RCPI parameter are in the range from 0 to 255, as defined in 18.3.10.7. This parameter is a measure by the PHY of the received channel power. RCPI indications of 8 bits are supported. RCPI shall be measured over the entire received frame or by other equivalent means that meet the specified accuracy.

18.2.4 TXSTATUS parameters

18.2.4.1 General

The parameters listed in Table 18-3 are defined as part of the TXSTATUS parameter list in the PHY-TXSTART.confirm service primitive.

Table 18-3—TXSTATUS parameters

Parameter	Associated primitive	Value
TIME_OF_DEPARTURE	PHY-TXSTART.confirm (TXSTATUS)	0 to 2^{32}– 1. The locally measured time when the first frame energy is sent by the transmitting port, in units equal to 1/ TIME_OF_DEPARTURE_ClockRate. This parameter is present only if TIME_OF_DEPARTURE_REQUESTED is true in the corresponding request.
TIME_OF_DEPARTURE_ClockRate	PHY-TXSTART.confirm (TXSTATUS)	0 to 2^{16}– 1. The clock rate, in units of MHz, is used to generate the TIME_OF_DEPARTURE value. This parameter is present only if TIME_OF_DEPARTURE_REQUESTED is true in the corresponding request.
TX_START_OF_FRAME_OFFSET	PHY-TXSTART.confirm (TXSTATUS)	0 to 2^{32}– 1. An estimate of the offset (in 10 ns units) from the point in time at which the start of the preamble corresponding to the frame was transmitted at the transmit antenna port to the point in time at which this primitive is issued to the MAC.

18.2.4.2 TXSTATUS TIME_OF_DEPARTURE

The allowed values for the TIME_OF_DEPARTURE parameter are integers in the range of 0 to 2^{32}– 1. This parameter is used to indicate when the first frame energy is sent by the transmitting port in units equal to 1/ TIME_OF_DEPARTURE_ClockRate. TIME_OF_DEPARTURE may be included in the transmitted frame in order for recipients on multiple channels to determine the time differences of air propagation times between transmitter and recipients and hence to compute the location of the transmitter.

18.2.4.3 TXSTATUS TIME_OF_DEPARTURE_ClockRate

TIME_OF_DEPARTURE_ClockRate indicates the clock rate used for TIME_OF_DEPARTURE.

18.3 OFDM PLCP sublayer

18.3.1 Introduction

Subclause 18.3 provides a convergence procedure in which PSDUs are converted to and from PPDUs. During transmission, the PSDU shall be provided with a PLCP preamble and header to create the PPDU. At the receiver, the PLCP preamble and header are processed to aid in demodulation and delivery of the PSDU.

18.3.2 PLCP frame format

18.3.2.1 General

Figure 18-1 shows the format for the PPDU including the OFDM PLCP preamble, OFDM PLCP header, PSDU, tail bits, and pad bits. The PLCP header contains the following fields: LENGTH, RATE, a reserved bit, an even parity bit, and the SERVICE field. In terms of modulation, the LENGTH, RATE, reserved bit, and parity bit (with 6 zero tail bits appended) constitute a separate single OFDM symbol, denoted SIGNAL, which is transmitted with the most robust combination of BPSK modulation and a coding rate of R = 1/2. The SERVICE field of the PLCP header and the PSDU (with 6 zero tail bits and pad bits appended), denoted as DATA, are transmitted at the data rate described in the RATE field and may constitute multiple OFDM symbols. The tail bits in the SIGNAL symbol enable decoding of the RATE and LENGTH fields immediately after the reception of the tail bits. The RATE and LENGTH fields are required for decoding the DATA part of the packet. In addition, the CCA mechanism is augmented by predicting the duration of the packet from the contents of the RATE and LENGTH fields, even if the data rate is not supported by the STA. Each of these fields is described in detail in 18.3.3, 18.3.4, and 18.3.5.

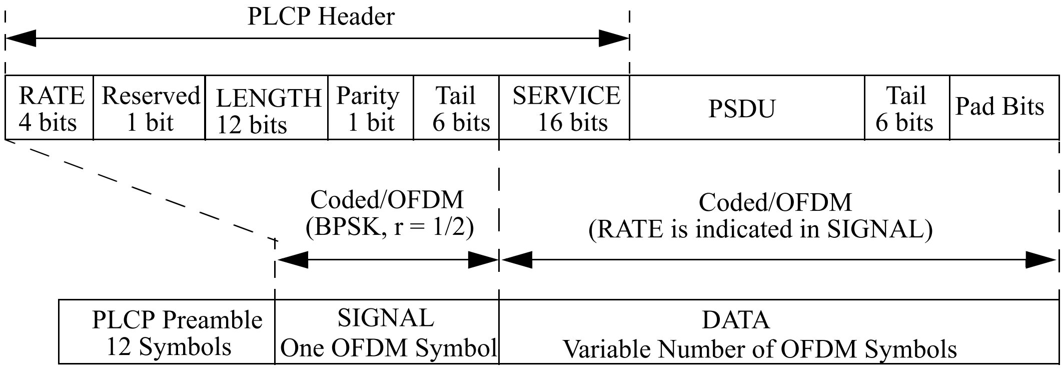


Figure 18-1—PPDU frame format

18.3.2.2 Overview of the PPDU encoding process

The encoding process is composed of many detailed steps, which are described fully in later subclauses, as noted below. The following overview intends to facilitate understanding the details of the convergence procedure:

a) Produce the PLCP Preamble field, composed of 10 repetitions of a “short training sequence” (used for AGC convergence, diversity selection, timing acquisition, and coarse frequency acquisition in the receiver) and two repetitions of a “long training sequence” (used for channel estimation and fine frequency acquisition in the receiver), preceded by a guard interval (GI). Refer to 18.3.3 for details.

b) Produce the PLCP header field from the RATE, LENGTH, and SERVICE fields of the TXVECTOR by filling the appropriate bit fields. The RATE and LENGTH fields of the PLCP header are encoded by a convolutional code at a rate of R = 1/2, and are subsequently mapped onto a single BPSK encoded OFDM symbol, denoted as the SIGNAL symbol. In order to facilitate a reliable and timely detection of the RATE and LENGTH fields, 6 zero tail bits are inserted into the PLCP header. The encoding of the SIGNAL field into an OFDM symbol follows the same steps for convolutional encoding, interleaving, BPSK modulation, pilot insertion, Fourier transform, and prepending a GI as described subsequently for data transmission with BPSK-OFDM modulated at coding rate 1/2. The contents of the SIGNAL field are not scrambled. Refer to 18.3.4 for details.

c) Calculate from RATE field of the TXVECTOR the number of data bits per OFDM symbol (N_{DBPS}), the coding rate (R), the number of bits in each OFDM subcarrier (N_{BPSC}), and the number of coded bits per OFDM symbol (N_{CBPS}). Refer to 18.3.2.3 for details.

d) Append the PSDU to the SERVICE field of the TXVECTOR. Extend the resulting bit string with zero bits (at least 6 bits) so that the resulting length is a multiple of N_{DBPS}. The resulting bit string constitutes the DATA part of the packet. Refer to 18.3.5.4 for details.

e) Initiate the scrambler with a pseudorandom nonzero seed, generate a scrambling sequence, and XOR it with the extended string of data bits. Refer to 18.3.5.5 for details.

f) Replace the six scrambled zero bits following the data with six nonscrambled zero bits. (Those bits return the convolutional encoder to the zero state and are denoted as tail bits.) Refer to 18.3.5.3 for details.

g) Encode the extended, scrambled data string with a convolutional encoder (R = 1/2). Omit (puncture) some of the encoder output string (chosen according to “puncturing pattern”) to reach the desired “coding rate.” Refer to 18.3.5.6 for details.

h) Divide the encoded bit string into groups of N_{CBPS} bits. Within each group, perform an “interleaving” (reordering) of the bits according to a rule corresponding to the desired RATE. Refer to 18.3.5.7 for details.

i) Divide the resulting coded and interleaved data string into groups of N_{BPSC} bits. For each of the bit groups, convert the bit group into a complex number according to the modulation encoding tables. Refer to 18.3.5.8 for details.

j) Divide the complex number string into groups of 48 complex numbers. Each such group is associated with one OFDM symbol. In each group, the complex numbers are numbered 0 to 47 and mapped hereafter into OFDM subcarriers numbered –26 to –22, –20 to –8, –6 to –1, 1 to 6, 8 to 20, and 22 to 26. The subcarriers –21, –7, 7, and 21 are skipped and, subsequently, used for inserting pilot subcarriers. The 0 subcarrier, associated with center frequency, is omitted and filled with the value 0. Refer to 18.3.5.10 for details.

k) Four subcarriers are inserted as pilots into positions –21, –7, 7, and 21. The total number of the subcarriers is 52 (48 + 4). Refer to 18.3.5.9 for details.

l) For each group of subcarriers –26 to 26, convert the subcarriers to time domain using inverse Fourier transform. Prepend to the Fourier-transformed waveform a circular extension of itself thus forming a GI, and truncate the resulting periodic waveform to a single OFDM symbol length by applying time domain windowing. Refer to 18.3.5.10 for details.

m) Append the OFDM symbols one after another, starting after the SIGNAL symbol describing the RATE and LENGTH fields. Refer to 18.3.5.10 for details.

n) Up-convert the resulting “complex baseband” waveform to an RF according to the center frequency of the desired channel and transmit. Refer to 18.3.2.5 and 18.3.8.2 for details.

An illustration of the transmitted frame and its parts appears in Figure 18-4 (in 18.3.3).

18.3.2.3 Modulation-dependent parameters

The modulation parameters dependent on the data rate used shall be set according to Table 18-4.

Table 18-4—Modulation-dependent parameters

Modulation	Coding rate (R)	Coded bits per subcarrier (N_{BPSC})	Coded bits per OFDM symbol (N_{CBPS})	Data bits per OFDM symbol (N_{DBPS})	Data rate (Mb/s) (20 MHz channel spacing)	Data rate (Mb/s) (10 MHz channel spacing)	Data rate (Mb/s) (5 MHz channel spacing)
BPSK	1/2	1	48	24	6	3	1.5
BPSK	3/4	1	48	36	9	4.5	2.25
QPSK	1/2	2	96	48	12	6	3
QPSK	3/4	2	96	72	18	9	4.5
16-QAM	1/2	4	192	96	24	12	6
16-QAM	3/4	4	192	144	36	18	9
64-QAM	2/3	6	288	192	48	24	12
64-QAM	3/4	6	288	216	54	27	13.5

18.3.2.4 Timing related parameters

Table 18-5 is the list of timing parameters associated with the OFDM PLCP.

Table 18-5—Timing-related parameters

Parameter	Value (20 MHz channel spacing)	Value (10 MHz channel spacing)	Value (5 MHz channel spacing)
N_{SD}: Number of data subcarriers	48	48	48
N_{SP}: Number of pilot subcarriers	4	4	4
N_{ST}: Number of subcarriers, total	52 ($N_{SD} + N_{SP}$)	52 ($N_{SD} + N_{SP}$)	52 ($N_{SD} + N_{SP}$)
Δ_F: Subcarrier frequency spacing	0.3125 MHz (=20 MHz/64)	0.15625 MHz (= 10 MHz/64)	0.078125 MHz (= 5 MHz/64)
T_{FFT}: Inverse Fast Fourier Transform (IFFT) / Fast Fourier Transform (FFT) period	3.2 μs ($1/\Delta_F$)	6.4 μs ($1/\Delta_F$)	12.8 μs ($1/\Delta_F$)

Table 18-5—Timing-related parameters *(continued)*

Parameter	Value (20 MHz channel spacing)	Value (10 MHz channel spacing)	Value (5 MHz channel spacing)
$T_{PREAMBLE}$: PLCP preamble duration	16 µs ($T_{SHORT} + T_{LONG}$)	32 µs ($T_{SHORT} + T_{LONG}$)	64 µs ($T_{SHORT} + T_{LONG}$)
T_{SIGNAL}: Duration of the SIGNAL BPSK-OFDM symbol	4.0 µs ($T_{GI} + T_{FFT}$)	8.0 µs ($T_{GI} + T_{FFT}$)	16.0 µs ($T_{GI} + T_{FFT}$)
T_{GI}: GI duration	0.8 µs ($T_{FFT}/4$)	1.6 µs ($T_{FFT}/4$)	3.2 µs ($T_{FFT}/4$)
T_{GI2}: Training symbol GI duration	1.6 µs ($T_{FFT}/2$)	3.2 µs ($T_{FFT}/2$)	6.4 µs ($T_{FFT}/2$)
T_{SYM}: Symbol interval	4 µs ($T_{GI} + T_{FFT}$)	8 µs ($T_{GI} + T_{FFT}$)	16 µs ($T_{GI} + T_{FFT}$)
T_{SHORT}: Short training sequence duration	8 µs ($10 \times T_{FFT}/4$)	16 µs ($10 \times T_{FFT}/4$)	32 µs ($10 \times T_{FFT}/4$)
T_{LONG}: Long training sequence duration	8 µs ($T_{GI2} + 2 \times T_{FFT}$)	16 µs ($T_{GI2} + 2 \times T_{FFT}$)	32 µs ($T_{GI2} + 2 \times T_{FFT}$)

18.3.2.5 Mathematical conventions in the signal descriptions

The transmitted signals are described in a complex baseband signal notation. The actual transmitted signal is related to the complex baseband signal by the following relation:

$$r_{(RF)}\langle t\rangle = Re\{r\langle t\rangle \exp\langle j2\pi f_c t\rangle\} \tag{18-1}$$

where

Re(.) represents the real part of a complex variable

f_c denotes the carrier center frequency

The transmitted baseband signal is composed of contributions from several OFDM symbols.

$$r_{PACKET}(t) = r_{PREAMBLE}(t) + r_{SIGNAL}(t - t_{SIGNAL}) + r_{DATA}(t - t_{DATA}) \tag{18-2}$$

The subframes of which Equation (18-2) are composed are described in 18.3.3, 18.3.4, and 18.3.5.10. The time offsets t$_{SUBFRAME}$ determine the starting time of the corresponding subframe; t_{SIGNAL} is equal to 16 µs for 20 MHz channel spacing, 32 µs for 10 MHz channel spacing, and 64 µs for 5 MHz channel spacing, and t_{DATA} is equal to 20 µs for 20 MHz channel spacing, 40 µs for 10 MHz channel spacing, and 80 µs for 5 MHz channel spacing.

All the subframes of the signal are constructed as an inverse Fourier transform of a set of coefficients, C_k, with C_k defined later as data, pilots, or training symbols in 18.3.3 to 18.3.5.

$$r_{SUBFRAME}(t) = w_{TSUBFRAME}(t) \sum_{k=-N_{ST}/2}^{N_{ST}/2} C_k \exp\,(j2\pi k\Delta_f)(t - T_{GUARD}) \tag{18-3}$$

The parameters Δ_F and N_{ST} are described in Table 18-5. The resulting waveform is periodic with a period of $T_{FFT} = 1/\Delta_F$. Shifting the time by T_{GUARD} creates the “circular prefix” used in OFDM to avoid ISI from the previous frame. Three kinds of T_{GUARD} are defined: for the short training sequence (= 0 μs), for the long training sequence (= T_{GI2}), and for data OFDM symbols (= T_{GI}). (Refer to Table 18-5.) The boundaries of the subframe are set by a multiplication by a time-windowing function, $w_{TSUBFRAME}(t)$, which is defined as a rectangular pulse, $w_T(t)$, of duration T, accepting the value $T_{SUBFRAME}$. The time-windowing function, $w_T(t)$, depending on the value of the duration parameter, T, may extend over more than one period, T_{FFT}. In particular, window functions that extend over multiple periods of the FFT are utilized in the definition of the preamble. Figure 18-2 illustrates the possibility of extending the windowing function over more than one period, T_{FFT}, and additionally shows smoothed transitions by application of a windowing function, as exemplified in Equation (18-4). In particular, window functions that extend over multiple periods of the FFT are utilized in the definition of the preamble.

$$w_T(t) = \begin{cases} \sin^2\left(\frac{\pi}{2}(0.5 + t/T_{TR})\right) & (-T_{TR}/2 < t < T_{TR}/2) \\ 1 & (T_{TR}/2 \le t < T - T_{TR}/2) \\ \sin^2\left(\frac{\pi}{2}(0.5 - (t-T)/T_{TR})\right) & (T - T_{TR}/2 \le t < T + T_{TR}/2) \end{cases} \qquad (18\text{-}4)$$

In the case of vanishing T_{TR}, the windowing function degenerates into a rectangular pulse of duration T. The normative specifications of generating the transmitted waveforms shall utilize the rectangular pulse shape. In implementation, higher T_{TR} is typically implemented in order to smooth the transitions between the consecutive subsections. This creates a small overlap between them, of duration T_{TR}, as shown in Figure 18-2. The transition time, T_{TR}, is about 100 ns. Smoothing the transition is required in order to reduce the spectral sidelobes of the transmitted waveform. However, the binding requirements are the spectral mask and modulation accuracy requirements, as detailed in 18.3.9.3 and 18.3.9.7. Time domain windowing, as described here, is just one way to achieve those objectives. The implementer may use other methods to achieve the same goal, such as frequency domain filtering. Therefore, the transition shape and duration of the transition are informative parameters.

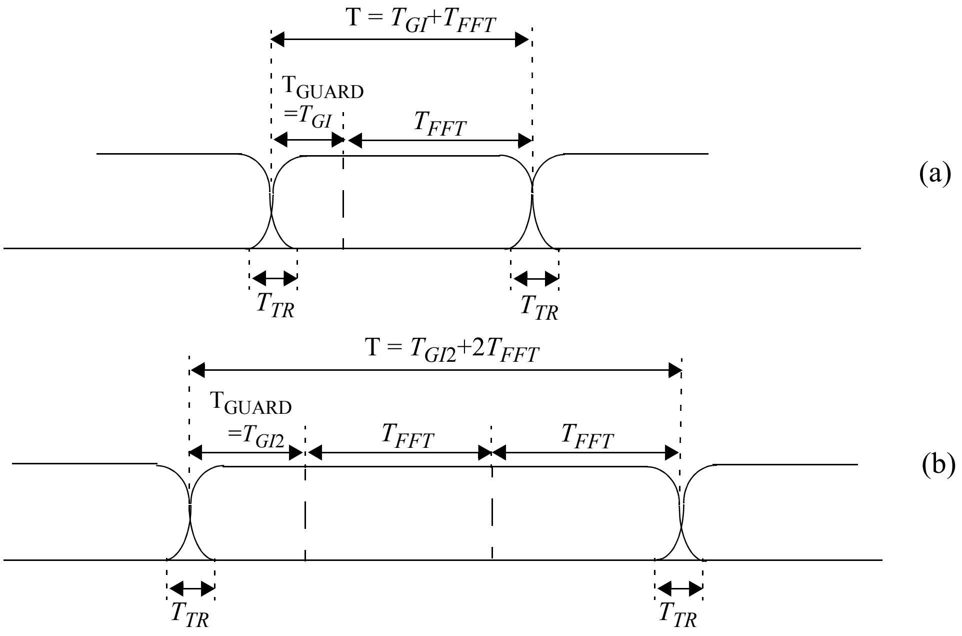


Figure 18-2—Illustration of OFDM frame with cyclic extension and windowing for (a) single reception or (b) two receptions of the FFT period

18.3.2.6 Discrete time implementation considerations

The following descriptions of the discrete time implementation are informational.

In a typical implementation, the windowing function is represented in discrete time. As an example, when a windowing function with parameters T = 4.0 µs and a T_{TR} = 100 ns is applied, and the signal is sampled at 20 Msample/s, it becomes

$$w_T[n] = w_T(nT_S) = \begin{Bmatrix} 1 & 1 \le n \le 79 \\ 0.5 & 0, 80 \\ 0 & otherwise \end{Bmatrix} \tag{18-5}$$

The common way to implement the inverse Fourier transform, as shown in Equation (18-3), is by an IFFT algorithm. If, for example, a 64-point IFFT is used, the coefficients 1 to 26 are mapped to the same numbered IFFT inputs, while the coefficients –26 to –1 are copied into IFFT inputs 38 to 63. The rest of the inputs, 27 to 37 and the 0 (dc) input, are set to 0. This mapping is illustrated in Figure 18-3. After performing an IFFT, the output is cyclically extended to the desired length.

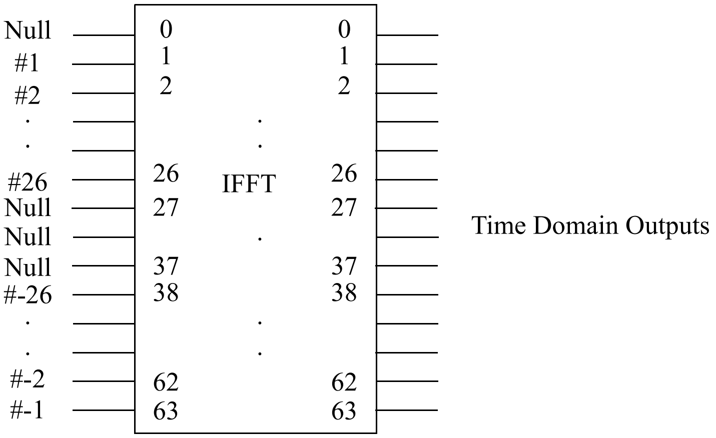


Figure 18-3—Inputs and outputs of inverse Fourier transform

18.3.3 PLCP preamble (SYNC)

The PLCP Preamble field is used for synchronization. It consists of 10 short symbols and two long symbols that are shown in Figure 18-4 and described in this subclause. The timings described in this subclause and shown in Figure 18-4 are for 20 MHz channel spacing. They are doubled for half-clocked (i.e., 10 MHz) channel spacing and are quadrupled for quarter-clocked (i.e., 5 MHz) channel spacing.

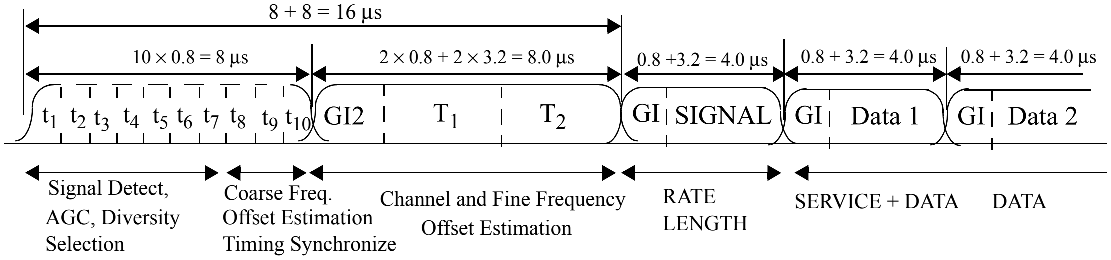

8 + 8 = 16 µs
10 × 0.8 = 8 µs
2 × 0.8 + 2 × 3.2 = 8.0 µs
0.8 +3.2 = 4.0 µs
0.8 + 3.2 = 4.0 µs
0.8 + 3.2 = 4.0 µs
t1 t2 t3 t4 t5 t6 t7 t8 t9 t10
GI2
T1
T2
GI
SIGNAL
GI
Data 1
GI
Data 2
Signal Detect, AGC, Diversity Selection
Coarse Freq. Offset Estimation Timing Synchronize
Channel and Fine Frequency Offset Estimation
RATE LENGTH
SERVICE + DATA
DATA

Figure 18-4—OFDM training structure

Figure 18-4 shows the OFDM training structure (PLCP preamble), where t_1 to t_{10} denote short training symbols and T_1 and T_2 denote long training symbols. The PLCP preamble is followed by the SIGNAL field and DATA. The total training length is 16 µs. The dashed boundaries in the figure denote repetitions due to the periodicity of the inverse Fourier transform.

A short OFDM training symbol consists of 12 subcarriers, which are modulated by the elements of the sequence S, given by

$$S_{-26,\,26} = \sqrt{(13/6)} \times \{0, 0, 1+j, 0, 0, 0, -1-j, 0, 0, 0, 1+j, 0, 0, 0, -1-j, 0, 0, 0, -1-j, 0, 0, 0, 1+j, 0, 0, 0, 0,$$

$$0, 0, 0, -1-j, 0, 0, 0, -1-j, 0, 0, 0, 1+j, 0, 0, 0, 1+j, 0, 0, 0, 1+j, 0, 0, 0, 1+j, 0,0\} \quad (18\text{-}6)$$

The multiplication by a factor of $\sqrt{(13/6)}$ is in order to normalize the average power of the resulting OFDM symbol, which utilizes 12 out of 52 subcarriers.

The signal shall be generated according to the following equation:

$$r_{SHORT}(t) = w_{TSHORT}(t) \sum_{k=-N_{ST}/2}^{N_{ST}/2} S_k \exp(j2\pi k\Delta_F t) \quad (18\text{-}7)$$

The fact that only spectral lines of $S_{-26:26}$ with indices that are a multiple of 4 have nonzero amplitude results in a periodicity of $T_{FFT}/4$ = 0.8 µs. The interval T_{SHORT} is equal to ten 0.8 µs periods (i.e., 8 µs).

Generation of the short training sequence is illustrated in Table L-2.

A long OFDM training symbol consists of 53 subcarriers (including the value 0 at dc), which are modulated by the elements of the sequence L, given by

$$L_{-26,\,26} = \{1, 1, -1, -1, 1, 1, -1, 1, -1, 1, 1, 1, 1, 1, 1, -1, -1, 1, 1, -1, 1, -1, 1, 1, 1, 1, 0,$$

$$1, -1, -1, 1, 1, -1, 1, -1, 1, -1, -1, -1, -1, -1, 1, 1, -1, -1, 1, -1, 1, -1, 1, 1, 1, 1\} \quad (18\text{-}8)$$

A long OFDM training symbol shall be generated according to the following equation:

$$r_{LONG}(t) = w_{TLONG}(t) \sum_{k=-N_{ST}/2}^{N_{ST}/2} L_k \exp(j2\pi k\Delta_F (t - T_{G12})) \quad (18\text{-}9)$$

where

$T_{G\,12}$ = 1.6 µs

Two periods of the long sequence are transmitted for improved channel estimation accuracy, yielding T_{LONG} = 1.6 + 2 × 3.2 = 8 µs.

An illustration of the long training sequence generation is given in Table L-5.

The sections of short repetitions and long repetitions shall be concatenated to form the preamble

$$r_{PREAMBLE}(t) = r_{SHORT}(t) + r_{LONG}(t - T_{SHORT}) \quad (18\text{-}10)$$

18.3.4 SIGNAL field

18.3.4.1 General

The OFDM training symbols shall be followed by the SIGNAL field, which contains the RATE and the LENGTH fields of the TXVECTOR. The RATE field conveys information about the type of modulation and the coding rate as used in the rest of the packet. The encoding of the SIGNAL single OFDM symbol shall be performed with BPSK modulation of the subcarriers and using convolutional coding at R = 1/2. The encoding procedure, which includes convolutional encoding, interleaving, modulation mapping processes, pilot insertion, and OFDM modulation, follows the steps described in 18.3.5.6, 18.3.5.7, and 18.3.5.9, as used for transmission of data with BPSK-OFDM modulated at coding rate 1/2. The contents of the SIGNAL field are not scrambled.

The SIGNAL field shall be composed of 24 bits, as illustrated in Figure 18-5. The four bits 0 to 3 shall encode the RATE. Bit 4 shall be reserved for future use. Bits 5–16 shall encode the LENGTH field of the TXVECTOR, with the LSB being transmitted first.

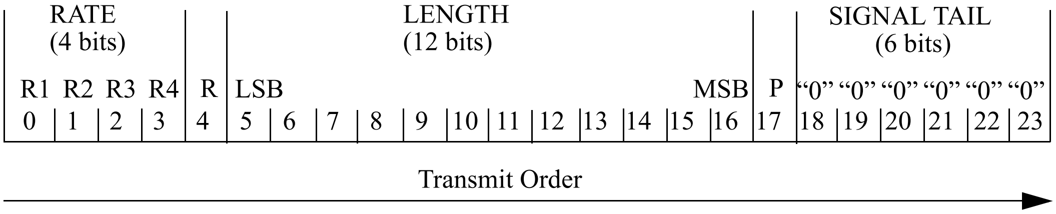


Figure 18-5—SIGNAL field bit assignment

The process of generating the SIGNAL OFDM symbol is illustrated in L.1.4.

18.3.4.2 RATE field

The bits R1–R4 shall be set, dependent on RATE, according to the values in Table 18-6.

Table 18-6—Contents of the SIGNAL field

R1–R4	Rate (Mb/s) (20 MHz channel spacing)	Rate (Mb/s) (10 MHz channel spacing)	Rate (Mb/s) (5 MHz channel spacing)
1101	6	3	1.5
1111	9	4.5	2.25
0101	12	6	3
0111	18	9	4.5
1001	24	12	6
1011	36	18	9
0001	48	24	12
0011	54	27	13.5

18.3.4.3 PLCP LENGTH field

The PLCP LENGTH field shall be an unsigned 12-bit integer that indicates the number of octets in the PSDU that the MAC is currently requesting the PHY to transmit. This value is used by the PHY to determine the number of octet transfers that will occur between the MAC and the PHY after receiving a request to start transmission. The transmitted value shall be determined from the LENGTH parameter in the TXVECTOR issued with the PHY-TXSTART.request primitive described in 7.3.5.5. The LSB shall be transmitted first in time. This field shall be encoded by the convolutional encoder described in 18.3.5.6.

18.3.4.4 Parity (P), Reserved (R), and SIGNAL TAIL fields

Bit 4 is reserved. It shall be set to 0 on transmit and ignored on receive. Bit 17 shall be a positive parity (even parity) bit for bits 0–16. The bits 18–23 constitute the SIGNAL TAIL field, and all 6 bits shall be set to 0.

18.3.5 DATA field

18.3.5.1 General

The DATA field contains the SERVICE field, the PSDU, the TAIL bits, and the PAD bits, if needed, as described in 18.3.5.3 and 18.3.5.4. All bits in the DATA field are scrambled, as described in 18.3.5.5.

18.3.5.2 SERVICE field

The IEEE 802.11 SERVICE field has 16 bits, which shall be denoted as bits 0–15. The bit 0 shall be transmitted first in time. The bits from 0–6 of the SERVICE field, which are transmitted first, are set to 0s and are used to synchronize the descrambler in the receiver. The remaining 9 bits (7–15) of the SERVICE field shall be reserved for future use. All reserved bits shall be set to 0. Refer to Figure 18-6.

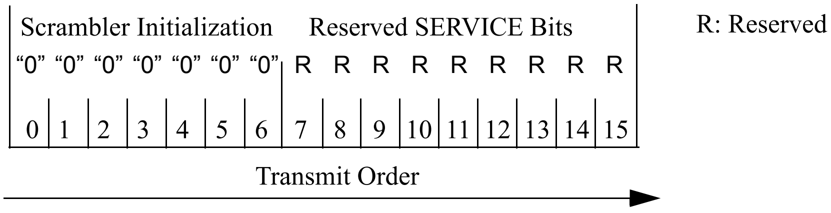


Figure 18-6—SERVICE field bit assignment

18.3.5.3 PPDU TAIL field

The PPDU TAIL field shall be six bits of 0, which are required to return the convolutional encoder to the zero state. This procedure improves the error probability of the convolutional decoder, which relies on future bits when decoding and which may be not be available past the end of the message. The PLCP tail bit field shall be produced by replacing six scrambled zero bits following the message end with six nonscrambled zero bits.

18.3.5.4 Pad bits (PAD)

The number of bits in the DATA field shall be a multiple of N_{CBPS}, the number of coded bits in an OFDM symbol (48, 96, 192, or 288 bits). To achieve that, the length of the message is extended so that it becomes a multiple of N_{DBPS}, the number of data bits per OFDM symbol. At least 6 bits are appended to the message, in order to accommodate the TAIL bits, as described in 18.3.5.3. The number of OFDM symbols, N_{SYM}; the number of bits in the DATA field, N_{DATA}; and the number of pad bits, N_{PAD}, are computed from the length of the PSDU (LENGTH) as follows:

$$N_{SYM} = \text{Ceiling}\left((16 + 8 \times \text{LENGTH} + 6)/N_{DBPS}\right) \tag{18-11}$$

$$N_{DATA} = N_{SYM} \times N_{DBPS} \tag{18-12}$$

$$N_{PAD} = N_{DATA} - (16 + 8 \times \text{LENGTH} + 6) \tag{18-13}$$

The function Ceiling (.) is a function that returns the smallest integer value greater than or equal to its argument value. The appended bits ("pad bits") are set to 0 and are subsequently scrambled with the rest of the bits in the DATA field.

An example of a DATA field that contains the SERVICE field, DATA, tail, and pad bits is given in L.1.5.1.

18.3.5.5 PLCP DATA scrambler and descrambler

The DATA field, composed of SERVICE, PSDU, tail, and pad parts, shall be scrambled with a length-127 frame-synchronous scrambler. The octets of the PSDU are placed in the transmit serial bit stream, bit 0 first and bit 7 last. The frame synchronous scrambler uses the generator polynomial *S(x)* as follows, and is illustrated in Figure 18-7:

$$S(x) = x^7 + x^4 + 1 \tag{18-14}$$

The 127-bit sequence generated repeatedly by the scrambler shall be (leftmost used first), 00001110 11110010 11001001 00000010 00100110 00101110 10110110 00001100 11010100 11100111 10110100 00101010 11111010 01010001 10111000 1111111, when the all ones initial state is used. The same scrambler is used to scramble transmit data and to descramble receive data. When transmitting, the initial state of the scrambler shall be set to a pseudorandom nonzero state. The seven LSBs of the SERVICE field shall be set to all zeros prior to scrambling to enable estimation of the initial state of the scrambler in the receiver.

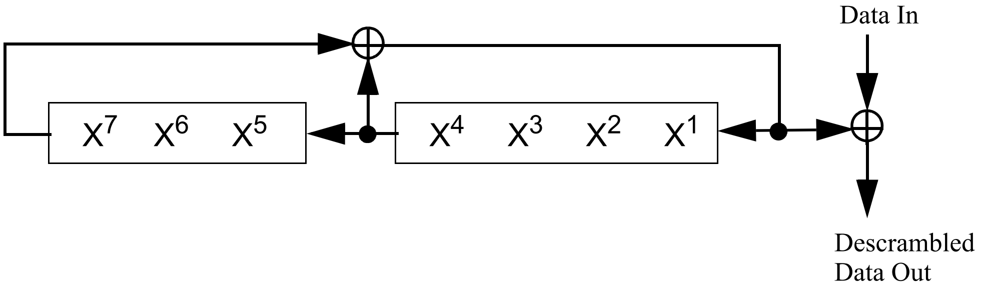


Figure 18-7—Data scrambler

An example of the scrambler output is illustrated in L.1.5.2.

18.3.5.6 Convolutional encoder

The DATA field, composed of SERVICE, PSDU, tail, and pad parts, shall be coded with a convolutional encoder of coding rate R = 1/2, 2/3, or 3/4, corresponding to the desired data rate. The convolutional encoder shall use the industry-standard generator polynomials, $g_0 = 133_8$ and $g_1 = 171_8$, of rate R = 1/2, as shown in Figure 18-8. The bit denoted as "A" shall be output from the encoder before the bit denoted as "B." Higher rates are derived from it by employing "puncturing." Puncturing is a procedure for omitting some of the encoded bits in the transmitter (thus reducing the number of transmitted bits and increasing the coding rate) and inserting a dummy "zero" metric into the convolutional decoder on the receive side in place of the omitted bits. The puncturing patterns are illustrated in Figure 18-9. Decoding by the Viterbi algorithm is recommended.

An example of encoding operation is shown in L.1.6.1.

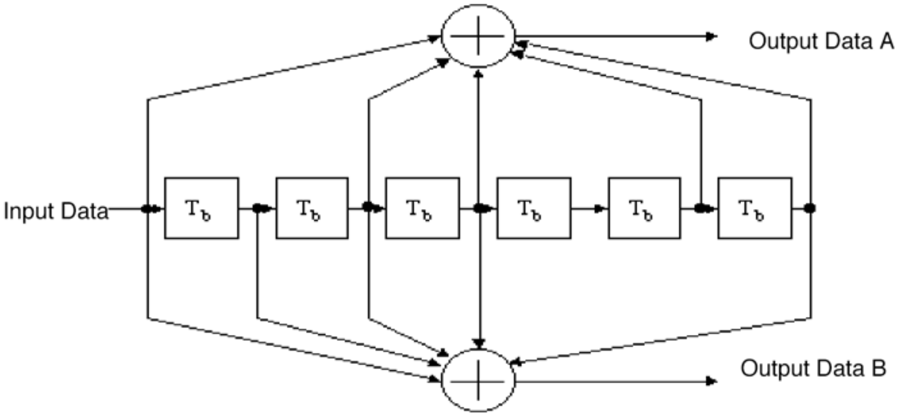


Figure 18-8—Convolutional encoder (k = 7)

18.3.5.7 Data interleaving

All encoded data bits shall be interleaved by a block interleaver with a block size corresponding to the number of bits in a single OFDM symbol, N_{CBPS}. The interleaver is defined by a two-step permutation. The first permutation ensures that adjacent coded bits are mapped onto nonadjacent subcarriers. The second ensures that adjacent coded bits are mapped alternately onto less and more significant bits of the constellation and, thereby, long runs of low reliability (LSB) bits are avoided.

The index of the coded bit before the first permutation shall be denoted by *k*; *i* shall be the index after the first and before the second permutation; and *j* shall be the index after the second permutation, just prior to modulation mapping.

The first permutation is defined by the rule

$$i = (N_{CBPS}/16)\ (k \bmod 16) + \mathrm{Floor}(k/16)\quad k = 0,1,\ldots,N_{CBPS} - 1 \tag{18-15}$$

The function Floor (.) denotes the largest integer not exceeding the parameter.

The second permutation is defined by the rule

$$j = s \times \mathrm{Floor}(i/s) + (i + N_{CBPS} - \mathrm{Floor}(16 \times i/N_{CBPS})) \bmod s\quad i = 0,1,\ldots\ N_{CBPS} - 1 \tag{18-16}$$

The value of s is determined by the number of coded bits per subcarrier, N_{BPSC}, according to

$$s = \max(N_{BPSC}/2,1) \tag{18-17}$$

The deinterleaver, which performs the inverse relation, is also defined by two permutations.

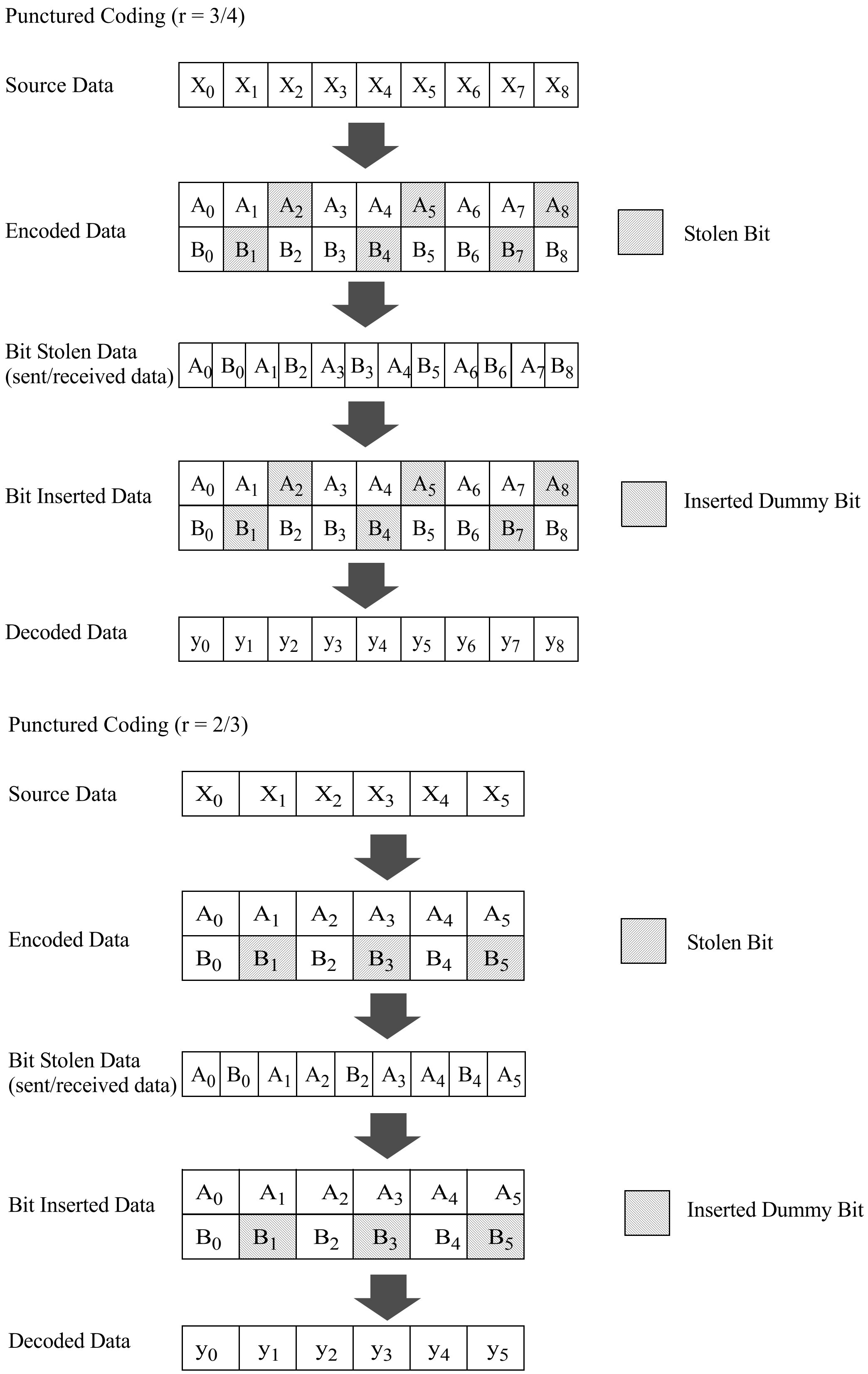


Figure 18-9—Example of the bit-stealing and bit-insertion procedure (r = 3/4, 2/3)

Here the index of the original received bit before the first permutation shall be denoted by *j*; *i* shall be the index after the first and before the second permutation; and *k* shall be the index after the second permutation, just prior to delivering the coded bits to the convolutional (Viterbi) decoder.

The first permutation is defined by the rule

$$i = s \times \text{Floor}(j/s) + (j + \text{Floor}(16 \times j/N_{CBPS})) \bmod s \quad j = 0,1,\ldots\ N_{CBPS} - 1 \tag{18-18}$$

where
s is defined in Equation (18-17).

This permutation is the inverse of the permutation described in Equation (18-16).

The second permutation is defined by the rule

$$k = 16 \times i - (N_{CBPS} - 1)\text{Floor}(16 \times i/N_{CBPS}) \quad i = 0,1,\ldots\ N_{CBPS} - 1 \tag{18-19}$$

This permutation is the inverse of the permutation described in Equation (18-15).

An example of interleaving operation is illustrated in L.1.6.2.

18.3.5.8 Subcarrier modulation mapping

The OFDM subcarriers shall be modulated by using BPSK, QPSK, 16-QAM, or 64-QAM, depending on the RATE requested. The encoded and interleaved binary serial input data shall be divided into groups of N_{BPSC} (1, 2, 4, or 6) bits and converted into complex numbers representing BPSK, QPSK, 16-QAM, or 64-QAM constellation points. The conversion shall be performed according to Gray-coded constellation mappings, illustrated in Figure 18-10, with the input bit, b_0, being the earliest in the stream. The output values, d, are formed by multiplying the resulting (I+jQ) value by a normalization factor K_{MOD}, as described in Equation (18-20).

$$d = (I + jQ) \times K_{MOD} \tag{18-20}$$

The normalization factor, K_{MOD}, depends on the base modulation mode, as prescribed in Table 18-7. Note that the modulation type can be different from the start to the end of the transmission, as the signal changes from SIGNAL to DATA, as shown in Figure 18-1. The purpose of the normalization factor is to achieve the same average power for all mappings. In practical implementations, an approximate value of the normalization factor may be used, as long as the device conforms with the modulation accuracy requirements described in 18.3.9.7.

Table 18-7—Modulation-dependent normalization factor K_{MOD}

Modulation	K_{MOD}
BPSK	1
QPSK	$1/\sqrt{2}$
16-QAM	$1/\sqrt{10}$
64-QAM	$1/\sqrt{42}$

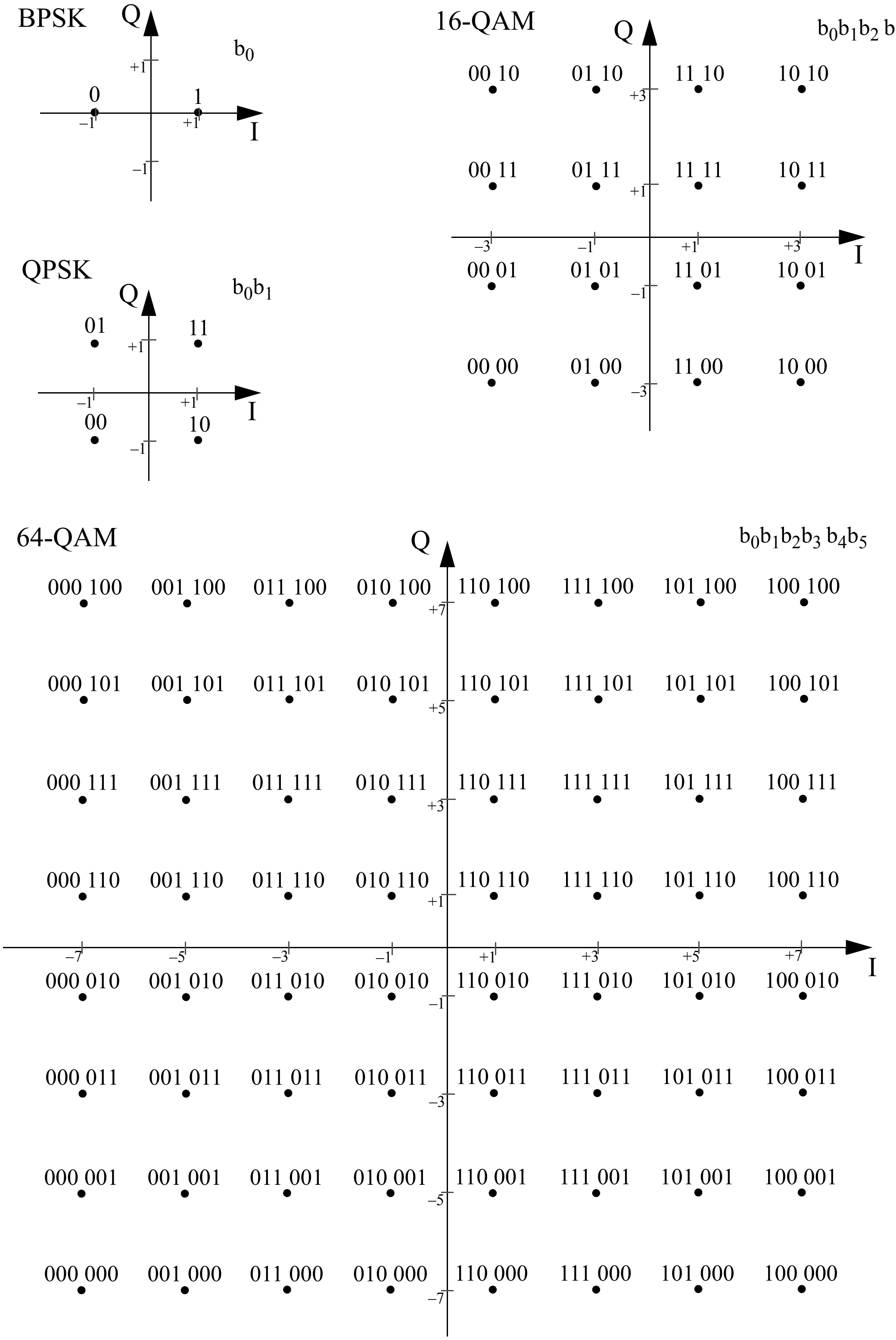


Figure 18-10—BPSK, QPSK, 16-QAM, and 64-QAM constellation bit encoding

For BPSK, b_0 determines the I value, as illustrated in Table 18-8. For QPSK, b_0 determines the I value and b_1 determines the Q value, as illustrated in Table 18-9. For 16-QAM, b_0b_1 determines the I value and b_2b_3 determines the Q value, as illustrated in Table 18-10. For 64-QAM, $b_0b_1b_2$ determines the I value and $b_3b_4b_5$ determines the Q value, as illustrated in Table 18-11.

Table 18-8—BPSK encoding table

Input bit (b_0)	I-out	Q-out
0	–1	0
1	1	0

Table 18-9—QPSK encoding table

Input bit (b_0)	I-out
0	–1
1	1

Input bit (b_1)	Q-out
0	–1
1	1

Table 18-10—16-QAM encoding table

Input bits (b_0 b_1)	I-out
00	–3
01	–1
11	1
10	3

Input bits (b_2 b_3)	Q-out
00	–3
01	–1
11	1
10	3

Table 18-11—64-QAM encoding table

Input bits (b_0 b_1 b_2)	I-out
000	–7
001	–5
011	–3
010	–1
110	1
111	3
101	5
100	7

Input bits (b_3 b_4 b_5)	Q-out
000	–7
001	–5
011	–3
010	–1
110	1
111	3
101	5
100	7

18.3.5.9 Pilot subcarriers

In each OFDM symbol, four of the subcarriers are dedicated to pilot signals in order to make the coherent detection robust against frequency offsets and phase noise. These pilot signals shall be put in subcarriers –21, –7, 7, and 21. The pilots shall be BPSK modulated by a pseudo-binary sequence to prevent the generation of spectral lines. The contribution of the pilot subcarriers to each OFDM symbol is described in 18.3.5.10.

18.3.5.10 OFDM modulation

The stream of complex numbers is divided into groups of $N_{SD} = 48$ complex numbers. This shall be denoted by writing the complex number $d_{k,n}$, which corresponds to subcarrier k of OFDM symbol n, as follows:

$$d_{k,n} \equiv d_{k+N_{SD} \times n}, \qquad k = 0, \ldots N_{SD} - 1, n = 0, \ldots N_{SYM} - 1 \tag{18-21}$$

The number of OFDM symbols, N_{SYM}, was introduced in 18.3.5.4.

An OFDM symbol, $r_{DATA,n}(t)$, is defined as

$$r_{DATA,n}(t) = w_{TSYM}(t)\left(\sum_{k=0}^{N_{SD}-1} d_{k,n} \exp((j2\pi M(k)\Delta_F(t - T_{GI})) + p_{n+1}\sum_{k=-N_{ST}/2}^{N_{ST}/2} P_k \exp(j2\pi k\Delta_F(t - T_{GI}))\right) \tag{18-22}$$

where the function, *M(k)*, defines a mapping from the logical subcarrier number 0 to 47 into frequency offset index –26 to 26, while skipping the pilot subcarrier locations and the 0th (dc) subcarrier.

$$M(k) = \begin{cases} k - 26 & 0 \le k \le 4 \\ k - 25 & 5 \le k \le 17 \\ k - 24 & 18 \le k \le 23 \\ k - 23 & 24 \le k \le 29 \\ k - 22 & 30 \le k \le 42 \\ k - 21 & 43 \le k \le 47 \end{cases} \tag{18-23}$$

The contribution of the pilot subcarriers for the n^{th} OFDM symbol is produced by inverse Fourier transform of sequence P, given by

$$P_{-26,26} = \{0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,$$
$$0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1, 0, 0, 0, 0, 0\} \tag{18-24}$$

The polarity of the pilot subcarriers is controlled by the sequence, p_n, which is a cyclic extension of the 127 elements sequence and is given by

$$\begin{aligned} p_{0..126v} = \{&1,1,1,1,\ -1,-1,-1,1,\ -1,-1,-1,-1,\ 1,1,-1,1,-1,-1,1,1,\ -1,1,1,-1,\ 1,1,1,1,\ 1,1,-1,1,\\ &1,1,-1,1,\ 1,-1,-1,1,\ 1,1,-1,1,\ -1,-1,-1,1,\ -1,1,-1,-1,\ 1,-1,-1,1,\ 1,1,1,1,\ -1,-1,1,1,\\ &-1,-1,1,-1,\ 1,-1,1,1,\ -1,-1,-1,1,\ 1,-1,-1,-1,\ -1,1,-1,-1,\ 1,-1,1,1,\ 1,1,-1,1,\ -1,1,-1,1,\\ &-1,-1,-1,-1,\ -1,1,-1,1,\ 1,-1,1,-1,\ 1,1,1,-1,\ -1,1,-1,-1,\ -1,1,1,1,\ -1,-1,-1,-1,\ -1,-1,-1\} \end{aligned} \quad (18\text{-}25)$$

The sequence p_n is generated by the scrambler defined by Figure 18-7 when the all ones initial state is used, and by replacing all 1s with –1 and all 0s with 1. Each sequence element is used for one OFDM symbol. The first element, p_0, multiplies the pilot subcarriers of the SIGNAL symbol, while the elements from p_1 on are used for the DATA symbols.

The subcarrier frequency allocation is shown in Figure 18-11. To avoid difficulties in D/A and A/D converter offsets and carrier feedthrough in the RF system, the subcarrier falling at DC (0th subcarrier) is not used.

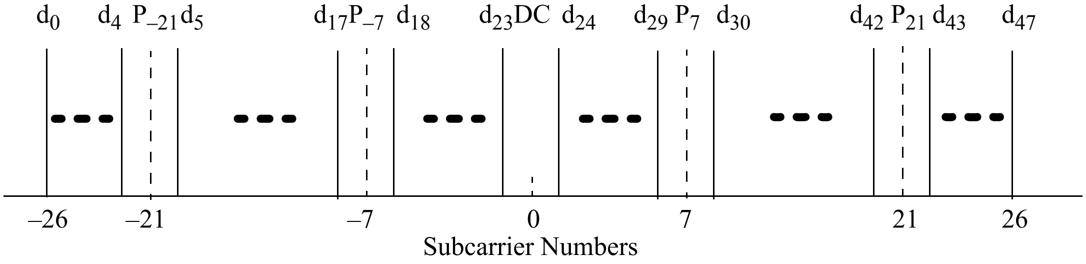


Figure 18-11—Subcarrier frequency allocation

The concatenation of N_{SYM} OFDM symbols is written as

$$r_{DATA}(t) = \sum_{n=0}^{N_{SYM}-1} r_{DATA,n}(t - nT_{SYM}) \quad (18\text{-}26)$$

An example of mapping into symbols is shown in L.1.6.3, as well as the scrambling of the pilot signals (see L.1.7). The final output of these operations is also shown in L.1.8.

18.3.6 CCA

PLCP shall provide the capability to perform CCA and report the result to the MAC. The CCA mechanism shall detect a “medium busy” condition with requirements specified in 18.3.10.6 and 18.3.12. This medium status report is indicated by the PHY_CCA.indication primitive.

18.3.7 PLCP data modulation and modulation rate change

The PLCP preamble shall be transmitted using an OFDM modulated fixed waveform. The IEEE 802.11 SIGNAL field, BPSK-OFDM modulated with coding rate 1/2, shall indicate the modulation and coding rate that shall be used to transmit the MPDU. The transmitter (receiver) shall initiate the modulation (demodulation) constellation and the coding rate according to the RATE indicated in the SIGNAL field. The MPDU transmission rate shall be set by the DATARATE parameter in the TXVECTOR, issued with the PHY-TXSTART.request primitive described in 18.2.2.

18.3.8 PMD operating specifications (general)

18.3.8.1 General

General specifications for the BPSK OFDM, QPSK OFDM, 16-QAM OFDM, and 64-QAM OFDM PMD sublayers are provided in 18.3.8.2 to 18.3.8.8. These specifications apply to both the receive and transmit functions and general operation of the OFDM PHY.

18.3.8.2 Outline description

The general block diagram of the transmitter and receiver for the OFDM PHY is shown in Figure 18-12. Major specifications for the OFDM PHY are listed in Table 18-12.

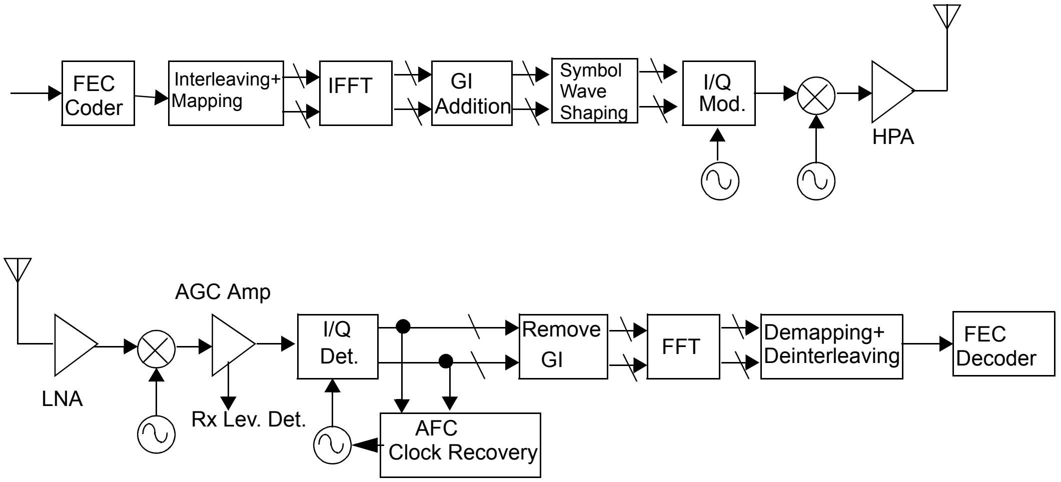


Figure 18-12—Transmitter and receiver block diagram for the OFDM PHY

Table 18-12—Major parameters of the OFDM PHY

Information data rate	**6, 9, 12, 18, 24, 36, 48, and 54 Mb/s (6, 12, and 24 Mb/s are mandatory) (20 MHz channel spacing)**	**3, 4.5, 6, 9, 12, 18, 24, and 27 Mb/s (3, 6, and 12 Mb/s are mandatory) (10 MHz channel spacing)**	**1.5, 2.25, 3, 4.5, 6, 9, 12, and 13.5 Mb/s (1.5, 3, and 6 Mb/s are mandatory) (5 MHz channel spacing)**
Modulation	BPSK OFDM QPSK OFDM 16-QAM OFDM 64-QAM OFDM	BPSK OFDM QPSK OFDM 16-QAM OFDM 64-QAM OFDM	BPSK OFDM QPSK OFDM 16-QAM OFDM 64-QAM OFDM
Error correcting code	K = 7 (64 states) convolutional code	K = 7 (64 states) convolutional code	K = 7 (64 states) convolutional code
Coding rate	1/2, 2/3, 3/4	1/2, 2/3, 3/4	1/2, 2/3, 3/4
Number of subcarriers	52	52	52
OFDM symbol duration	4.0 μs	8.0 μs	16.0 μs
GI	0.8 μs[a] (T_{GI})	1.6 μs (T_{GI})	3.2 μs (T_{GI})
Occupied bandwidth	16.6 MHz	8.3 MHz	4.15 MHz

[a] Refer to 18.3.2.5.

18.3.8.3 Regulatory requirements

WLANs implemented in accordance with this standard are subject to equipment certification and operating requirements established by regional and national regulatory administrations. The PMD specification establishes minimum technical requirements for interoperability, based upon established regulations at the time this standard was issued. These regulations are subject to revision, or may be superseded. Requirements that are subject to local geographic regulations are annotated within the PMD specification. Regulatory requirements that do not affect interoperability are not addressed in this standard. Implementers are referred to the regulatory sources in Annex D for further information. Operation in countries within defined regulatory domains may be subject to additional or alternative national regulations.

18.3.8.4 Operating channel frequencies

18.3.8.4.1 Operating frequency range

The OFDM PHY shall not operate in frequency bands not allocated by a regulatory body in its operational region. Regulatory requirements for a given frequency band are set by the regulatory authority responsible for spectrum management in a given geographic region or domain. The particular channelization to be used for this standard is dependent on such allocation, as well as the associated regulations for use of the allocations. These regulations are subject to revision, or may be superseded.

In some regulatory domains, several frequency bands may be available for OFDM PHY-based WLANs. These bands may be contiguous or not, and different regulatory limits may be applicable. A compliant OFDM PHY shall support at least one frequency band in at least one regulatory domain. The support of specific regulatory domains, and bands within the domains, shall be indicated by PLME attributes dot11RegDomainsImplementedValue and dot11FrequencyBandsImplemented.

The OFDM PHY shall use dot11CurrentFrequency to determine the operating frequency.

18.3.8.4.2 Channel numbering

Channel center frequencies are defined at every integral multiple of 5 MHz above Channel starting frequency. The relationship between center frequency and channel number is given by Equation (18-27):

$$\text{Channel center frequency} = \text{Channel starting frequency} + 5 \times n_{ch} \text{ (MHz)} \tag{18-27}$$

where

$n_{ch} = 1,\ldots 200.$

Channel starting frequency is defined as dot11ChannelStartingFactor × 500 kHz or
is defined as 5 GHz for systems where dot11OperatingClassesRequired is false or not defined.

For example, dot11ChannelStartingFactor = 10000 indicates that Channel 0 center frequency is 5.000 GHz. A channel center frequency of 5.000 GHz shall be indicated by dot11ChannelStartingFactor = 8000 and n_{ch} = 200. An SME managing multiple channel sets can change the channel set being managed by changing the value of dot11ChannelStartingFactor.

This definition provides a unique numbering system for all channels with 5 MHz between center frequencies, as well as the flexibility to define channelization sets for all current and future regulatory domains.

18.3.8.4.3 Channelization

The set of valid operating channel numbers by regulatory domain is defined in Annex E. As shown in Figure 18-11, no subcarrier is allocated on the channel center frequency.

18.3.8.5 Transmit and receive in-band and out-of-band spurious emissions

The OFDM PHY shall conform to in-band and out-of-band spurious emissions as set by regulatory bodies.

18.3.8.6 TX RF delay

The TX RF delay time shall be defined as the time between the issuance of a PMD.DATA.request primitive to the PMD and the start of the corresponding symbol at the air interface.

18.3.8.7 Slot time

The slot time for the OFDM PHY shall be 9 µs for 20 MHz channel spacing, shall be 13 µs for 10 MHz channel spacing, and shall be 21 µs for 5 MHz channel spacing.

Where dot11OperatingClassesRequired is true, the value of the slot time shall be increased by the value of 3 µs × coverage class. The default value of coverage class shall be 0.

NOTE—Distributed coordination function (DCF) operation over larger BSS diameters is facilitated by relaxing some PHY timing parameters, while maintaining compatibility with existing implementations in small BSS diameters.

18.3.8.8 Transmit and receive antenna port impedance

The transmit and receive antenna port(s) impedance shall be 50 Ω if the port is exposed.

18.3.9 PMD transmit specifications

18.3.9.1 General

The transmit specifications associated with the PMD sublayer are described in 18.3.9.2 to 18.3.9.8. In general, these are specified by primitives from the PLCP, and the transmit PMD entity provides the actual means by which the signals required by the PLCP primitives are imposed onto the medium.

18.3.9.2 Transmit power levels

The maximum allowable output power is measured in accordance with practices specified by the appropriate regulatory bodies.

18.3.9.3 Transmit spectrum mask

The transmit spectrum mask by regulatory domain is defined in Annex D and Annex E.

NOTE—In the presence of additional regulatory restrictions, the device needs to meet both the regulatory requirements and the mask defined here, i.e., its emissions need to be no higher at any frequency offset than the minimum of the values specified in the regulatory and default masks.

For operation using 20 MHz channel spacing, the transmitted spectrum shall have a 0 dBr (dB relative to the maximum spectral density of the signal) bandwidth not exceeding 18 MHz, –20 dBr at 11 MHz frequency offset, –28 dBr at 20 MHz frequency offset, and the maximum of –40 dBr and –53 dBm/MHz at 30 MHz frequency offset and above. The transmitted spectral density of the transmitted signal shall fall within the spectral mask, as shown in Figure 18-13. The measurements shall be made using a 100 kHz resolution bandwidth and a 30 kHz video bandwidth.

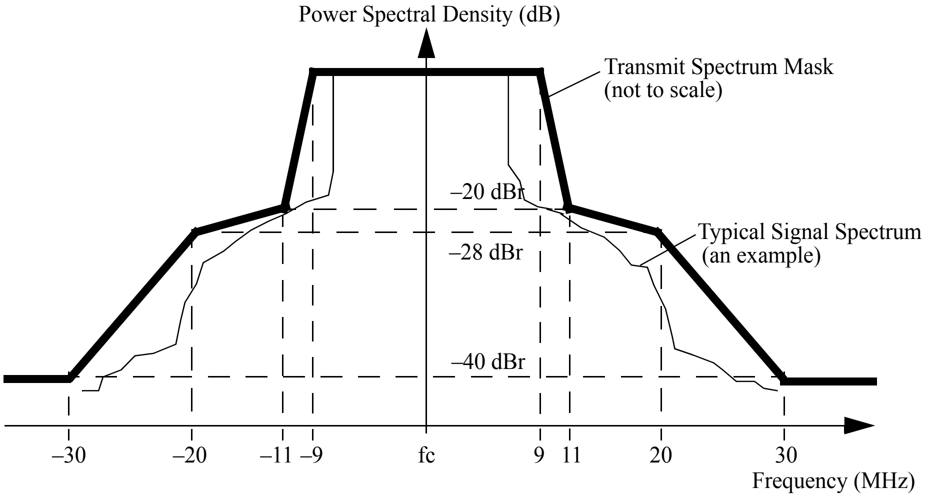


Figure 18-13—Transmit spectrum mask for 20 MHz transmission

For operation using 10 MHz channel spacing, the transmitted spectrum shall have a 0 dBr bandwidth not exceeding 9 MHz, –20 dBr at 5.5 MHz frequency offset, –28 dBr at 10 MHz frequency offset, and the maximum of –40 dBr and –50 dBm/MHz at 15 MHz frequency offset and above. The transmitted spectral density of the transmitted signal shall fall within the spectral mask, as shown in Figure 18-14. The measurements shall be made using a 100 kHz resolution bandwidth and a 30 kHz video bandwidth.

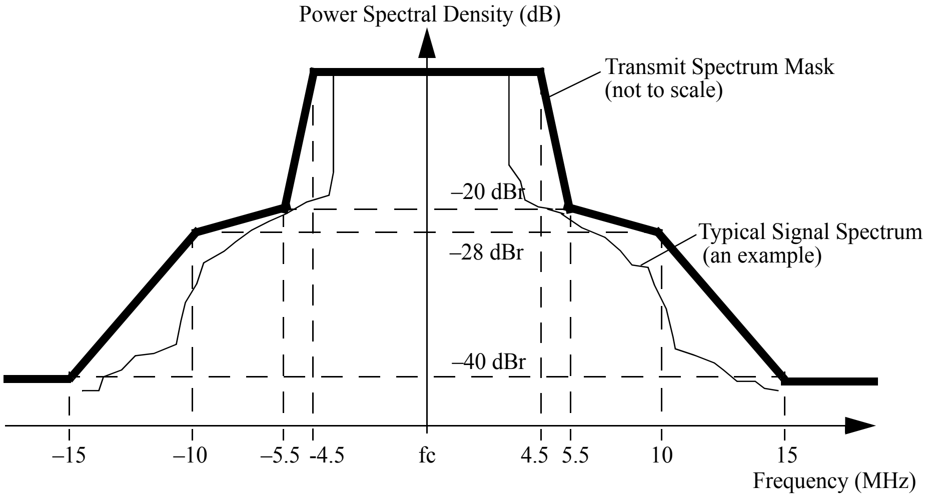


Figure 18-14—Transmit spectrum mask for 10 MHz transmission

For operation using 5 MHz channel spacing, the transmitted spectrum shall have a 0 dBr bandwidth not exceeding 4.5 MHz, –20 dBr at 2.75 MHz frequency offset, –28 dBr at 5 MHz frequency offset, and the maximum of –40 dBr and –47 dBm/MHz at 7.5 MHz frequency offset and above. The transmitted spectral density of the transmitted signal shall fall within the spectral mask, as shown in Figure 18-15. The measurements shall be made using a 100 kHz resolution bandwidth and a 30 kHz video bandwidth.

18.3.9.4 Transmission spurious

Spurious transmissions from compliant devices shall conform to national regulations.

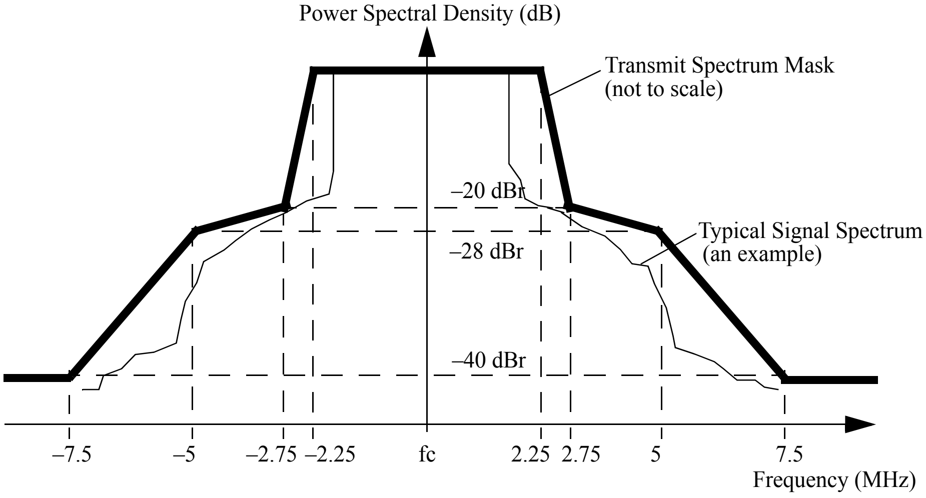


Figure 18-15—Transmit spectrum mask for 5 MHz transmission

18.3.9.5 Transmit center frequency tolerance

The transmitted center frequency tolerance shall be ±20 ppm maximum for 20 MHz and 10 MHz channels and shall be ±10 ppm maximum for 5 MHz channels. The transmit center frequency and the symbol clock frequency shall be derived from the same reference oscillator.

18.3.9.6 Symbol clock frequency tolerance

The symbol clock frequency tolerance shall be ±20 ppm maximum for 20 MHz and 10 MHz channels, and shall be ±10 ppm maximum for 5 MHz channels. The transmit center frequency and the symbol clock frequency shall be derived from the same reference oscillator.

18.3.9.7 Modulation accuracy

18.3.9.7.1 Introduction

Transmit modulation accuracy specifications are described in 18.3.9.7. The test method is described in 18.3.9.8.

18.3.9.7.2 Transmitter center frequency leakage

Certain transmitter implementations may cause leakage of the center frequency component. Such leakage (which manifests itself in a receiver as energy in the center frequency component) shall not exceed –15 dB relative to overall transmitted power or, equivalently, +2 dB relative to the average energy of the rest of the subcarriers. The data for this test shall be derived from the channel estimation phase.

18.3.9.7.3 Transmitter spectral flatness

The average energy of the constellations in each of the spectral lines –16.. –1 and +1.. +16 shall deviate no more than ± 4 dB from their average energy. The average energy of the constellations in each of the spectral lines –26.. –17 and +17.. +26 shall deviate no more than +4/–6 dB from the average energy of spectral lines –16.. –1 and +1.. +16. The data for this test shall be derived from the channel estimation step.

18.3.9.7.4 Transmitter constellation error

The relative constellation RMS error, averaged over subcarriers, OFDM frames, and packets, shall not exceed a data-rate dependent value according to Table 18-13.

Table 18-13—Allowed relative constellation error versus data rate

Relative constellation error (dB)	Modulation	Coding rate (R)
–5	BPSK	1/2
–8	BPSK	3/4
–10	QPSK	1/2
–13	QPSK	3/4
–16	16-QAM	1/2
–19	16-QAM	3/4
–22	64-QAM	2/3
–25	64-QAM	3/4

18.3.9.8 Transmit modulation accuracy test

The transmit modulation accuracy test shall be performed by instrumentation capable of converting the transmitted signal into a stream of complex samples at 20 Msample/s or more, with sufficient accuracy in terms of I/Q arm amplitude and phase balance, dc offsets, phase noise, etc. A possible embodiment of such a setup is converting the signal to a low IF with a microwave synthesizer, sampling the signal with a digital oscilloscope and decomposing it digitally into quadrature components.

The sampled signal shall be processed in a manner similar to an actual receiver, according to the following steps, or an equivalent procedure:

a) Start of frame shall be detected.

b) Transition from short sequences to channel estimation sequences shall be detected, and fine timing (with one sample resolution) shall be established.

c) Coarse and fine frequency offsets shall be estimated.

d) The packet shall be derotated according to estimated frequency offset.

e) The complex channel response coefficients shall be estimated for each of the subcarriers.

f) For each of the data OFDM symbols: transform the symbol into subcarrier received values, estimate the phase from the pilot subcarriers, derotate the subcarrier values according to estimated phase, and divide each subcarrier value with a complex estimated channel response coefficient.

g) For each data-carrying subcarrier, find the closest constellation point and compute the Euclidean distance from it.

h) Compute the RMS average of all errors in a packet. It is given by

$$Error_{RMS} = \frac{\displaystyle\sum_{i=1}^{N_f} \sqrt{\frac{\displaystyle\sum_{j=1}^{L_P}\left[\sum_{k=1}^{52}\{(I(i,j,k)-I_0(i,j,k))^2+(Q(i,j,k)-Q_0(i,j,k))^2\}\right]}{52L_P \times P_0}}}{N_f} \quad (18\text{-}28)$$

where

L_P is the length of the packet;

N_f is the number of frames for the measurement;

($I_0(i,j,k)$, $Q_0(i,j,k)$)denotes the ideal symbol point of the ith frame, jth OFDM symbol of the frame, kth subcarrier of the OFDM symbol in the complex plane;

($I(i,j,k)$, $Q(i,j,k)$)denotes the observed point of the ith frame, jth OFDM symbol of the frame, kth subcarrier of the OFDM symbol in the complex plane (see Figure 18-16);

P_0 is the average power of the constellation.

The vector error on a phase plane is shown in Figure 18-16.

The test shall be performed over at least 20 frames (N_f), and the RMS average shall be taken. The packets under test shall be at least 16 OFDM symbols long. Random data shall be used for the symbols.

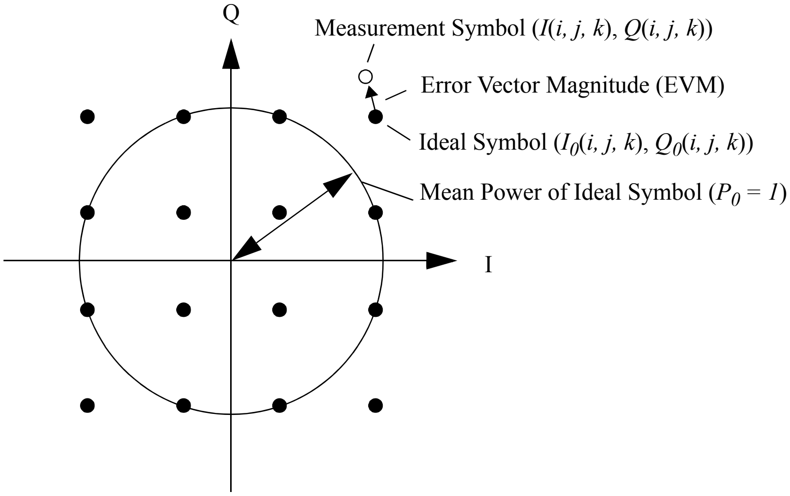

Q
Measurement Symbol ($I(i, j, k)$, $Q(i, j, k)$)
Error Vector Magnitude (EVM)
Ideal Symbol ($I_0(i, j, k)$, $Q_0(i, j, k)$)
Mean Power of Ideal Symbol ($P_0 = 1$)
I

Figure 18-16—Constellation error

18.3.9.9 Time of Departure accuracy

The Time of Departure accuracy test evaluates TIME_OF_DEPARTURE against aTxPmdTxStartRMS and aTxPmdTxStartRMS against TIME_OF_DEPARTURE_ACCURACY_TEST_THRESH as defined Annex T with the following test parameters:

— MULTICHANNEL_SAMPLING_RATE is $20 \times 10^6\left(1 + \left\lceil \frac{f_H - f_L}{20\text{ MHz}} \right\rceil\right)$ sample/s

where

f_H is the nominal center frequency in Hz of the highest channel in the channel set

f_L is the nominal center frequency in Hz of the lowest channel in the channel set, the channel set is the set of channels upon which frames providing measurements are transmitted, the channel set comprises channels uniformly spaced across $f_H - f_L \geq 50$ MHz

$\lceil x \rceil$ equals the smallest integer equal to or larger than x

— FIRST_TRANSITION_FIELD is the Short symbols.

— SECOND_TRANSITION_FIELD is the Long symbols.

— TRAINING_FIELD is the Long symbols windowed in a manner which should approximate the windowing described in 18.3.2.5 with T_{TR} = 100 ns for 20 MHz channel spacing, T_{TR} = 200 ns for 10 MHz channel spacing and T_{TR} = 400 ns for 5 MHz channel spacing.

— TIME_OF_DEPARTURE_ACCURACY_TEST_THRESH is 80 ns.

NOTE—The indicated windowing applies to the time of departure accuracy test equipment, and not the transmitter or receiver.

18.3.10 PMD receiver specifications

18.3.10.1 Introduction

The receive specifications associated with the PMD sublayer are described in 18.3.10.2 to 18.3.10.6.

18.3.10.2 Receiver minimum input sensitivity

The packet error ratio (PER) shall be 10% or less when the PSDU length is 1000 octets and the rate-dependent input level is as shown in Table 18-14. The minimum input levels are measured at the antenna connector (noise factor of 10 dB and 5 dB implementation margins are assumed).

Table 18-14—Receiver performance requirements

Modulation	Coding rate (R)	Adjacent channel rejection (dB)	Alternate adjacent channel rejection (dB)	Minimum sensitivity (dBm) (20 MHz channel spacing)	Minimum sensitivity (dBm) (10 MHz channel spacing)	Minimum sensitivity (dBm) (5 MHz channel spacing)
BPSK	1/2	16	32	–82	–85	–88
BPSK	3/4	15	31	–81	–84	–87
QPSK	1/2	13	29	–79	–82	–85
QPSK	3/4	11	27	–77	–80	–83
16-QAM	1/2	8	24	–74	–77	–80
16-QAM	3/4	4	20	–70	–73	–76
64-QAM	2/3	0	16	–66	–69	–72
64-QAM	3/4	–1	15	–65	–68	–71

18.3.10.3 Adjacent channel rejection

The adjacent channel rejection shall be measured by setting the desired signal's strength 3 dB above the rate-dependent sensitivity specified in Table 18-14 and raising the power of the interfering signal until 10% PER is caused for a PSDU length of 1000 octets. The power difference between the interfering and the desired channel is the corresponding adjacent channel rejection. The interfering signal in the adjacent channel shall be a conformant OFDM signal, unsynchronized with the signal in the channel under test. For a conformant OFDM PHY the corresponding rejection shall be no less than specified in Table 18-14.

An optional enhanced performance specification is provided for systems requiring improved immunity to out-of-channel interfering emissions. If a STA has dot11ACRType equal to 2, the adjacent channel rejection shall be no less than specified in Table 18-15. The interfering signal in the adjacent channel shall be a conformant OFDM signal, using transmit mask M (see D.2.4), unsynchronized with the signal in the channel under test. The corresponding minimum receiver sensitivities for each modulation and coding rate are the same as in Table 18-14.

NOTE—Transmit mask M is equivalent to class C (see D.2.2).

Table 18-15—Optional enhanced receiver performance requirements

Modulation	Coding rate (R)	Adjacent channel rejection (dB)	Nonadjacent channel rejection (dB)
BPSK	1/2	28	42
BPSK	3/4	27	41
QPSK	1/2	25	39
QPSK	3/4	23	37
16-QAM	1/2	20	34
16-QAM	3/4	16	30
64-QAM	2/3	12	26
64-QAM	3/4	11	25

18.3.10.4 Nonadjacent channel rejection

The nonadjacent channel rejection shall be measured by setting the desired signal's strength 3 dB above the rate-dependent sensitivity specified in Table 18-14, and raising the power of the interfering signal until a 10% PER occurs for a PSDU length of 1000 octets. The power difference between the interfering and the desired channel is the corresponding nonadjacent channel rejection. The interfering signal in the nonadjacent channel shall be a conformant OFDM signal, unsynchronized with the signal in the channel under test. For a conformed OFDM PHY, the corresponding rejection shall be no less than specified in Table 18-14.

An optional enhanced performance specification is provided for systems requiring improved immunity to out-of-channel interfering emissions. If a STA has dot11ACRType equal to 2, the nonadjacent channel rejection shall be no less than specified in Table 18-15. The interfering signal in the nonadjacent channel shall be a conformant OFDM signal, using transmit mask M (see D.2.4), unsynchronized with the signal in the channel under test. The corresponding minimum receiver sensitivities for each modulation and coding rate are the same as in Table 18-14.

18.3.10.5 Receiver maximum input level

The receiver shall provide a maximum PER of 10% at a PSDU length of 1000 octets, for a maximum input level of –30 dBm measured at the antenna for any baseband modulation.

18.3.10.6 CCA requirements

CCA shall detect a medium busy condition when the carrier sense/clear channel assessment (CS/CCA) mechanism detects a channel busy condition. For the operating classes requiring CCA-Energy Detect (CCA-ED), CCA shall also detect a medium busy condition when CCA-ED detects a channel busy condition.

The start of a valid OFDM transmission at a receive level equal to or greater than the minimum modulation and coding rate sensitivity (–82 dBm for 20 MHz channel spacing, –85 dBm for 10 MHz channel spacing, and –88 dBm for 5 MHz channel spacing) shall cause CS/CCA to indicate busy with a probability > 90% within 4 µs for 20 MHz channel spacing, 8 µs for 10 MHz channel spacing, and 16 µs for 5 MHz channel spacing. If the preamble portion was missed, the receiver shall hold the CCA signal busy for any signal 20 dB above the minimum modulation and coding rate sensitivity (–62 dBm for 20 MHz channel spacing, –65 dBm for 10 MHz channel spacing, and –68 dBm for 5 MHz channel spacing).

NOTE—CS/CCA detect time is based on finding the short sequences in the preamble, so when T_{SYM} doubles, so does CS/CCA detect time.

For improved spectrum sharing, CCA-ED is required in some bands. The behavior class indicating CCA-ED is given in Table D-2. The operating classes requiring the corresponding CCA-ED behavior class are given in E.1. A STA that is operating within an operating class that requires CCA-ED shall operate with CCA-ED. The CCA-ED shall not be required for license-exempt operation in any band.

CCA-ED shall indicate a channel busy condition when the received signal strength exceeds the CCA-ED threshold as given by dot11OFDMEDThreshold. The CCA-ED thresholds for the operating classes requiring CCA-ED are subject to the criteria in D.2.5.

NOTE—The requirement to hold the CCA signal busy for any signal 20dB above the minimum modulation and coding rate sensitivity (–62 dBm for 20 MHz channel spacing, –65 dBm for 10 MHz channel spacing, and –68 dBm for 5 MHz channel spacing) is a mandatory energy detect requirement on all Clause 18 receivers. Support for CCA-ED is an additional requirement that relates specifically to the sensitivities described in D.2.5.

18.3.10.7 Received Channel Power Indicator Measurement

The RCPI indicator is a measure of the received RF power in the selected channel for a received frame. This parameter shall be a measure by the PHY sublayer of the received RF power in the channel measured over the entire received frame or by other equivalent means that meet the specified accuracy. RCPI shall be a monotonically increasing, logarithmic function of the received power level defined in dBm. The allowed values for the RCPI parameter shall be an 8-bit value in the range from 0 to 220, with indicated values rounded to the nearest 0.5 dB as follows:

0: Power ≤ – 110 dBm

1: Power = – 109.5 dBm

2: Power = – 109.0 dBm

and so on where

$$\text{RCPI} = \text{Int}\{(\text{Power in dBm} + 110) \times 2\} \text{ for } 0 \text{ dBm} > \text{Power} > -110 \text{ dBm}$$

220: Power ≥ – 0 dBm

221–254: Reserved

255: Measurement not available

RCPI shall equal the received RF power within an accuracy of ±5 dB (95% confidence interval) within the specified dynamic range of the receiver. The received RF power shall be determined assuming a receiver noise equivalent bandwidth equal to the channel bandwidth multiplied by 1.1.

18.3.11 Transmit PLCP

The transmit PLCP is shown in Figure 18-17. In order to transmit data, the PHY-TXSTART.request primitive shall be enabled so that the PHY entity shall be in the transmit state. Further, the PHY shall be set to operate at the appropriate frequency through STA management via the PLME. Other transmit parameters, such as DATARATE and TX power, are set via the PHY-SAP with the PHY-TXSTART.request(TXVECTOR) primitive, as described in 18.2.2.

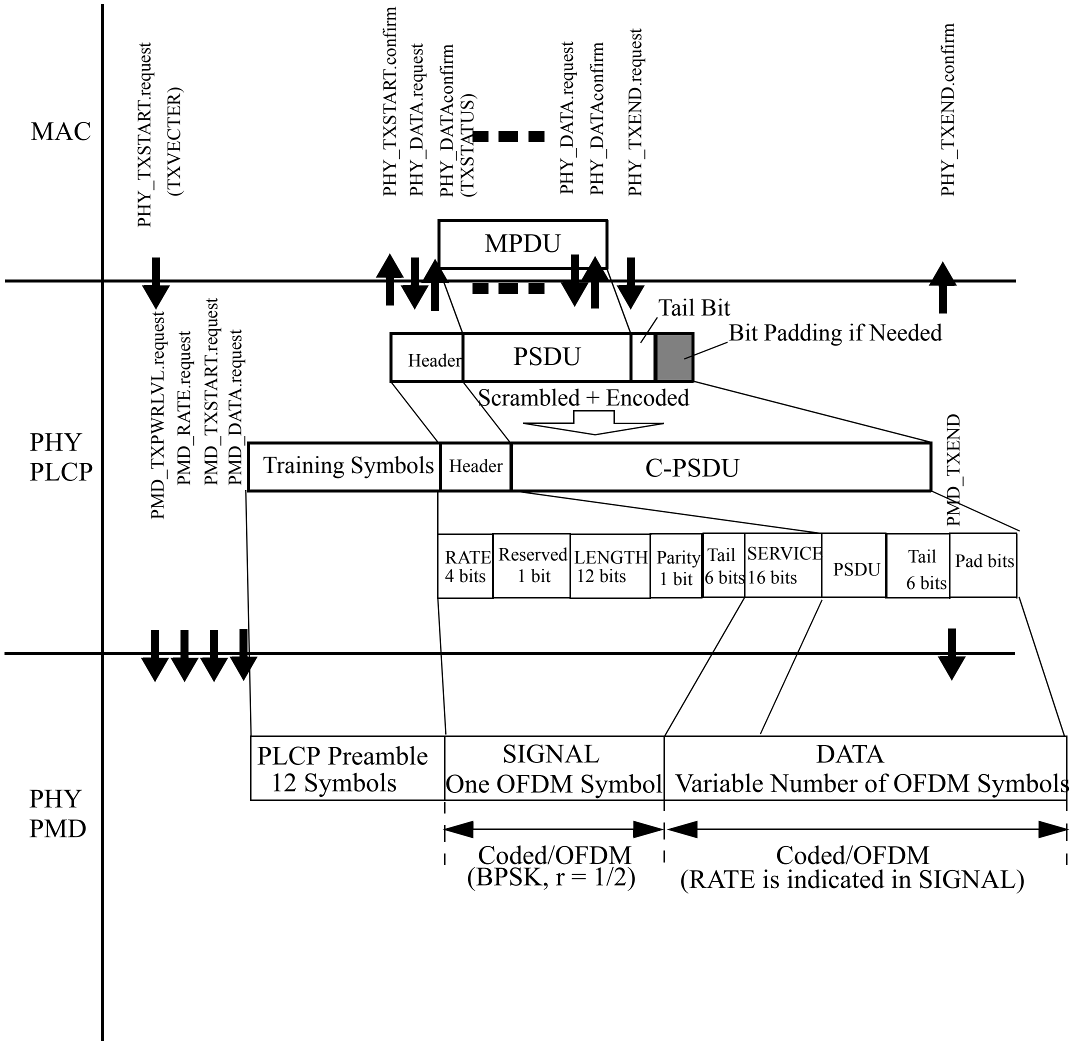


Figure 18-17—Transmit PLCP

A clear channel shall be indicated by a PHY-CCA.indication(IDLE) primitive. The MAC considers this indication before issuing the PHY-TXSTART.request primitive. Transmission of the PPDU shall be initiated after receiving the PHY-TXSTART.request(TXVECTOR) primitive. The TXVECTOR elements for the PHY-TXSTART.request primitive are the PLCP header parameters DATARATE, SERVICE, and LENGTH and the PMD parameters TXPWR_LEVEL and TIME_OF_DEPARTURE_REQUESTED.

The PLCP shall issue PMD_TXPWRLVL and PMD_RATE primitives to configure the PHY. The PLCP shall then issue a PMD_TXSTART.request primitive, and transmission of the PLCP preamble and PLCP header, based on the parameters passed in the PHY-TXSTART.request primitive, shall be immediately initiated. If dot11MgmtOptionTODImplemented and dot11MgmtOptionTODActivated are true or if dot11MgmtOptionTimingMsmtActivated is true and the TXVECTOR parameter TIME_OF_DEPARTURE_REQUESTED is true, then the PLCP shall issue a PHY_TXSTART.confirm(TXSTATUS) primitive to the MAC, forwarding the TIME_OF_DEPARTURE corresponding to the time when the first frame energy is sent by the transmitting port and the TIME_OF_DEPARTURE_ClockRate parameter within the TXSTATUS vector. If dot11MgmtOptionTimingMsmtActivated is true, then the PLCP shall forward the value of TX_START_OF_FRAME_OFFSET in the TXSTATUS vector. Once PLCP preamble transmission is started, the PHY entity shall immediately initiate PLCP header encoding then data scrambling and data encoding, where the data shall be exchanged between the MAC and the PHY through a series of PHY-DATA.request(DATA) primitives issued by the MAC, and PHY-DATA.confirm primitives issued by the PHY. The modulation rate change, if any, shall be initiated from the SERVICE field data of the PLCP header, as described in 18.3.2.

The PHY proceeds with PSDU transmission through a series of data octet transfers from the MAC. The PLCP header parameter, SERVICE, and PSDU are encoded by the convolutional encoder with the bit-stealing function described in 18.3.5.6. At the PMD layer, the data octets are sent in bit 0–7 order and presented to the PHY through PMD_DATA.request primitives. Transmission can be prematurely terminated by the MAC through the PHY-TXEND.request primitive. PHY-TXSTART shall be disabled by the issuance of the PHY-TXEND.request primitive. Normal termination occurs after the transmission of the final bit of the last PSDU octet, according to the number supplied in the OFDM PHY preamble LENGTH field.

The packet transmission shall be completed and the PHY entity shall enter the receive state (i.e., PHY-TXSTART shall be disabled). Each PHY-TXEND.request primitive is acknowledged with a PHY-TXEND.confirm primitive from the PHY. If the coded PSDU (C-PSDU) length is not a multiple of the OFDM symbol length N_{CBPS}, the PSDU is padded prior to scrambling and coding (see 18.3.5.4).

In the PMD, the GI shall be inserted in every OFDM symbol as a countermeasure against severe delay spread.

A typical state machine implementation of the transmit PLCP is provided in Figure 18-18. Requests (.request) and confirmations(.confirm) are issued once with designated states.

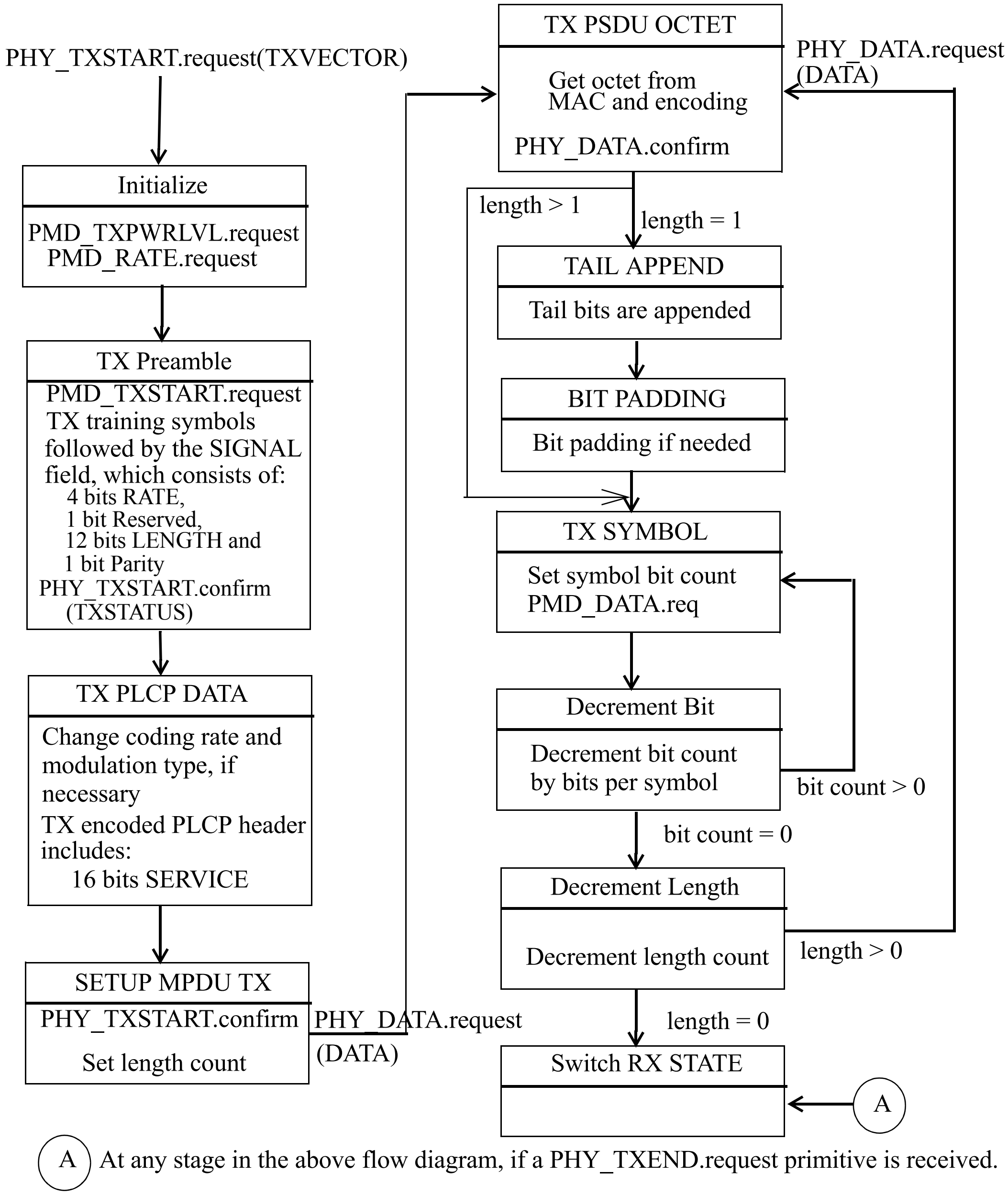


Figure 18-18—PLCP transmit state machine

18.3.12 Receive PLCP

The receive PLCP is shown in Figure 18-19. In order to receive data, the PHY-TXSTART.request primitive shall be disabled so that the PHY entity is in the receive state. Further, through STA management (via the PLME) the PHY is set to the appropriate frequency. Other receive parameters, such as RSSI, RCPI, and indicated DATARATE, may be accessed via the PHY-SAP.

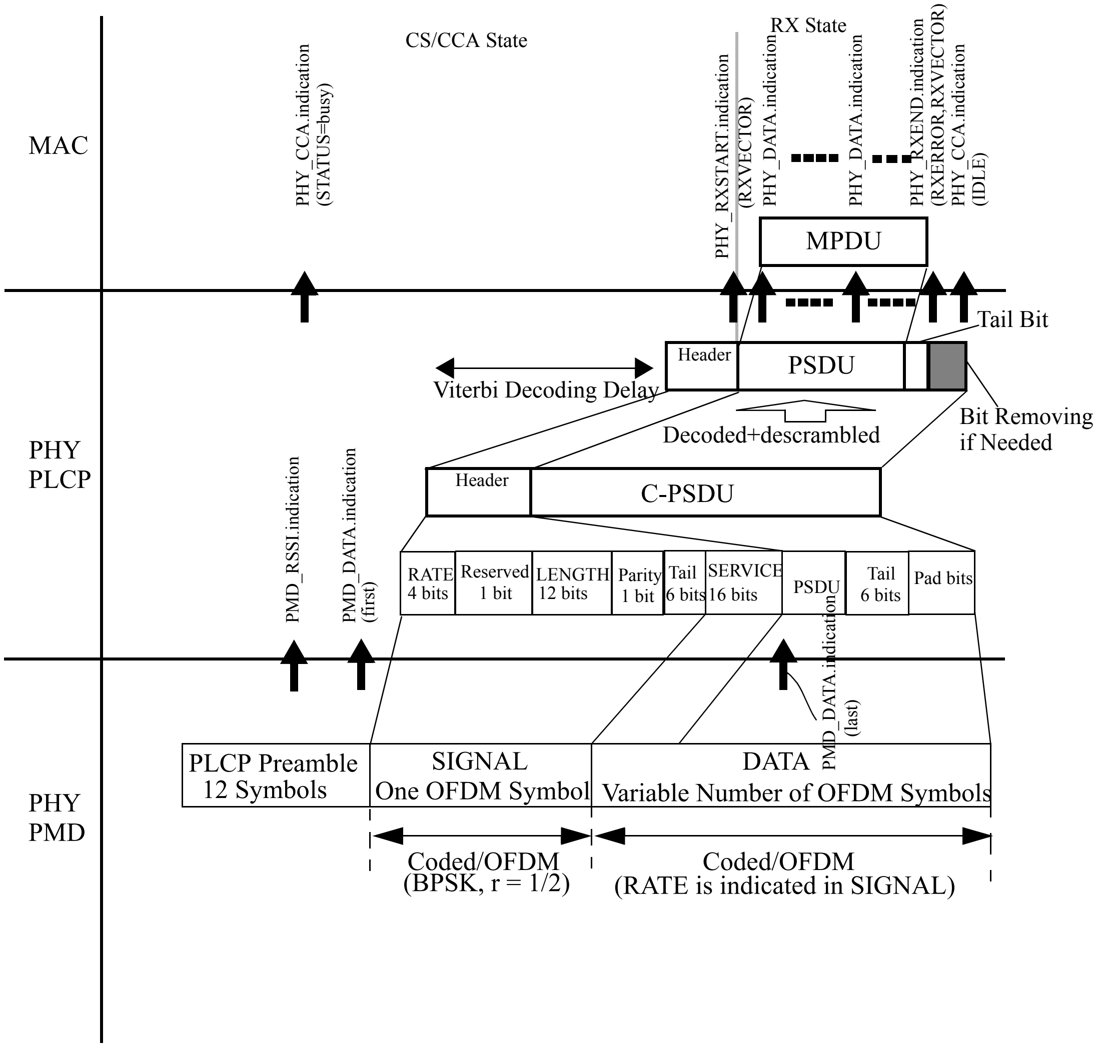


Figure 18-19—Receive PLCP

Upon receiving the transmitted PLCP preamble, a PMD_RSSI.indication primitive shall report a significant received signal strength level to the PLCP. This indicates activity to the MAC via PHY_CCA.indication primitive. A PHY_CCA.indication(BUSY) primitive shall be issued for reception of a signal prior to correct reception of the PLCP frame. The PMD primitive PMD_RSSI is issued to update the RSSI and parameter reported to the MAC.

After a PHY-CCA.indication primitive is issued, the PHY entity shall begin receiving the training symbols and searching for the SIGNAL in order to set the length of the data stream, the demodulation type, and the decoding rate. Once the SIGNAL is detected, without any errors detected by a single parity (even), FEC decode shall be initiated and the PLCP IEEE 802.11 SERVICE fields and data shall be received, decoded (a Viterbi decoder is recommended), and checked by ITU-T CRC-32. If the FCS by the ITU-T CRC-32 check fails, the PHY receiver shall return to the RX IDLE state, as depicted in Figure 18-19. Should the status of CCA return to the IDLE state during reception prior to completion of the full PLCP processing, the PHY receiver shall return to the RX IDLE state.

If the PLCP header reception is successful (and the SIGNAL field is completely recognizable and supported), a PHY-RXSTART.indication(RXVECTOR) primitive shall be issued. If dot11MgmtOptionTimingMsmtActivated is true, the PLCP shall do the following:

- Complete receiving the PLCP header and verify the validity of the PLCP Header.
- If the PLCP header reception is successful (and the SIGNAL field is completely recognizable and supported), a PHY-RXSTART.indication(RXVECTOR) primitive shall be issued and RX_START_OF_FRAME_OFFSET parameter within the RXVECTOR shall be forwarded (see 18.2.3).

NOTE—The RX_START_OF_FRAME_OFFSET value is used as described in 6.3.57 in order to estimate when the start of the preamble for the incoming frame was detected on the medium at the receive antenna port.

The RXVECTOR associated with this primitive includes the SIGNAL field, the SERVICE field, the PSDU length in octets, and the RSSI. Also, in this case, the OFDM PHY shall ensure that the CCA indicates a busy medium for the intended duration of the transmitted frame, as indicated by the LENGTH field.

The received PSDU bits are assembled into octets, decoded, and presented to the MAC using a series of PHY-DATA.indication(DATA) primitive exchanges. The rate change indicated in the IEEE 802.11 SIGNAL field shall be initiated from the SERVICE field data of the PLCP header, as described in 18.3.2. The PHY shall proceed with PSDU reception. After the reception of the final bit of the last PSDU octet indicated by the PLCP preamble LENGTH field, the receiver shall be returned to the RX IDLE state, as shown in Figure 18-19. A PHY-RXEND.indication(NoError) primitive shall be issued.

In the event that a change in the RSSI causes the status of the CCA to return to the IDLE state before the complete reception of the PSDU, as indicated by the PLCP LENGTH field, the error condition shall be reported to the MAC using a PHY-RXEND.indication(CarrierLost) primitive. The OFDM PHY shall ensure that the CCA indicates a busy medium for the intended duration of the transmitted packet.

If the indicated rate in the SIGNAL field is not receivable, a PHY-RXSTART.indication primitive shall not be issued. The PHY shall indicate the error condition using a PHY-RXEND.indication(UnsupportedRate) primitive and hold CCA busy for the calculated duration of the PPDU. If the PLCP header is receivable, but the parity check of the PLCP header is not valid, a PHY-RXSTART.indication primitive shall not be issued. The PHY shall indicate the error condition using a PHY-RXEND.indication(FormatViolation) primitive.

Any data received after the indicated data length are considered pad bits (to fill out an OFDM symbol) and should be discarded.

A typical state machine implementation of the receive PLCP is given in Figure 18-20.

18.4 OFDM PLME

18.4.1 PLME_SAP sublayer management primitives

Table 18-16 lists the MIB attributes that may be accessed by the PHY entities and the intralayer of higher level LMEs. These attributes are accessed via the PLME-GET, PLME-SET, PLME-RESET, and PLME-CHARACTERISTICS primitives defined in 6.5.

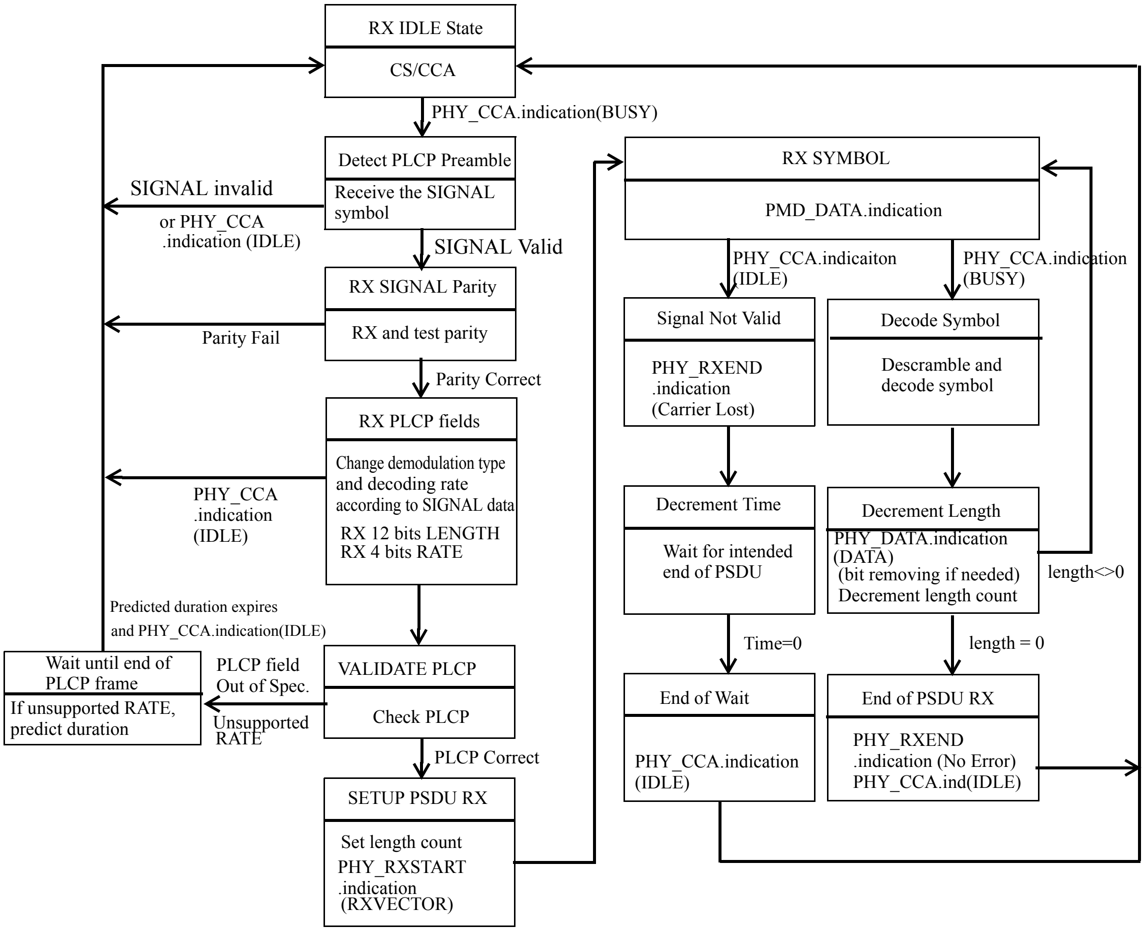

RX IDLE State
CS/CCA
PHY_CCA.indication(BUSY)
Detect PLCP Preamble
Receive the SIGNAL symbol
SIGNAL invalid
or PHY_CCA .indication (IDLE)
SIGNAL Valid
RX SIGNAL Parity
RX and test parity
Parity Fail
Parity Correct
RX PLCP fields
Change demodulation type and decoding rate according to SIGNAL data
RX 12 bits LENGTH
RX 4 bits RATE
PHY_CCA .indication (IDLE)
Predicted duration expires and PHY_CCA.indication(IDLE)
Wait until end of PLCP frame
If unsupported RATE, predict duration
PLCP field Out of Spec.
Unsupported RATE
VALIDATE PLCP
Check PLCP
PLCP Correct
SETUP PSDU RX
Set length count
PHY_RXSTART .indication (RXVECTOR)
RX SYMBOL
PMD_DATA.indication
PHY_CCA.indicaiton (IDLE)
PHY_CCA.indication (BUSY)
Signal Not Valid
PHY_RXEND .indication (Carrier Lost)
Decode Symbol
Descramble and decode symbol
Decrement Time
Wait for intended end of PSDU
Decrement Length
PHY_DATA.indication (DATA) (bit removing if needed) Decrement length count
length<>0
Time=0
length = 0
End of Wait
PHY_CCA.indication (IDLE)
End of PSDU RX
PHY_RXEND .indication (No Error) PHY_CCA.ind(IDLE)

Figure 18-20—PLCP receive state machine

18.4.2 OFDM PHY MIB

All OFDM PHY MIB attributes are defined in Annex C, with specific values defined in Table 18-16. The column titled “Operational semantics” in Table 18-16 contains two types: static and dynamic. Static MIB attributes are fixed and cannot be modified for a given PHY implementation. Dynamic MIB attributes can be modified by some management entity.

Table 18-16—MIB attribute default values/ranges

Managed object	Default value/range	Operational semantics
dot11 PHY Operation Table		
dot11PHYtype	OFDM-5. (04)	Static
dot11CurrentRegDomain	Implementation dependent	Dynamic
dot11CurrentFrequencyBand	Implementation dependent	Dynamic

Table 18-16—MIB attribute default values/ranges *(continued)*

Managed object	Default value/range	Operational semantics
dot11 PHY Antenna Table		
dot11CurrentTxAntenna	Implementation dependent	Dynamic
dot11DiversitySupportImplemented	Implementation dependent	Static
dot11CurrentRxAntenna	Implementation dependent	Dynamic
dot11 PHY Tx Power Table		
dot11NumberSupportedPowerLevelsImplemented	Implementation dependent	Static
dot11TxPowerLevel1	Implementation dependent	Static
dot11TxPowerLevel2	Implementation dependent	Static
dot11TxPowerLevel3	Implementation dependent	Static
dot11TxPowerLevel4	Implementation dependent	Static
dot11TxPowerLevel5	Implementation dependent	Static
dot11TxPowerLevel6	Implementation dependent	Static
dot11TxPowerLevel7	Implementation dependent	Static
dot11TxPowerLevel8	Implementation dependent	Static
dot11CurrentTxPowerLevel	Implementation dependent	Dynamic
dot11 Reg Domains Supported Table		
dot11RegDomainsImplementedValue	Implementation dependent	Static
dot11FrequencyBandsSupported	Implementation dependent	Static
dot11 PHY Antennas List Table		
dot11SupportedTxAntenna	Implementation dependent	Static
dot11SupportedRxAntenna	Implementation dependent	Static
dot11DiversitySelectionRx	Implementation dependent	Dynamic
dot11 Supported Data Rates Tx Table		
dot11ImplementedDataRatesTxValue	6, 9, 12, 18, 24, 36, 48, and 54 Mb/s for 20 MHz channel spacing (Mandatory rates: 6, 12, and 24) 3, 4.5, 6, 9, 12, 18, 24, and 27 Mb/s for 10 MHz channel spacing (Mandatory rates: 3, 6, and 12) 1.5, 2.25, 3, 4.5, 6, 9, 12, and 13.5 Mb/s for 5 MHz channel spacing (Mandatory rates: 1.5, 3, and 6)	Static

Table 18-16—MIB attribute default values/ranges *(continued)*

Managed object	Default value/range	Operational semantics
dot11 Supported Data Rates Rx Table		
dot11ImplementedDataRatesRxValue	6, 9, 12, 18, 24, 36, 48, and 54 Mb/s for 20 MHz channel spacing (Mandatory rates: 6, 12, and 24) 3, 4.5, 6, 9, 12, 18, 24, and 27 Mb/s for 10 MHz channel spacing (Mandatory rates: 3, 6, and 12) 1.5, 2.25, 3, 4.5, 6, 9, 12, and 13.5 Mb/s for 5 MHz channel spacing (Mandatory rates: 1.5, 3, and 6)	Static
dot11 PHY OFDM Table		
dot11CurrentFrequency	Implementation dependent	Dynamic
dot11TIThreshold	Implementation dependent	Dynamic
dot11ChannelStartingFactor	Implementation dependent	Dynamic
dot11OFDMEDThreshold	Implementation dependent	Dynamic
dot11ACRType	Implementation dependent	Dynamic

18.4.3 OFDM TXTIME calculation

The value of the TXTIME parameter returned by the PLME-TXTIME.confirm primitive shall be calculated according to the following equation:

$$\mathrm{TXTIME} = T_{PREAMBLE} + T_{SIGNAL} + T_{SYM} \times N_{SYM} \tag{18-29}$$

where

$T_{PREAMBLE}$ is defined in Table 18-5
T_{SIGNAL} is defined in Table 18-5
T_{SYM} is defined in Table 18-5
N_{SYM} is given by Equation (18-11).

18.4.4 OFDM PHY characteristics

The static OFDM PHY characteristics, provided through the PLME-CHARACTERISTICS service primitive, are shown in Table 18-17. The definitions for these characteristics are given in 6.5.

Table 18-17—OFDM PHY characteristics

Characteristics	Value (20 MHz channel spacing)	Value (10 MHz channel spacing)	Value (5 MHz channel spacing)
aSlotTime	9 µs	13 µs	21 µs
aSIFSTime	16 µs	32 µs	64 µs
aCCATime	< 4 µs	< 8 µs	< 16 µs
aPHY-RX-START-Delay	25 µs	49 µs	97 µs
aRxTxTurnaroundTime	< 2 µs	< 2 µs	< 2 µs
aTxPLCPDelay	Implementation dependent as long as the requirements of aRxTxTurnaroundTime are met.	Implementation dependent as long as the requirements of aRxTxTurnaroundTime are met.	Implementation dependent as long as the requirements of aRxTxTurnaroundTime are met.
aRxPLCPDelay	Implementation dependent as long as the requirements of aSIFSTime and aCCATime are met.	Implementation dependent as long as the requirements of aSIFSTime and aCCATime are met.	Implementation dependent as long as the requirements of aSIFSTime and aCCATime are met.
aRxTxSwitchTime	<< 1 µs	<< 1 µs	<< 1 µs
aTxRampOnTime	Implementation dependent as long as the requirements of aRxTxTurnaroundTime are met.	Implementation dependent as long as the requirements of aRxTxTurnaroundTime are met.	Implementation dependent as long as the requirements of aRxTxTurnaroundTime are met.
aTxRampOffTime	Implementation dependent as long as the requirements of aSIFSTime are met.	Implementation dependent as long as the requirements of aSIFSTime are met.	Implementation dependent as long as the requirements of aSIFSTime are met.
aTxRFDelay	Implementation dependent as long as the requirements of aRxTxTurnaroundTime are met.	Implementation dependent as long as the requirements of aRxTxTurnaroundTime are met.	Implementation dependent as long as the requirements of aRxTxTurnaroundTime are met.
aRxRFDelay	Implementation dependent as long as the requirements of aSIFSTime and aCCATime are met.	Implementation dependent as long as the requirements of aSIFSTime and aCCATime are met.	Implementation dependent as long as the requirements of aSIFSTime and aCCATime are met.
aAirPropagationTime	<< 1 µs	<< 1 µs	<< 1 µs
aMACProcessingDelay	< 2 µs	< 2 µs	< 2 µs
aPreambleLength	16 µs	32 µs	64 µs
aPLCPHeaderLength	4 µs	8 µs	16 µs
aMPDUMaxLength	4095	4095	4095
aCWmin	15	15	15
aCWmax	1023	1023	1023

18.5 OFDM PMD sublayer

18.5.1 Scope and field of application

Subclause 18.5 describes the PMD services provided to the PLCP for the OFDM PHY. Also defined in 18.5 are the functional, electrical, and RF characteristics required for interoperability of implementations conforming to this specification. The relationship of this specification to the entire OFDM PHY is shown in Figure 18-21.

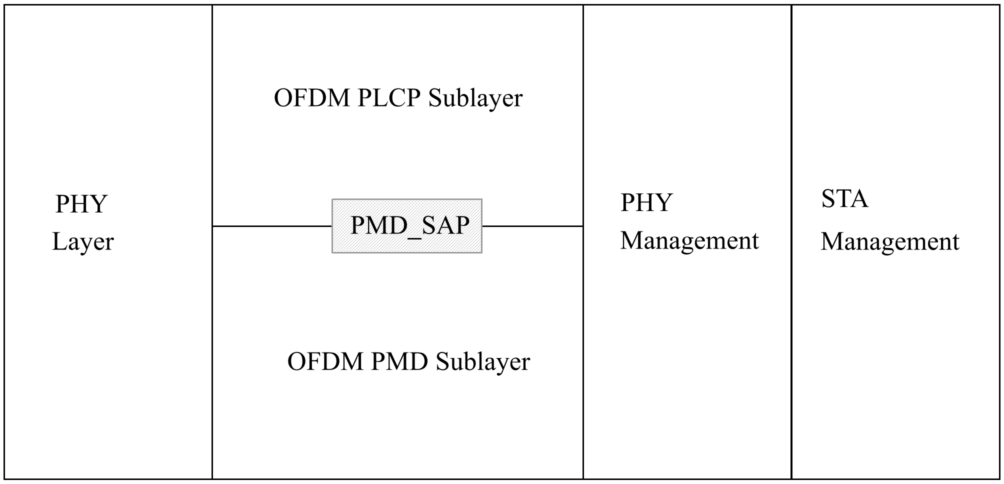


Figure 18-21—PMD layer reference model

18.5.2 Overview of service

The OFDM PMD sublayer accepts PLCP sublayer service primitives and provides the actual means by which data are transmitted or received from the medium. The combined function of the OFDM PMD sublayer primitives and parameters for the receive function results in a data stream, timing information, and associated receive signal parameters being delivered to the PLCP sublayer. A similar functionality shall be provided for data transmission.

18.5.3 Overview of interactions

The primitives associated with the IEEE 802.11 PLCP sublayer to the OFDM PMD fall into two basic categories:

a) Service primitives that support PLCP peer-to-peer interactions;
b) Service primitives that have local significance and support sublayer-to-sublayer interactions.

18.5.4 Basic service and options

18.5.4.1 General

All of the service primitives described in 18.5.4 are considered mandatory, unless otherwise specified.

18.5.4.2 PMD_SAP peer-to-peer service primitives

Table 18-18 indicates the primitives for peer-to-peer interactions.

Table 18-18—PMD_SAP peer-to-peer service primitives

Primitive	Request	Indicate	Confirm	Response
PMD_DATA	X	X	—	—

18.5.4.3 PMD_SAP sublayer-to-sublayer service primitives

Table 18-19 indicates the primitives for sublayer-to-sublayer interactions.

Table 18-19—PMD_SAP sublayer-to-sublayer service primitives

Primitive	Request	Indicate	Confirm	Response
PMD_TXSTART	X	—	—	—
PMD_TXEND	X	—	—	—
PMD_TXPWRLVL	X	—	—	—
PMD_RATE	X	—	—	—
PMD_RSSI	—	X	—	—
PMD_RCPI	—	X	—	—

18.5.4.4 PMD_SAP service primitive parameters

Table 18-20 shows the parameters used by one or more of the PMD_SAP service primitives.

Table 18-20—List of parameters for the PMD primitives

Parameter	Associated primitive	Value (20 MHz channel spacing)	Value (10 MHz channel spacing)	Value (5 MHz channel spacing)
TXD_UNIT	PMD_DATA.request	One(1), Zero(0): one OFDM symbol value	One(1),Zero(0): one OFDM symbol value	One(1),Zero(0): one OFDM symbol value
RXD_UNIT	PMD_DATA. indication	One(1), Zero(0): one OFDM symbol value	One(1),Zero(0): one OFDM symbol value	One(1),Zero(0): one OFDM symbol value
TXPWR_LEVEL	PMD_TXPWRLVL. request	1–8 (max of 8 levels)	1–8 (max of 8 levels)	1–8 (max of 8 levels)
RATE	PMD_RATE.request	12 Mb/s (for BPSK) 24 Mb/s (for QPSK) 48 Mb/s (for 16-QAM) 72 Mb/s (for 64-QAM)	6 Mb/s (for BPSK) 12 Mb/s (for QPSK) 24 Mb/s (for 16-QAM) 36 Mb/s (for 64-QAM)	3 Mb/s (for BPSK) 6 Mb/s (for QPSK) 12Mb/s (for 16-QAM) 18 Mb/s (for 64-QAM)

Table 18-20—List of parameters for the PMD primitives *(continued)*

Parameter	Associated primitive	Value (20 MHz channel spacing)	Value (10 MHz channel spacing)	Value (5 MHz channel spacing)
RSSI	PMD_RSSI. indication	0–8 bits of RSSI	0–8 bits of RSSI	0–8 bits of RSSI
RCPI	PMD_RCPI. indication	0–255	0–255	0–255

18.5.5 PMD_SAP detailed service specification

18.5.5.1 Introduction

Subclause 18.5.5 describes the services provided by each PMD primitive.

18.5.5.2 PMD_DATA.request

18.5.5.2.1 Function

This primitive defines the transfer of data from the PLCP sublayer to the PMD entity.

18.5.5.2.2 Semantics of the service primitive

This primitive shall provide the following parameter:

```
  PMD_DATA.request(
                              TXD_UNIT
                              )
```

The TXD_UNIT parameter shall be the n-bit combination of 0 and 1 for one symbol of OFDM modulation. If the length of a coded MPDU (C-MPDU) is shorter than n bits, 0 bits are added to form an OFDM symbol. This parameter represents a single block of data which, in turn, shall be used by the PHY to be encoded into an OFDM transmitted symbol.

18.5.5.2.3 When generated

This primitive shall be generated by the PLCP sublayer to request transmission of one OFDM symbol. The data clock for this primitive shall be supplied by the PMD layer based on the OFDM symbol clock.

18.5.5.2.4 Effect of receipt

The PMD performs transmission of the data.

18.5.5.3 PMD_DATA.indication

18.5.5.3.1 Function

This primitive defines the transfer of data from the PMD entity to the PLCP sublayer.

18.5.5.3.2 Semantics of the service primitive

This primitive shall provide the following parameter:

```
PMD_DATA.indication(
                                RXD_UNIT
                                )
```

The RXD_UNIT parameter shall be 0 or 1, and shall represent either a signal field bit or a data field bit after the decoding of the convolutional code by the PMD entity.

18.5.5.3.3 When generated

This primitive, generated by the PMD entity, forwards received data to the PLCP sublayer. The data clock for this primitive shall be supplied by the PMD layer based on the OFDM symbol clock.

18.5.5.3.4 Effect of receipt

The PLCP sublayer interprets the bits that are recovered as part of the PLCP or passes the data to the MAC sublayer as part of the MPDU.

18.5.5.4 PMD_TXSTART.request

18.5.5.4.1 Function

This primitive, generated by the PHY PLCP sublayer, initiates PPDU transmission by the PMD layer.

18.5.5.4.2 Semantics of the service primitive

The semantics of this primitive are as follows:

```
PMD_TXSTART.request
```

18.5.5.4.3 When generated

This primitive shall be generated by the PLCP sublayer to initiate the PMD layer transmission of the PPDU. The PHY-TXSTART.request primitive shall be provided to the PLCP sublayer prior to issuing the PMD_TXSTART command.

18.5.5.4.4 Effect of receipt

PMD_TXSTART initiates transmission of a PPDU by the PMD sublayer.

18.5.5.5 PMD_TXEND.request

18.5.5.5.1 Function

This primitive, generated by the PHY PLCP sublayer, ends PPDU transmission by the PMD layer.

18.5.5.5.2 Semantics of the service primitive

The semantics of this primitive are as follows:

```
PMD_TXEND.request
```

18.5.5.5.3 When generated

This primitive shall be generated by the PLCP sublayer to terminate the PMD layer transmission of the PPDU.

18.5.5.5.4 Effect of receipt

PMD_TXEND terminates transmission of a PPDU by the PMD sublayer.

18.5.5.6 PMD_TXPWRLVL.request

18.5.5.6.1 Function

This primitive, generated by the PHY PLCP sublayer, selects the power level used by the PHY for transmission.

18.5.5.6.2 Semantics of the service primitive

This primitive shall provide the following parameter:

```
PMD_TXPWRLVL.request(
                        TXPWR_LEVEL
                        )
```

TXPWR_LEVEL selects which of the transmit power levels should be used for the current packet transmission. The number of available power levels shall be determined by the MIB parameter aNumberSupportedPowerLevels. See 18.3.9.2 for further information on the OFDM PHY power level control capabilities.

18.5.5.6.3 When generated

This primitive shall be generated by the PLCP sublayer to select a specific transmit power. This primitive shall be applied prior to setting PMD_TXSTART into the transmit state.

18.5.5.6.4 Effect of receipt

PMD_TXPWRLVL immediately sets the transmit power level to that given by TXPWR_LEVEL.

18.5.5.7 PMD_RATE.request

18.5.5.7.1 Function

This primitive, generated by the PHY PLCP sublayer, selects the modulation rate that shall be used by the OFDM PHY for transmission.

18.5.5.7.2 Semantics of the service primitive

This primitive shall provide the following parameter:

```
PMD_RATE.request(
                        RATE
                        )
```

RATE selects which of the OFDM PHY data rates shall be used for MPDU transmission. See 18.3.8.7 for further information on the OFDM PHY modulation rates. The OFDM PHY rate change capability is described in detail in 18.3.7.

18.5.5.7.3 When generated

This primitive shall be generated by the PLCP sublayer to change or set the current OFDM PHY modulation rate used for the MPDU portion of a PPDU.

18.5.5.7.4 Effect of receipt

The receipt of PMD_RATE selects the rate that shall be used for all subsequent MPDU transmissions. This rate shall be used for transmission only. The OFDM PHY shall still be capable of receiving all the required OFDM PHY modulation rates.

18.5.5.8 PMD_RSSI.indication

18.5.5.8.1 Function

This primitive, generated by the PMD sublayer, provides the receive signal strength to the PLCP and MAC entity.

18.5.5.8.2 Semantics of the service primitive

This primitive shall provide the following parameter:

```
PMD_RSSI.indication(
                        RSSI
                        )
```

The RSSI shall be a measure of the RF energy received by the OFDM PHY. RSSIs of up to 8 bits (256 levels) are supported.

18.5.5.8.3 When generated

This primitive shall be generated by the PMD when the OFDM PHY is in the receive state. It shall be available continuously to the PLCP which, in turn, shall provide the parameter to the MAC entity.

18.5.5.8.4 Effect of receipt

This parameter shall be provided to the PLCP layer for information only. The RSSI may be used as part of a CCA scheme.

18.5.5.9 PMD_RCPI.indication

18.5.5.9.1 Function

This primitive, generated by the PMD sublayer, provides the RCPI to the PLCP and MAC entity.

18.5.5.9.2 Semantics of the service primitive

The primitive shall provide the following parameter:

```
PMD_RCPI.indication(
                        RCPI
                        )
```

The RCPI shall be a measure of the channel power received by the OFDM PHY. RCPI indications of 8 bits are supported, as defined in 18.3.10.7.

18.5.5.9.3 When generated

This primitive shall be generated by the PMD when the OFDM PHY is in the receive state. It shall be continuously available to the PLCP, which in turn provides the parameter to the MAC entity.

18.5.5.9.4 Effect of receipt

This parameter shall be provided to the PLCP layer for information only. The RCPI may be used in conjunction with RSSI to measure input signal quality.

19. Extended Rate PHY (ERP) specification

19.1 Overview

19.1.1 General

This clause specifies further rate extension of the PHY for the DSSS system of Clause 16 and the extensions of Clause 17. Hereinafter the PHY defined in this clause is known as the ERP. This PHY operates in the 2.4 GHz ISM band.

19.1.2 Introduction

The ERP builds on the payload data rates of 1 and 2 Mb/s, as described in Clause 16, that use DSSS modulation and builds on the payload data rates of 1, 2, 5.5, and 11 Mb/s, as described in Clause 17, that use DSSS, CCK, and optional PBCC modulations. The ERP draws from Clause 18 to provide additional payload data rates of 6, 9, 12, 18, 24, 36, 48, and 54 Mb/s. Of these rates, transmission and reception capability for 1, 2, 5.5, 6, 11, 12, and 24 Mb/s data rates is mandatory.

Two additional optional ERP-PBCC modulation modes with payload data rates of 22 and 33 Mb/s are defined. An ERP-PBCC STA may implement 22 Mb/s alone or 22 and 33 Mb/s. An optional modulation mode known as DSSS-OFDM is also incorporated with payload data rates of 6, 9, 12, 18, 24, 36, 48, and 54 Mb/s.

The ERP-PBCC option is obsolete. Consequently, this option may be removed in a later revision of the standard.

The use of the DSSS-OFDM option is deprecated, and this option may be removed in a later revision of the standard.

19.1.3 Operational modes

The radio portion of all Clause 19-compliant ERP systems implements all mandatory modes of Clause 18 and Clause 17, except it uses the 2.4 GHz frequency band and channelization plan specified in 17.4.6. The ERP has the capability to decode all Clause 16 and Clause 17 PLCPs and all ERP-OFDM PLCPs. In addition, it is mandatory that all ERP-compliant equipment be capable of sending and receiving the short preamble that is (and remains) optional for Clause 17 PHYs.

The ERP has the capability to detect ERP and Clause 17 preambles whenever a CCA is requested. Because protection mechanisms are not required in all cases, the ERP CCA mechanisms for all preamble types shall be active at all times.

An ERP BSS is capable of operating in any combination of available ERP modes (Clause 19 PHYs) and NonERP modes (Clause 16 or Clause 17 PHYs). For example, a BSS could operate in an ERP-OFDM-only mode, a mixed mode of ERP-OFDM and ERP-DSSS/CCK, or a mixed mode of ERP-DSSS/CCK and NonERP. When options are enabled, combinations are also allowed.

The changes to other parts of this standard required to implement the ERP are summarized as follows:

a) ERP-DSSS/CCK
    1) The PHY uses the capabilities of Clause 17 with the following exceptions:
        i) Support of the short PLCP PPDU header format capability of 17.2.2.3 is mandatory.
        ii) CCA (see 17.4.8.5) has a mechanism that detects all mandatory Clause 19 sync symbols.
        iii) The maximum input signal level (see 17.4.8.3) is –20 dBm.

iv) Locking the transmit center frequency and the symbol clock frequency to the same reference oscillator is mandatory.

b) ERP-OFDM

1) The PHY uses the capabilities of Clause 18 with the following exceptions:

i) The frequency plan is in accordance with 17.4.6.2 and 17.4.6.3 instead of 18.3.8.4.

ii) CCA has a mechanism that detects all mandatory Clause 19 sync symbols.

iii) The frequency accuracy (see 18.3.9.5 and 18.3.9.6) is ±25 PPM.

iv) The maximum input signal level (see 18.3.10.5) is –20 dBm.

v) The slot time is 20 µs in accordance with 17.3.3, except that an optional 9 µs slot time may be used when the BSS consists of only ERP STAs.

vi) SIFS time is 10 µs in accordance with 17.3.3. See 19.3.2.4 for more detail.

c) ERP-PBCC (Optional)

1) This is a single carrier modulation scheme that encodes the payload using a 256-state packet binary convolutional code. These are extensions to the PBCC modulation in Clause 17. ERP-PBCC modes with payload data rates of 22 Mb/s and 33 Mb/s are defined in 19.6.

d) DSSS-OFDM (Optional)

1) This is a hybrid modulation combining a DSSS preamble and header with an OFDM payload transmission. DSSS-OFDM modes with payload data rates of 6, 9, 12, 18, 24, 36, 48, and 54 Mb/s are defined in 19.7.

2) If the optional DSSS-OFDM mode is used, the supported rates in that mode are the same as the ERP-OFDM supported rates.

The 2.4 GHz ISM band is a shared medium, and coexistence with other devices such as Clause 16 and Clause 17 STAs is an important issue for maintaining high performance in Clause 19 (ERP) STAs. The ERP modulations (ERP-OFDM, ERP-PBCC, and DSSS-OFDM) have been designed to coexist with existing Clause 16 and Clause 17 STAs. This coexistence is achieved by several means, including virtual CS (RTS/CTS or CTS-to-self), CSMA/CA protocols, and MSDU fragmentation.

19.1.4 Scope

This clause specifies the ERP entity and the deviations from earlier clauses to accommodate it. It is organized by reference to the relevant earlier clauses to avoid excessive duplication.

The ERP consists of the following two protocol functions:

a) A physical layer convergence function that adapts the capabilities of the PMD system to the PHY service available. This function is supported by the PLCP, which defines a method for mapping the MPDUs into a framing format suitable for sending and receiving user data and management information between two or more STAs using the associated PMD system. The PHY exchanges PPDUs that contain PSDUs. The MAC uses the PHY service, so each MPDU corresponds to a PSDU that is carried in a PPDU.

b) A PMD system, whose function defines the characteristics and method of transmitting and receiving data through a WM between two or more STAs; each using the ERP.

19.1.5 ERP functions

The architecture of the ERP is depicted in the ISO/IEC basic reference model shown in Figure 17-10 of 17.4.1. The ERP contains three functional entities: the PMD function, the PLCP, and the layer management function.

The ERP service is provided to the MAC through the PHY service primitives described in Clause 7. Interoperability is addressed by use of the CS mechanism specified in 9.3.2.1 and the protection mechanism in 9.23. This mechanism allows NonERP STAs to know of ERP traffic that they cannot demodulate so that they may defer the medium to that traffic.

19.2 PHY-specific service parameter list

The architecture of the IEEE 802.11 MAC is intended to be PHY independent. Some PHY implementations require PHY-dependent MAC state machines running in the MAC sublayer in order to meet certain PMD requirements. The PHY-dependent MAC state machine resides in a sublayer defined as the MLME. In certain PMD implementations, the MLME may need to interact with the PLME as part of the normal PHY SAP primitives. These interactions are defined by the PLME parameter list currently defined in the PHY service primitives as TXVECTOR, TXSTATUS, and RXVECTOR. The list of these parameters and the values they may represent are defined in the specific PHY specifications for each PMD. This subclause addresses the TXVECTOR, TXSTATUS, and RXVECTOR for the ERP. The service parameters for TXVECTOR, TXSTATUS, and RXVECTOR shall follow 18.2.2, 18.2.4, and 18.2.3, respectively.

Several service primitives include a parameter vector. DATARATE and LENGTH are described in 7.3.4.5. The remaining parameters are considered to be management parameters and are specific to this PHY.

The parameters in Table 19-1 are defined as part of the TXVECTOR parameter list in the PHY-TXSTART.request and PLME_TXTIME.request primitives.

Table 19-1—TXVECTOR parameters

Parameter	Value
DATARATE	The rate used to transmit the PSDU in Mb/s. Allowed value depends on value of MODULATION parameter: ERP-DSSS: 1 and 2 ERP-CCK: 5.5 and 11 ERP-OFDM: 6, 9, 12, 18, 24, 36, 48, and 54 ERP-PBCC: 5.5, 11, 22, and 33 DSSS-OFDM: 6, 9, 12, 18, 24, 36, 48, and 54
LENGTH	The length of the PSDU in octets. Range: 1–4095
PREAMBLE_TYPE	The preamble used for the transmission of the PPDU. Enumerated type for which the allowed value depends on value of MODULATION parameter: ERP-OFDM: null ERP-DSSS, ERP-CCK, ERP-PBCC, DSSS-OFDM: SHORTPREAMBLE, LONGPREAMBLE
MODULATION	The modulation used for the transmission of this PSDU. Enumerated type: ERP-DSSS, ERP-CCK, ERP-OFDM, ERP-PBCC, DSSS-OFDM
SERVICE	The scrambler initialization vector. When the modulation format selected is ERP-OFDM or DSSS-OFDM, seven null bits are used for scrambler initialization as described in 18.3.5.2. The remaining bits are reserved. For all other ERP modulations that all start with ERP-DSSS short or long preamble, the bits of the SERVICE field are defined in Table 19-4 and the SERVICE field is not applicable in the TXVECTOR. Therefore, the entire field is reserved.

Table 19-1—TXVECTOR parameters

Parameter	Value *(continued)*
TXPWR_LEVEL	The transmit power level. The definition of these levels is up to the implementer. 1–8
TIME_OF_DEPARTURE_REQUESTED	False, true. When true, the MAC entity requests that the PHY PLCP entity measures and reports time of departure parameters corresponding to the time when the first frame energy is sent by the transmitting port; when false, the MAC entity requests that the PHY PLCP entity neither measures nor reports time of departure parameters.

The parameters in Table 19-2 are defined as part of the TXSTATUS parameter list in the PHY-TXSTART. confirm service primitive.

Table 19-2—TXSTATUS parameters

Parameter	Value
TIME_OF_DEPARTURE	0 to 2^{32}– 1. The locally measured time when the first frame energy is sent by the transmitting port, in units equal to 1/ TIME_OF_DEPARTURE_ClockRate. This parameter is present only if TIME_OF_DEPARTURE_REQUESTED is true in the corresponding request.
TIME_OF_DEPARTURE_ClockRate	0 to 2^{16}–1. The clock rate, in units of MHz, is used to generate the TIME_OF_DEPARTURE value. This parameter is present only if TIME_OF_DEPARTURE_REQUESTED is true in the corresponding request.
TX_START_OF_FRAME_OFFSET	0 to 2^{32}– 1. An estimate of the offset (in 10 ns units) from the point in time at which the start of the preamble corresponding to the frame was transmitted at the transmit antenna port to the point in time at which this primitive is issued to the MAC.

The parameters in Table 19-3 are defined as part of the RXVECTOR parameter list in the PHY-RXSTART.indication primitive. When implementations require the use of these vectors, some or all of these parameters may be used in the vectors.

Table 19-3—RXVECTOR parameters

Parameter	Value
DATARATE	The rate at which the PSDU was received in Mb/s. Allowed value depends on value of MODULATION parameter: ERP-DSSS: 1 and 2 ERP-CCK: 5.5 and 11 ERP-OFDM: 6, 9, 12, 18, 24, 36, 48, and 54 ERP-PBCC: 5.5, 11, 22, and 33 DSSS-OFDM: 6, 9, 12, 18, 24, 36, 48, and 54
LENGTH	The length of the PSDU in octets. Range: 1–4095

Table 19-3—RXVECTOR parameters *(continued)*

Parameter	Value
PREAMBLE_TYPE	The preamble type detected during reception of the PPDU. Enumerated type for which the allowed value depends on value of MODULATION parameter: ERP-OFDM: null ERP-DSSS, ERP-CCK, ERP-PBCC, PBCC, DSSS-OFDM: SHORTPREAMBLE, LONGPREAMBLE.
MODULATION	The modulation used for the reception of this PSDU. Enumerated types: ERP-DSSS, ERP-CCK, ERP-OFDM, ERP-PBCC, DSSS-OFDM
SERVICE	Null.
RSSI	The RSSI is a measure of the RF energy received by the ERP. The 8-bit value is in the range of 0 to RSSI maximum as described in 18.2.3.3.
RCPI	The RCPI is a measure of the received channel power and is included when dot11RadioMeasurementActivated is true. The 8-bit RCPI value is described in 18.2.3.6 and 17.4.5.17.
RX_START_OF_FRAME_OFFSET	0 to 2^{32}– 1. An estimate of the offset (in 10 ns units) from the point in time at which the start of the preamble corresponding to the incoming frame arrived at the receive antenna port to the point in time at which this primitive is issued to the MAC.

19.3 Extended Rate PLCP sublayer

19.3.1 Introduction

Subclause 19.3 provides a PLCP for the ERP. The convergence procedure specifies how PSDUs are converted to and from PPDUs at the transmitter and receiver. The PPDU is formed during data transmission by appending the PSDU to the Extended Rate PLCP preamble and header. At the receiver, the PLCP preamble and header are processed to aid in the demodulation and delivery of the PSDU.

19.3.2 PPDU format

19.3.2.1 General

An ERP STA shall support three different preamble and header formats. The first is the long preamble and header described in 19.3.2.2 (and based on 17.2.2.2 with redefinition of reserved bits defined therein). This PPDU provides interoperability with Clause 17 STAs when using the 1, 2, 5.5, and 11 Mb/s data rates; the optional DSSS-OFDM modulation at all OFDM rates; and the optional ERP-PBCC modulation at all ERP-PBCC rates. The second is the short preamble and header described in 19.3.2.3 (and based on 17.2.2.3 where it is optional). The short preamble supports the rates 2, 5.5, and 11 Mb/s as well as DSSS-OFDM and ERP-PBCC. The third is the ERP-OFDM preamble and header specified in 19.3.2.4 (and based on 18.3.2). The ERP has two optional PPDU formats, described in 19.3.2.5 and 19.3.2.7, to support the optional DSSS-OFDM modulation rates.

19.3.2.2 Long preamble PPDU format

19.3.2.2.1 General

Figure 17-1 of 17.2.2.2 shows the basic format for the long preamble PPDU. This preamble is appropriate for use with the 1, 2, 5.5, and 11 Mb/s (Clause 17) modes and is compatible with BSSs using these modes. To support the optional modes included in the ERP, the long preamble PPDU only differs from 17.2.2.2 in the following:

a) The use of one bit in the SERVICE field to indicate when the optional ERP-PBCC mode is being used.

b) The use of two additional bits in the SERVICE field to resolve the length ambiguity when the optional ERP-PBCC-22 and ERP-PBCC-33 modes are being used.

c) Three additional optional rates given by the following SIGNAL field octets where the LSB is transmitted first in time:

   1) X'DC' (MSB to LSB) for 22 Mb/s ERP-PBCC

   2) X'21' (MSB to LSB) for 33 Mb/s ERP-PBCC

   3) X'1E' (MSB to LSB) for all DSSS-OFDM rates

Three bits of the SERVICE field have been defined to support the optional modes of the ERP standard. Table 19-4 shows graphically the assignment of the bits within the SERVICE field. The bits b0, b1, and b4 are reserved and shall be set to 0. Bit b2 is used to indicate that the transmit frequency and symbol clocks are derived from the same oscillator. For all ERP systems, the Locked Clock Bit shall be set to 1, when transmitting at an ERP-PBCC rate or at a data rate described in Clause 17. Bit b3 is used to indicate if the data are modulated using the optional ERP-PBCC modulation. Bit b3 is defined in 17.2.3.5 with the caveat that the ERP-PBCC mode now has the additional optional rates of 22 Mb/s and 33 Mb/s as defined in 19.3.3.2. Bits b5, b6, and b7 are used to resolve data field length ambiguities for the optional ERP-PBCC-11 to ERP-PBCC-33 modes. These bits are fully defined in 19.6. Bit b7 is also used to resolve data field length ambiguities for the CCK 11 Mb/s mode and is defined in 17.2.3.6. Bits b3, b5, and b6 are set to 0 for CCK.

Table 19-4—SERVICE field bit definitions

b0	b1	b2	b3	b4	b5	b6	b7
Reserved	Reserved	Locked Clock Bit 0 = not locked 1 = locked	Modulation Selection 0 = Not ERP-PBCC 1 = ERP-PBCC	Reserved	Length Extension Bit (ERP-PBCC)	Length Extension Bit (ERP-PBCC)	Length Extension Bit

19.3.2.2.2 ERP PLCP length field calculation

For the long and short preamble modes other than PBCC, the length field shall be calculated as in 17.2.3.6.

19.3.2.2.3 ERP-PBCC PLCP length (LENGTH) field calculation

For the ERP-PBCC PLCP length field, the transmitted value shall be determined from the LENGTH and DataRate parameters in the TXVECTOR issued with the PMD-TXSTART.request primitive described in 17.4.5.7.

The length field provided in the TXVECTOR is in octets and is converted to microseconds for inclusion in the PLCP LENGTH field. The Length Extension bits are provided to resolve the ambiguity in the number of octets that is described by an integer number of microseconds for any data rate over 8 Mb/s. These bits are used to indicate which of the smaller potential number of octets is correct.

- 11 Mb/s PBCC: see 17.2.3.6.
- 22 Mb/s ERP-PBCC: Length = (number of octets + 1) × 4/11, rounded up to the next integer; the SERVICE field bits b6 and b7 shall each indicate a 0 if the rounding took less than 4/11; the SERVICE field bit b6 shall indicate a 0, and b7 shall indicate a 1 if the rounding took 4/11 or more and less than 8/11; and the SERVICE field bit b6 shall indicate a 1, and b7 shall indicate a 0 if the rounding took 8/11 or more.
- 33 Mb/s ERP-PBCC: Length = (number of octets + 1) × 8/33, rounded up to the next integer; the SERVICE field bits b5, b6, and b7 shall each indicate a 0 if the rounding took less than 8/33; the SERVICE field bit b5 shall indicate a 0, b6 shall indicate a 0, and b7 shall indicate a 1 if the rounding took more than or equal to 8/33 and less than 16/33; the SERVICE field bit b5 shall indicate 0, b6 shall indicate a 1, and b7 shall indicate a 0 if the rounding took more than or equal to 16/33 and less than 24/33; the SERVICE field bit b5 shall indicate 0, b6 shall indicate a 1, and b7 shall indicate a 1 if the rounding took more than or equal to 24/33 and less than 32/33; the SERVICE field bit b5 shall indicate 1, b6 shall indicate a 0, and b7 shall indicate a 0 if the rounding took 32/33 or more.

At the receiver, the number of octets in the MPDU is calculated as follows:

- 22 Mb/s ERP-PBCC: Number of octets = (Length × 11/4) – 1, rounded down to the next integer, minus 1 if the SERVICE field bit b6 is a 0 and b7 is a 1, or minus 2 if the SERVICE field bit b6 is a 1 and b7 is a 0.
- 33 Mb/s ERP-PBCC: Number of octets = (Length × 33/8) – 1, rounded down to the next integer, minus 1 if the SERVICE field bit b5 is a 0, b6 is a 0, and b7 is a 1, or minus 2 if the SERVICE field bit b5 is a 0, b6 is a 1, and b7 is a 0, or minus 3 if the SERVICE field bit b5 is a 0, b6 is a 1, and b7 is a 1, or minus 4 if the SERVICE field bit b5 is a 1, b6 is a 0, and b7 is a 0.

Table 19-5 shows an example calculation for several packet lengths of ERP-PBCC at 22 Mb/s.

Table 19-5—Example of LENGTH calculations for ERP-PBCC-22

TX octets	(Octets+1) × 4/11	LENGTH	Length Extension bit b6	Length Extension bit b7	LENGTH × 11/4	Floor(X)	RX octets
1023	372.364	373	0	1	1025.75	1025	1023
1024	372.727	373	0	0	1025.75	1025	1024
1025	373.091	374	1	0	1028.50	1028	1025
1026	373.455	374	0	1	1028.50	1028	1026

19.3.2.3 Short preamble PPDU format

Figure 17-2 of 17.2.2.3 shows the basic format for the short preamble PPDU. For the ERP, support for this preamble is mandatory. The short preamble is appropriate for use with 2, 5.5, and 11 Mb/s modes. The bits of the Short PLCP SERVICE field and RATE field are the same as for the Long PLCP SERVICE field and RATE field and are defined in 19.3.2.2.

19.3.2.4 ERP-OFDM PPDU format

The format, preamble, and headers for the ERP-OFDM PLCP PPDU are described in 18.3.2 to 18.3.5. For the ERP-OFDM modes, the DATA field that contains the SERVICE field, the PSDU, the TAIL bits, and the PAD bits shall follow 18.3.5.

For ERP-OFDM modes, an ERP packet is followed by a period of no transmission with a length of 6 μs called the signal extension. The purpose of this extension is to make the TXTIME calculation in 19.8.3 result in a transmission duration interval that includes an additional 6 μs. The SIFS time for Clause 18 packets is 16 μs, and the SIFS time for Clause 17 packets is 10 μs. The longer SIFS time in Clause 18 is to allow extra time for the convolutional decode process to finish. As Clause 19 packets use a SIFS time of 10 μs, this extra 6 μs length extension is used to ensure that the transmitter computes the Duration field in the MAC header incorporating the 6 μs of “idle time” following each ERP-OFDM transmission. This ensures that the NAV value of Clause 17 STAs is set correctly.

The “CS mechanism” described in 9.3.2.1 combines the NAV state and the STA’s transmitter status with physical CS to determine the busy/idle state of the medium. The time interval between frames is called the IFS. A STA shall determine that the medium is idle through the use of the CCA mechanism for the interval specified. The starting reference of slot boundaries is the end of the last symbol of the previous frame on the medium. For ERP-OFDM frames, this includes the length extension. For ERP-OFDM frames, a STA shall generate the PHY RX_END indication, 6 μs after the end of the last symbol of the previous frame on the medium. This adjustment shall be performed by the STA based on local configuration information set using the PLME SAP.

19.3.2.5 DSSS-OFDM long preamble PPDU format

Both long and short preambles and headers as previously described in 19.3.2.2 and 19.3.2.3 are used with DSSS-OFDM.

For all DSSS-OFDM rates and preamble modes, the PLCP SIGNAL field described in 17.2.3.4 shall be set to a 3 Mb/s value. That is, the 8 bit value is set to X'1E' (MSB to LSB). For DSSS-OFDM, this value is simply a default setting used for BSS compatibility and to ensure that NonERP STAs read the length field and defer the medium for that time even though they cannot demodulate the MPDU due to unsupported rates.

Figure 19-1 shows the PPDU format for the long preamble case. As seen, the PSDU is appended to the PLCP preamble and the PLCP header. The PLCP preamble is the same as described in 17.2.3.2 and 17.2.3.3. The PLCP header is similar to the one described in 19.3.2.2. The PSDU has a format that is nearly identical to a Clause 18 PLCP. The differences are described in 19.3.3.4.

The scrambler of 17.2.4 is used to scramble the DSSS-OFDM PLCP header, and the scrambler in 18.3.5.5 is used to scramble the data symbols in the OFDM segment.

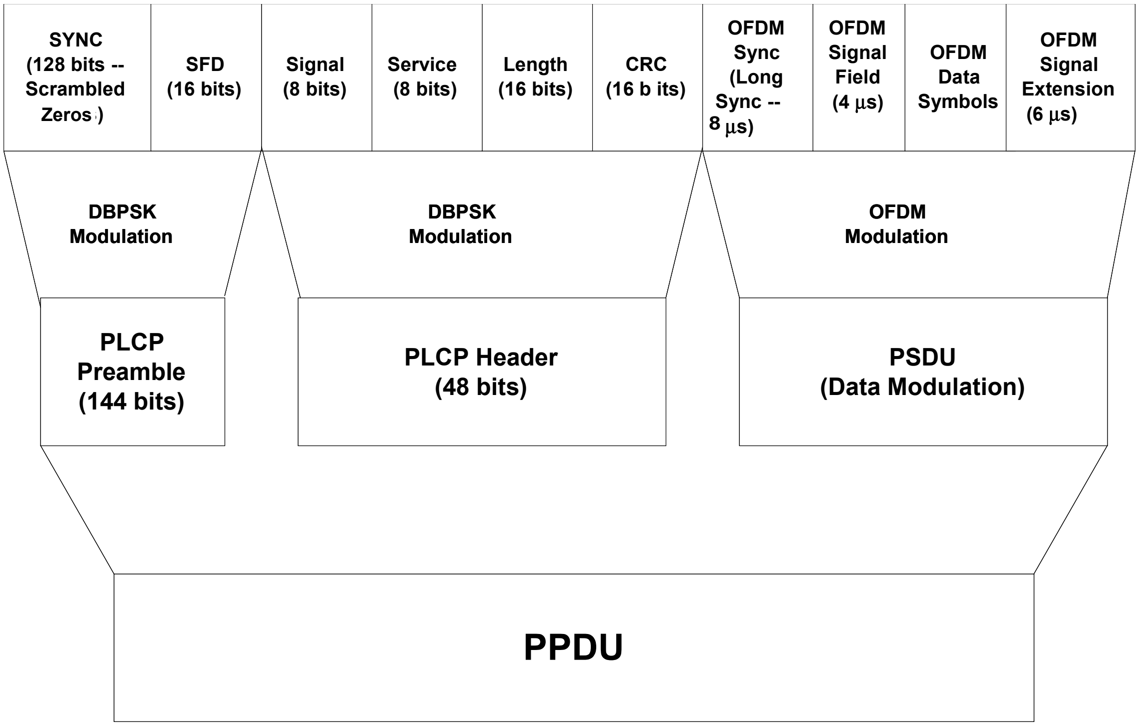


Figure 19-1—Long preamble PPDU format for DSSS-OFDM

19.3.2.6 DSSS-OFDM PLCP length field calculation

For both the long and the short preamble PLCP cases, the length field calculation in terms of data packet length is as follows:

$$\begin{aligned}&\text{LENGTH} = \text{PSDUsyncOFDM} + \text{PSDUSignalOFDM} + \\&\quad 4 \times \text{Ceiling}((\text{PLCPServiceBits} + 8 \times (\text{NumberOfOctets}) + \text{PadBits}) / N_{DBPS}) + \text{SignalExtension}\end{aligned}$$

where

PSDUsyncOFDM	is 8 μs (OFDM long training symbols)
PSDUSignalOFDM	is 4 μs
Ceiling	is a function that returns the smallest integer value greater than or equal to its argument value
PLCPServiceBits	is 8 bits
NumberOfOctets	is the number of data octets in the PSDU
PadBits	is 6 bits
N_{DBPS}	is the number of data bits per OFDM symbol
SignalExtension	is 6 μs

The length field is defined in units of microseconds and shall correspond to the calculated length of the PSDU. Note that the length extension bits in the Signal field are not needed or used for DSSS-OFDM.

19.3.2.7 Short DSSS-OFDM PLCP PPDU format

The short PLCP preamble and header are used to maximize the throughput by reducing the overhead associated with the preamble and header. Figure 19-2 shows the short preamble PLCP PPDU format. As seen, the PSDU is appended to the PLCP preamble and the PLCP header. The short PLCP preamble is described in 17.2.3.9 and 17.2.3.10. The PLCP header is as described in 19.3.2.5. The PSDU has a format that is nearly identical to Clause 18 PLCP. The differences are described in 19.3.3.4.

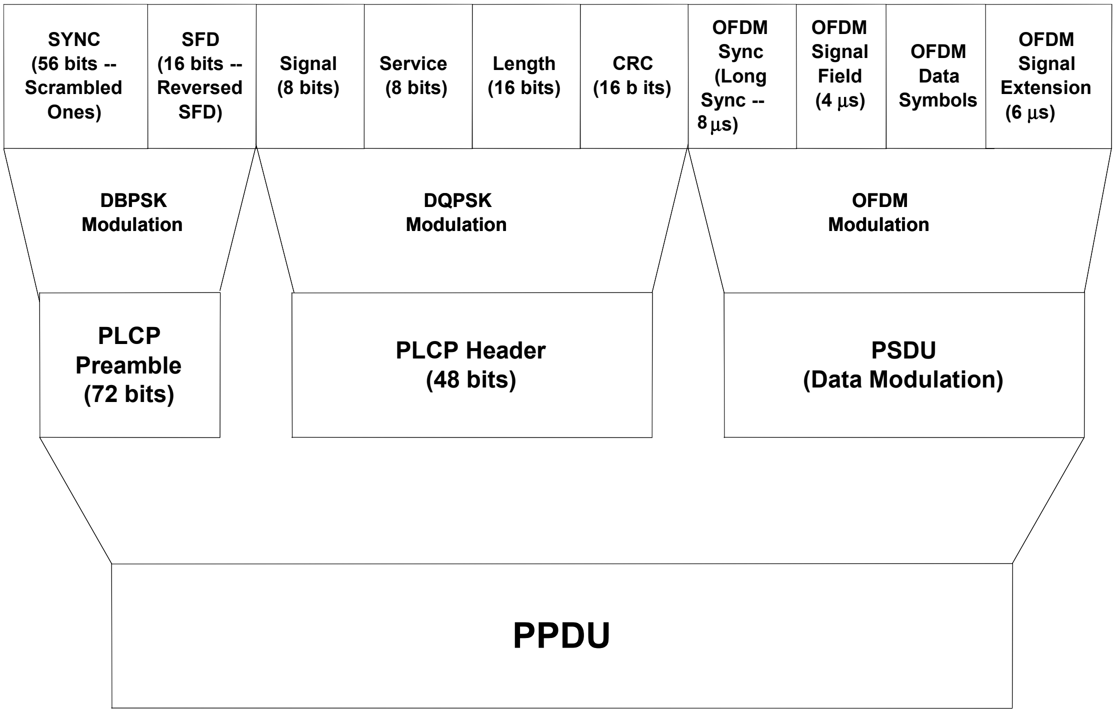


Figure 19-2—Short preamble PPDU format for DSSS-OFDM

19.3.3 PLCP data modulation and rate change

19.3.3.1 Long and short preamble formats

The long and short PLCP preamble and the long PLCP header shall be transmitted using the 1 Mb/s DBPSK modulation. The short PLCP header shall be transmitted using the 2 Mb/s modulation. The SIGNAL and SERVICE fields combined shall indicate the modulation and rate that shall be used to transmit the PSDU. The transmitter and receiver shall initiate the modulation and rate indicated by the SIGNAL and SERVICE fields, starting with the first octet of the PSDU. The PSDU transmission rate shall be set by the DATARATE parameter in the TXVECTOR, issued with the PHY-TXSTART.request primitive described in 17.4.5.2.

Four modulation formats are mandatory, 1 Mb/s and 2 Mb/s ERP-DSSS and 5.5 Mb/s and 11 Mb/s ERP-CCK, and they are specified in 17.4.6.4.

Four optional ERP-PBCC modulation formats and data rates are specified for the ERP. They shall be based on PBCC 5.5, 11, 22, and 33 Mb/s modulations. The rates of 5.5 Mb/s and 11 Mb/s are described in 17.4.6.7. No change in the spectral mask of 17.4.7.4 is required for these modes.

19.3.3.2 ERP-PBCC 22 Mb/s and 33 Mb/s formats

In the PBCC encoder, incoming data are first encoded with a packet binary convolutional code. A cover code (as defined in PBCC modes in 17.4.6.7) is applied to the encoded data prior to transmission through the channel.

The packet binary convolutional code that is used is a 256-state, rate 2/3 code. The generator matrix for the code is given as

$$G = \begin{bmatrix} 1 + D^4 & D & D + D^3 \\ D^3 & 1 + D^2 + D^4 & D + D^3 \end{bmatrix} \qquad (19\text{-}1)$$

In octal notation, the generator matrix is given by

$$G = \begin{bmatrix} 21 & 2 & 12 \\ 10 & 25 & 12 \end{bmatrix} \qquad (19\text{-}2)$$

As the system is frame (PPDU) based, the encoder shall be in state zero; i.e., all memory elements contain 0, at the beginning of every PPDU. The encoder shall also be placed in a known state at the end of every PPDU to prevent the data bits near the end of the PPDU from being decoded incorrectly. This is achieved by appending 1 octet containing all zeros to the end of the PPDU prior to transmission and discarding the final octet of each received PPDU.

An encoder block diagram is shown in Figure 19-3. It consists of two paths of four memory elements each. For every pair of data bits input, three output bits are generated. The output of the convolutional code is mapped to an 8-PSK constellation; each 3-bit output sequence from the packet binary convolutional encoder is used to produce one symbol. This yields a throughput of two information bits per symbol. In ERP-PBCC-22 and ERP-PBCC-33, the input data stream is divided into pairs of adjacent bits. In each pair, the first bit is fed to the upper input of the convolutional encoder, and the second is fed to the lower input of the convolutional encoder. An illustration of the mapping for the jth (j≥0) pair of input bits (b2j, b2j+1) is given in Figure 19-3.

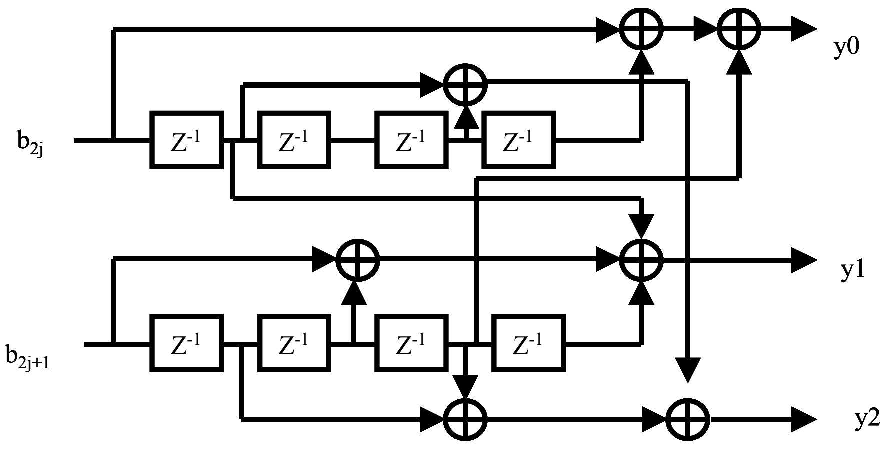


Figure 19-3—22/33 Mb/s ERP-PBCC convolutional encoder

The phase of the first complex chip of the 22 Mb/s PSDU shall be defined with respect to the phase of the last chip of the PCLP header, i.e., the last chip of the CRC check. The phase of the first complex chip of the 33 Mb/s PSDU shall be defined with respect to the phase of the last chip of the clock switch section, i.e., the last chip of the ReSync field. The bits (y2 y1 y0) = (0,0,0) shall indicate the same phase as the last chip of the CRC check. The other seven combinations of (y2 y1 y0) shall be defined with respect to this reference phase as shown in Figure 19-4.

The mapping from BCC outputs to 8-PSK constellation points is determined by a pseudorandom cover sequence. The cover sequence is the same one as described in 17.4.6.7. The current binary value of this sequence at every given point in time is taken as shown in Figure 19-4. The mapping is shown in Figure 19-4.

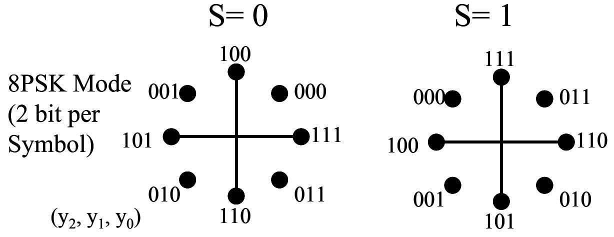


Figure 19-4—ERP-PBCC-22 and ERP-PBCC-33 cover code mapping

ERP-PBCC mode achieves a 33 Mb/s data rate by using a 16.5 MHz clock for the data portion of the packet. The data portion is otherwise identical to the 22 Mb/s ERP-PBCC modulations. The structure and clock speed of the preamble is the same as in Clause 17. An extra clock switch section between the preamble and the data portion is added, with the format described below. The same pulse shape shall be used in each clock domain.

When the clock is switched from 11 MHz to 16.5 MHz, the clock switching structure in Figure 19-5 is used.

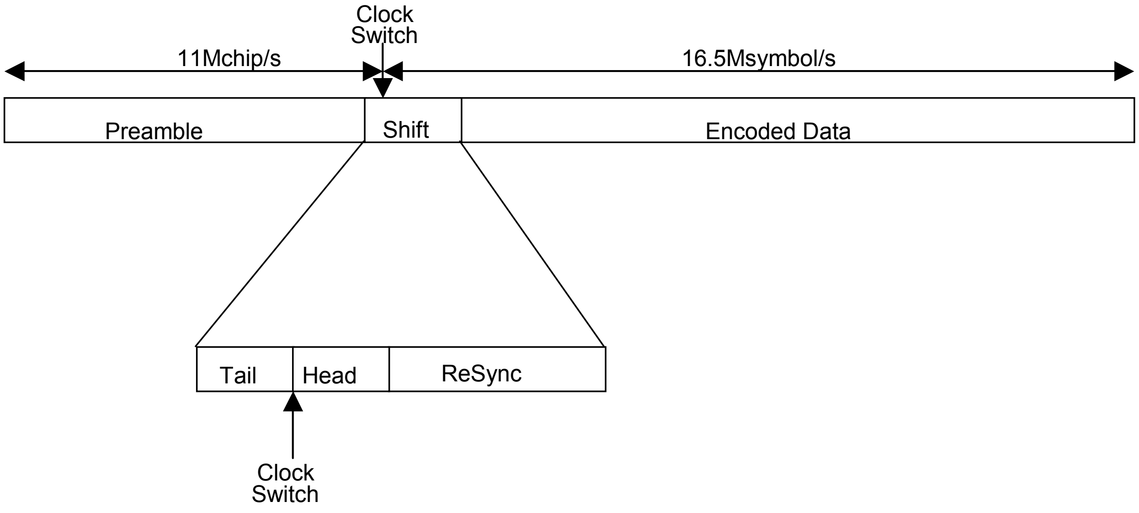


Figure 19-5—33 Mb/s clock switching

The tail is 3 clock cycles at 11 Mchip/s and the head is 3 clock cycles at 16.5 Msymbol/s (QPSK). The resync is 9 clock cycles at 16.5 Msymbol/s. The total clock switching time (tail and head and resync) is 1 µs. The tail bits are 1 1 1, the head bits are 0 0 0, and the resync bits are 1 0 0 0 1 1 1 0 1. The modulation is BPSK, which is phase synchronous with the previous symbol.

19.3.3.3 ERP-OFDM format

PLCP modulation and rate change for the ERP-OFDM frame format follows 18.3.7.

19.3.3.4 Long and short DSSS-OFDM PLCP format

19.3.3.4.1 General

The scrambler of 17.2.4 is used to scramble the DSSS-OFDM PLCP header, and the scrambler in 18.3.5.5 is used to scramble the data symbols in the OFDM segment.

19.3.3.4.2 Overview of the DSSS-OFDM PLCP PSDU encoding process

This subclause contains the definitions and procedure for forming the PSDU portion of the DSSS-OFDM PLCP. Figure 19-6 shows an expanded view of the DSSS-OFDM PSDU. The PSDU is composed of four major sections. The first is the long sync training sequence that is used for acquisition of receiver parameters by the OFDM demodulator. The long sync training sequence for DSSS-OFDM is identical to the long training symbols described in 18.3.3. The second section is the OFDM SIGNAL field that provides the demodulator information on the OFDM data rate and length of the OFDM data section. The SIGNAL field for DSSS-OFDM is identical to the SIGNAL field described in 18.3.4. After the SIGNAL field is the data section of the PSDU. This is identical to the modulation procedure described in 18.3.2.2 in Step c) to Step m). After the data section, the PSDU for DSSS-OFDM appends a signal extension section to provide additional processing time for the OFDM demodulator. This signal extension is a period of no transmission as described in 19.3.3.4.6.

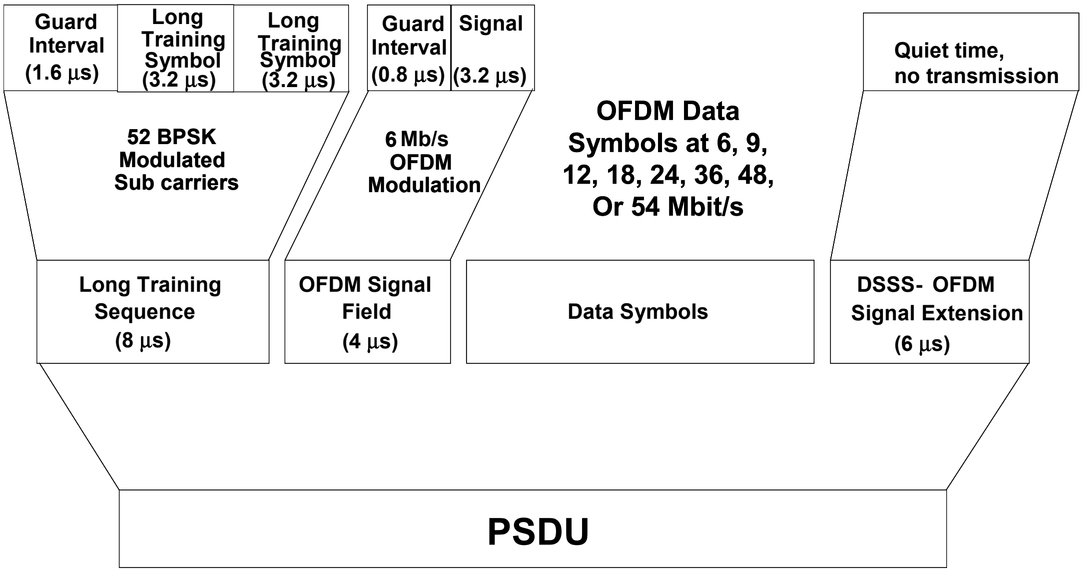

Guard Interval (1.6 µs)
Long Training Symbol (3.2 µs)
Long Training Symbol (3.2 µs)
Guard Interval (0.8 µs)
Signal (3.2 µs)
Quiet time, no transmission
52 BPSK Modulated Sub carriers
6 Mb/s OFDM Modulation
OFDM Data Symbols at 6, 9, 12, 18, 24, 36, 48, Or 54 Mbit/s
Long Training Sequence (8 µs)
OFDM Signal Field (4 µs)
Data Symbols
DSSS- OFDM Signal Extension (6 µs)
PSDU

Figure 19-6—DSSS-OFDM PSDU

19.3.3.4.3 Long sync training sequence definition

The long sync training sequence is defined in 18.3.3.

19.3.3.4.4 OFDM signal field definition

The DSSS-OFDM SIGNAL field is defined in 18.3.4. Note that the length conveyed by the SIGNAL field is calculated as described in 18.3.4. That is, the length conveyed by this field does not include the signal extension described in 19.3.3.4.6.

19.3.3.4.5 Data symbol definition

The same process as Step c) to Step m) of 18.3.2.2 is used to encode the data symbols' portion of the DSSS-OFDM PSDU.

19.3.3.4.6 DSSS-OFDM signal extension

The DSSS-OFDM signal extension shall be a period of no transmission of 6 µs length. It is inserted to allow more time to finish the convolutional decoding of the OFDM segment waveform and still meet the 10 µs SIFS requirement of the ERP.

19.3.4 PLCP transmit procedure

The transmit procedure depends on the data rate and modulation format requested. For data rates of 1, 2, 5.5, 11, 22, and 33 Mb/s, the PLCP transmit procedure shall follow 17.2.5. For the ERP_OFDM rates of 6, 12, and 24 Mb/s and the rates of 9, 18, 36, 48, and 54 Mb/s, the PLCP transmit procedure shall follow 18.3.11.

The transmit procedures for the optional DSSS-OFDM mode using the long or short PLCP preamble and header are the same as those described in 17.2.5, and they do not change apart from the ability to transmit a higher rate PSDU using DSSS-OFDM.

19.3.5 CCA

The PLCP shall provide the capability to perform a CCA and report the results of the assessment to the MAC. The CCA mechanism shall detect a "medium busy" condition for all supported preamble and header types. That is, the CCA mechanism shall detect that the medium is busy for the PLCP PPDUs specified in 18.3.3 and 17.2.2. The CCA mechanism performance requirements are given in 19.4.7.

The ERP shall provide the capability to perform CCA according to the following method:

> **CCA Mode** (ED and CS): A combination of CS and energy above threshold. CCA shall have a mechanism for CS that detects all mandatory Clause 19 sync symbols. This CCA's mode's CS shall include both Barker code sync detection and OFDM sync symbol detection. CCA shall report busy at least while a PPDU with energy above the ED threshold is being received at the antenna.

The ED status shall be given by the PMD primitive, PMD_ED. The CS status shall be given by PMD_CS. The status of PMD_ED and PMD_CS is used in the PLCP convergence procedure to indicate activity to the MAC through the PHY-CCA.indication primitive. A busy channel shall be indicated by PHY-CCA.indication primitive of class BUSY. A clear channel shall be indicated by PHY-CCA.indication primitive of class IDLE.

19.3.6 PLCP receive procedure

This subclause describes the procedure used by receivers of the ERP. An ERP receiver shall be capable of receiving 1, 2, 5.5, and 11 Mb/s PLCPs using either the long or short preamble formats described in Clause 17 and shall be capable of receiving 6, 12, and 24 Mb/s using the modulation and preamble described in Clause 18. The PHY may also implement the ERP-PBCC modulation at rates of 5.5, 11, 22, and 33 Mb/s; the ERP-OFDM modulations at rates of 9, 18, 36, 48, and 54 Mb/s; and/or the DSSS-OFDM modulation rates of 6, 9, 12, 18, 24, 36, 48, and 54 Mb/s. The receiver shall be capable of detecting the preamble type (ERP-OFDM, short preamble, or long preamble) and the modulation type. These values shall be reported in the RXVECTOR (see 19.2).

Upon the receipt of a PPDU, the receiver shall first distinguish between the ERP-OFDM preamble and the single carrier modulations (long or short preamble). In the case where the preamble is an ERP-OFDM preamble, the PLCP receive procedure shall follow the procedure described in 18.3.12. Otherwise, the receiver shall then distinguish between the long preamble and short preamble as specified in 17.2.2. The receiver shall then demodulate the SERVICE field to determine the modulation type as specified in 19.3.2.2 or 19.3.2.3. For short preamble and long preamble using ERP-DSSS, ERP-CCK, or ERP-PBCC modulations, the receiver shall then follow the receive procedure described in 17.2.6.

A receiver that supports DSSS-OFDM is capable of receiving all rates specified in Clause 16 and all mandatory rates in Clause 18 and Clause 17. If the SIGNAL field indicates 3 Mb/s, the receiver shall attempt to receive a DSSS-OFDM packet. The remaining receive procedures for a DSSS-OFDM-capable receiver are the same as those described in 17.2.6, and they do not change apart from the ability to receive DSSS-OFDM in the PSDU. If DSSS-OFDM is being received, the receiver shall handle the modulation transition requirements as described in 19.7.3. The receiver shall then follow the receive procedure described in 18.3.12.

19.4 ERP PMD operating specifications (general)

19.4.1 Introduction

Subclauses 19.4.2 to 19.4.8 provide general specifications for the ERP PMD sublayers. These specifications are based on 18.3.8 except where noted.

19.4.2 Regulatory requirements

All systems shall comply with the appropriate regulatory requirements for operation in the 2.4 GHz band.

19.4.3 Operating channel frequencies

The ERP shall operate in the frequency ranges specified in 17.4.6.3, as allocated by regulatory bodies in the United States, Europe, and Japan. OFDM operation in channel 14 may not be allowed in Japan. The channel numbering and the number of operating channels shall follow Table 17-9 of 17.4.6.3.

19.4.4 Transmit and receive in-band and out-of-band spurious emissions

The ERP shall conform to in-band and out-of-band spurious emissions as set by the appropriate regulatory bodies for the 2.4 GHz band.

19.4.5 Slot time

The slot time is 20 µs, except that an optional 9 µs slot time may be used when the BSS consists of only ERP STAs capable of supporting this option. The optional 9 µs slot time shall not be used if the network has one or more NonERP STAs associated. For IBSS, the Short Slot Time subfield shall be set to 0, corresponding to a 20 µs slot time.

19.4.6 SIFS value

The ERP shall use a SIFS of 10 µs.

19.4.7 CCA performance

The CCA shall indicate true if there is no CCA “medium busy” indication. The CCA parameters are subject to the following criteria:

a) When a valid signal with a signal power of –76 dBm or greater at the receiver antenna connector is present at the start of the PHY slot, the receiver’s CCA indicator shall report the channel busy with probability CCA_Detect_Probabilty within a CCA_Time. CCA_Time is SlotTime –

RxTxTurnaroundTime. CCA_Detect_Probabilty is the probability that the CCA does respond correctly to a valid signal. The values for these parameters are found in Table 19-6. Note that the CCA Detect Probability and the power level are performance requirements.

b) In the event that a correct PLCP header is received, the ERP shall hold the CCA signal inactive (channel busy) for the full duration, as indicated by the PLCP LENGTH field. Should a loss of CS occur in the middle of reception, the CCA shall indicate a busy medium for the intended duration of the transmitted PPDU.

Table 19-6—CCA parameters

Parameter	Slot time = 20 μs	Slot time = 9 μs
SlotTime	20 μs	9 μs
RxTxTurnaroundTime	5 μs	5 μs
CCA_Time	15 μs	4 μs
CCA_Detect_Probability	> 99%	> 90%

19.4.8 PMD transmit specifications

19.4.8.1 General

The PMD transmit specifications shall follow 18.3.9 with the exception of the transmit power level (18.3.9.2), the transmit center frequency tolerance (18.3.9.5), the symbol clock frequency tolerance (18.3.9.6), and the time of departure accuracy (18.3.9.9). Regulatory requirements may have an effect on the combination of maximum transmit power and spectral mask if the resulting signals violate restricted band emission limits.

19.4.8.2 Transmit power levels

The maximum transmit power level shall meet the requirements of the local regulatory body.

19.4.8.3 Transmit center frequency tolerance

The transmit center frequency tolerance shall be ± 25 PPM maximum. The transmit center frequency and symbol clock frequency shall be derived from the same reference oscillator (locked).

19.4.8.4 Symbol clock frequency tolerance

The symbol clock frequency tolerance shall be ± 25 PPM maximum. The transmit center frequency and symbol clock frequency shall be derived from the same reference oscillator (locked oscillators). This means that the error in PPM for the carrier and the symbol timing shall be the same.

19.4.8.5 Time of Departure accuracy

The time of departure specifications shall follow 18.3.9.9 for PPDUs transmitted using ERP-OFDM format and 17.4.7.10 for PPDUs transmitted using ERP-DSSS/CCK, ERP-PBCC, and DSSS-OFDM formats.

19.5 ERP operation specifications

19.5.1 General

Subclause 19.5 describes the receive specifications for the PMD sublayer. The receive specification for the ERP-OFDM modes shall follow 18.3.10 with the exception of the receiver maximum input level (18.3.10.5) and the adjacent channel rejection (18.3.10.3). The receive specifications for the ERP-DSSS modes shall follow 17.4.8 with the exception of the receiver maximum input level (17.4.8.3).

19.5.2 Receiver minimum input level sensitivity

The PER of the ERP-OFDM modes shall be less than 10% at a PSDU length of 1000 octets for the input levels of Table 18-14 of 18.3.10. Input levels are specific for each data rate and are measured at the antenna connector. A noise figure of 10 dB and an implementation loss of 5 dB are assumed. The PER of the ERP-DSSS modes shall be as specified in 17.4.8.2.

19.5.3 Adjacent channel rejection

Adjacent channels at 2.4 GHz are defined to be at ± 25 MHz spacing. The adjacent channel rejection shall be measured by setting the desired signal's strength 3 dB above the rate-dependent sensitivity specified in Table 18-14 of 18.3.10 and raising the power of the interfering signal until 10% PER is caused for a PSDU length of 1000 octets. The power difference between the interfering and the desired channel is the corresponding adjacent channel rejection. The interfering signal in the adjacent channel shall be a conformant OFDM signal, unsynchronized with the signal in the channel under test. For an OFDM PHY, the corresponding rejection shall be no less than specified in Table 18-14 of 18.3.10.

The alternative adjacent channel rejection of Table 18-14 shall not be required for the ERP.

The adjacent channel rejection of the ERP-DSSS modes shall follow 17.4.8.4.

19.5.4 Receive maximum input level capability

The PER shall be less than 10% at a PSDU length of 1000 octets for an input level of –20 dBm measured at the antenna connector for any supported modulation signal or data rate (i.e., 1, 2, 5.5, 6, 9, 11, 12, 18, 22, 24, 33, 36, 48, 54 Mb/s).

19.5.5 Transmit spectral mask

The transmit spectral mask for the ERP-OFDM modes shall follow 18.3.9.3 and is shown in Figure 18-13 therein. The transmit spectral mask for the ERP-DSSS modes shall follow 17.4.7.4 and is shown in Figure 17-18 therein.

19.6 ERP-PBCC operation specifications

19.6.1 General

The ERP-PBCC receiver specifications shall follow 17.4.8 except as noted.

These optional modes provide systems with the ability to achieve data rates of 22 Mb/s and 33 Mb/s in modes that are fully backwards compatible with Clause 16 and Clause 17 BSSs without requiring additional coordination or protection mechanisms. In addition, the 22 Mb/s ERP-PBCC mode is spectrally identical to Clause 17 BSSs. Four optional ERP-PBCC modulation formats and data rates are specified for the ERP. They shall be based on PBCC 5.5, 11, 22, and 33 Mb/s modulations. The rates of 5.5 Mb/s and 11 Mb/s are described in 17.4.6.7.

19.6.2 Receiver minimum input level sensitivity

For the 22 Mb/s ERP-PBCC mode, the frame error ratio shall be less than 8×10^{-2} at a PSDU length of 1024 octets for an input level of –76 dBm measured at the antenna connector. For the 33 Mb/s ERP-PBCC mode, the corresponding input level shall be –74 dBm.

19.6.3 Receiver adjacent channel rejection

The adjacent channel rejection shall be equal to or better than 35 dB, with an FER of 8×10^{-2} using ERP-PBCC modulation and a PSDU length of 1024 octets. The adjacent channel rejection shall be measured using the following method. Input an ERP-PBCC modulated signal of the same rate at a level 6 dB greater than specified in 19.6.2. In an adjacent channel (25 MHz separation as defined by the channel numbering), input a signal modulated in a similar fashion, which adheres to the transmit mask specified in 17.4.7.4, to a level 41 dB above the level specified in 19.6.2. The adjacent channel signal shall be derived from a separate signal source. It shall not be a frequency-shifted version of the reference channel. Under these conditions, the FER shall be no worse than 8×10^{-2}.

19.7 DSSS-OFDM operation specifications

19.7.1 General

This optional mode provides systems with the ability to use OFDM in a mode that is fully compatible with Clause 16 and Clause 17 BSSs without requiring additional coordination. That is, it does not need a protection mechanism. This compatibility requires the use of Clause 17 long and short preambles and inclusion of a signal extension field to match SIFS spacing of Clause 17 systems. By reusing the Clause 17 preambles, this optional mode ensures that the Clause 17 CCA and SIFS interval function properly when ERP and NonERP STAs interoperate. When this option is enabled, the same rates shall be supported in both ERP-OFDM and DSSS-OFDM. The DSSS-OFDM PMD transmit and receive specifications shall follow the related ERP-OFDM specifications in 19.5.

19.7.2 Overview

This optional extension of the DSSS system builds on the payload data rates of 1, 2, 5.5, and 11 Mb/s, as described in Clause 17, to provide 6, 9, 12, 18, 24, 36, 48, and 54 Mb/s payload data rates while reusing the preambles (short and long) described by Clause 17. The capability described in this subclause is called DSSS-OFDM. This optional capability complements the Extended Rate OFDM mode described in Clause 19 by combining OFDM modulation with DSSS preambles. As a result, for DSSS-OFDM, the PPDU format described in 17.2.2 is relatively unchanged. The major change is to the format of the PSDU. The Clause 17 single carrier PSDU is replaced by a PSDU that is very similar to the PSDUs described in Clause 18. This subclause highlights the differences. In addition, 19.7.3 specifies the radio and physical layer behavior of the transition from the Barker symbol-modulated preamble and the OFDM-modulated data for PSDU.

19.7.3 Single carrier to multicarrier transition requirements

19.7.3.1 General

The spectrum mask for the DSSS-OFDM waveform shall meet the requirements as shown in Figure 18-13 of 18.3.9.3.

The single carrier signal segment of the packet shall have a coherent relationship with the multicarrier (OFDM) segment of the packet. All characteristics of the signal shall be transferable from one symbol to the next, even when transitioning to the OFDM segment. This enables high-performance, coherent receiver operation across the whole packet. This requirement is no different in nature than that stated in Clause 16, Clause 18, and Clause 17. The distinction is that those clauses use a signaling scheme that is either just single carrier or just multicarrier. In contrast, for this mode, both single carrier signaling and multicarrier signaling are used within the context of a single packet.

This subclause specifies the coherent relationship between the single carrier segment and the OFDM segment, so that the receiver has the opportunity to track through the transition without any forced parameter reacquisition. The single carrier preamble and header provide all parametric information required for demodulation of the OFDM segment to within conventional estimation-in-noise accuracy. Although multicarrier sync features are provided for convenience at OFDM segment onset, if and how to use the multicarrier sync for reacquisition is an implementer’s decision. Multicarrier sync is not necessary. The packet is coherent throughout.

As shown in Figure 19-7, the ideal transition would provide a constant carrier frequency and phase, a constant power, a constant spectrum, and a constant timing relationship. Constant in this context means that the same clock crystal that sets the frequencies and timing of each part is the same through the transition. This allows the frequency and timing tracking loops to work undisturbed through the transition. Subclauses 19.7.3.2 to 19.7.3.7 establish the ideal transition characteristics for the transmit signal. Subclause 19.7.3.8 specifies the required implementation fidelity or accuracy.

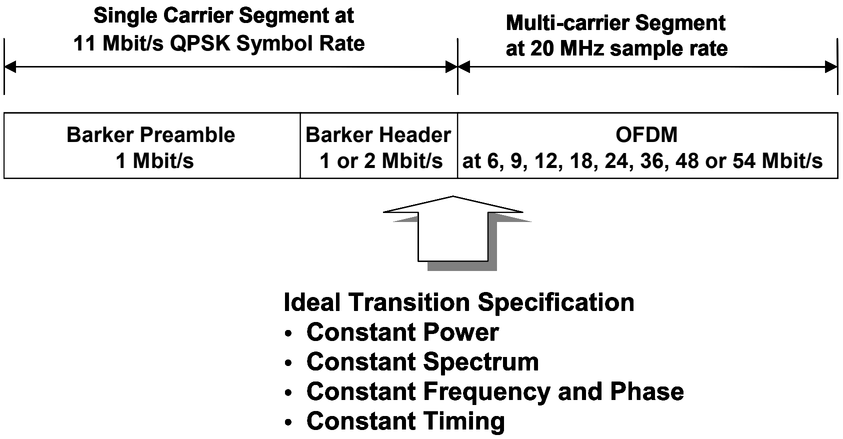

Single Carrier Segment at
11 Mbit/s QPSK Symbol Rate
Multi-carrier Segment
at 20 MHz sample rate
Barker Preamble
1 Mbit/s
Barker Header
1 or 2 Mbit/s
OFDM
at 6, 9, 12, 18, 24, 36, 48 or 54 Mbit/s
Ideal Transition Specification
• Constant Power
• Constant Spectrum
• Constant Frequency and Phase
• Constant Timing

Figure 19-7—Single carrier to multicarrier transition definition

19.7.3.2 Spectral binding requirement

19.7.3.2.1 General

The spectral binding requirement allows the receiver’s estimate of the channel state information to be transferred from the single carrier packet segment to the multicarrier packet segment. This requirement establishes a coherent relationship between the end-to-end frequency responses of the single carrier and multicarrier segments.

During reception of the single carrier preamble and header, the receiver may estimate the channel impulse response. In practice, this could be accomplished through Barker code correlation. The channel impulse response contains end-to-end frequency response information about the linear distortion experienced by the signal due to filters and multipath. This distortion might be mitigated with an equalizer or other commonly known techniques.

The channel impulse response estimate generated during the single carrier packet segment includes the single carrier's pulse-shaping, filter frequency response used to control the single carrier's transmit spectrum and transmit impulse response. The single carrier's pulse-shaping filter may be distinct from the shaping technique used for the multicarrier segment.

The spectral binding requirement states that the linear distortions experienced by the single carrier signal and the linear distortions experienced by the multicarrier signal have a known relationship. This relationship is defined by this specification and shall be manifested by all compliant transmit radios. This allows any receiver to exploit channel information derived during the single carrier segment and reuse the channel information during the multicarrier segment, if desired.

Three elements have been itemized for this specification to achieve spectral binding. All three elements are necessary to achieve spectral binding, and they are discussed in the next three subclauses. The first element focuses on distortions common to both the single carrier packet segment and the multicarrier packet segment. The second element deals with pulse-shaping unique to the OFDM packet segment. The third element deals with pulse-shaping unique to the single carrier packet segment. The multicarrier pulse shape discussion precedes the single carrier's pulse shape discussion because it is believed this is a more comfortable progression, due to similar multicarrier pulse-shaping considerations contained in Clause 18.

19.7.3.2.2 Common linear distortions

Separate from the single carrier and the multicarrier pulse shaping, transmit signal generation is designed to provide linear distortion continuity to the receiver's demodulation algorithms. The common linear distortion requirement is illustrated in Figure 19-8, where it is shown that the processing at the receiver assumes that the dominant linear distortions are induced on all waveform segments. The receiver observes the composite linear distortion due to imperfect transmit radio filters, due to multipath filtering, and due to imperfect receive radio filters. In general, the receiver is unable to decompose the distortion into separate physical components and is only able to observe the aggregate effect. This specification constrains only the linear distortions in the transmit radio, because that is what is necessary to ensure interoperability. The soft switch that appears in Figure 19-8 is a conceptual element to implement the transition, as described below.

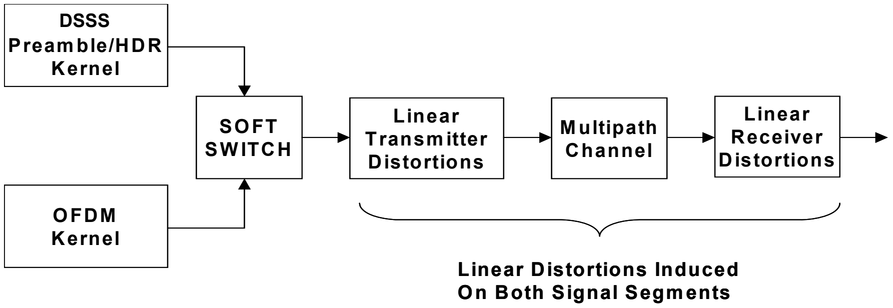

DSSS Preamble/HDR Kernel
OFDM Kernel
SOFT SWITCH
Linear Transmitter Distortions
Multipath Channel
Linear Receiver Distortions
Linear Distortions Induced On Both Signal Segments

Figure 19-8—Linear distortions common to the single carrier and multicarrier signal segments

In short, this common linear distortion requirement states that the dominant filters in the transmit radio stay invariant and common to all waveform segments. Once the receiver has determined the end-to-end impulse response, channel information is assumed to be common to both the single carrier signal and the multicarrier signal. This enables receiver design of linear-distortion mitigation techniques that do not require a reacquisition after transitioning to OFDM.

19.7.3.2.3 Symbol shaping unique to the DSSS-OFDM segment

OFDM spectral shaping may be achieved using two mechanisms: (1) Time-domain convolution filtering may be used to shape the spectrum. (2) Time-domain window tapering of OFDM symbol onset and termination may be used to shape the spectrum. This second mechanism can be viewed as frequency domain convolution. The first mechanism shall be common to both the OFDM and single carrier if it is a dominant distortion mechanism. The second mechanism may be unique to the OFDM segment, because it does not affect the frequency response of the 52 subcarriers.

The first spectral shaping mechanism using time-domain convolution filtering shall be common to both the single carrier and multicarrier segments for the reasons described in the preceding section. The receiver should not see intrapacket frequency response discontinuities.

Convolution filtering may be budgeted in various ways. One option would be to use a single filter that both the single carrier and multicarrier segments use. Another option would be to use two different physical filter realizations, one for the single carrier segment and a second for the multicarrier segment, say, for reason of distinct sample rates or bit precision. With this second implementation option, the designer shall ensure the frequency response of the filter is common to both packet segments.

The second shaping mechanism, which uses frequency-domain convolution through time-domain subcarrier onset-and-termination shaping, may be unique to the OFDM segment. This unique technique is acceptable because it does not modify the required frequency response of the 52 subcarriers.

Spectral shaping by tapering the OFDM symbol onset-and-termination using a time-domain window is described in Clause 18 and is equally germane to Clause 19 systems. For convenience, one of the relevant figures from Clause 18 is repeated here as Figure 19-9. Clause 18 suggested that the tapering transition duration is 0.1 μs.

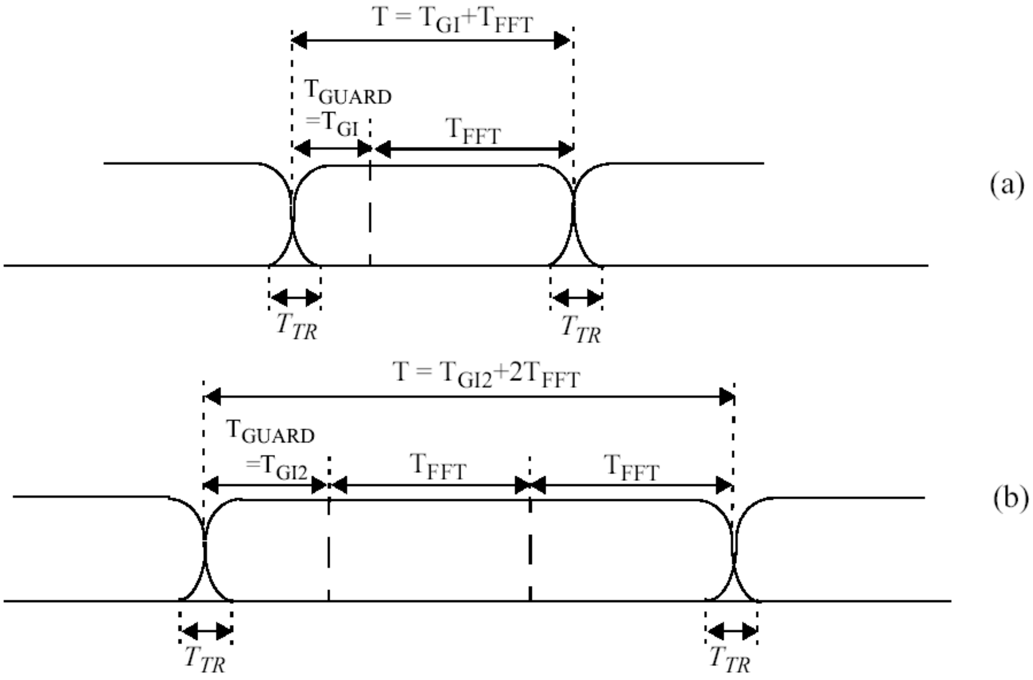


Figure 19-9—Spectral shaping achieved by OFDM symbol onset and termination shaping

The effect of time-domain windowing on a single subcarrier's power spectrum is shown in Figure 19-10 for two cases. The first case is rectangular time-domain windowing of an OFDM symbol. The second case is for the Clause 18 suggested time-domain windowing of an OFDM symbol with a 0.1 µs transition. Note the difference in frequency-dependent amplitude roll-off. Adding the 52 frequency-bin-centered individual subcarrier power spectral densities generates the composite 52 subcarrier power spectrum.

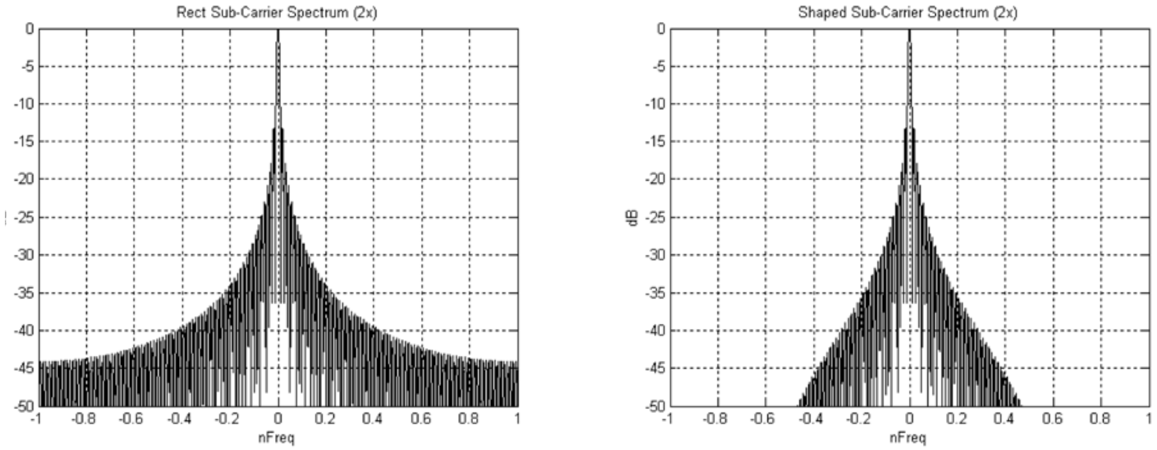


Figure 19-10—Subcarrier spectrums for rectangular windowing and Clause 18 suggested windowing

This type of OFDM spectrum control does not affect the relative amplitudes and phases of the individual subcarriers. Instead, it affects each subcarrier's power spectral density. Consequently, this type of spectrum control has a benign effect on the relative spectrums of single carrier and the multicarrier packet segments.

To achieve the design goal, the implementer may budget spectral shaping in the transmit radio. Some of the spectral shaping may be achieved using time-domain convolution filtering, and some may be achieved through time-domain windowing of the OFDM. In any case, the transmit implementation shall provide frequency response coherency.

19.7.3.2.4 Pulse shaping unique to the single carrier segment

This subclause describes the pulse-shaping requirements of the single carrier segment of the DSSS-OFDM packet. To establish frequency response coherency, it is necessary to specify the frequency response of the single carrier signal that establishes a coherent relationship to the frequency response of the OFDM.

The frequency response of the single carrier pulse is patterned after the tandem OFDM. The pattern is the OFDM signal as described in Clause 18, with an example provided in Annex L. The ideal OFDM signal has a flat amplitude response and zero-phase offset across 52 subcarriers. Clause 18 establishes the ideal frequency-response relationship among the 12 short SYNC subcarriers, 52 long SYNC subcarriers, the 52 SIGNAL field subcarriers, and the 52 data field subcarriers. Similarly, the ideal relationship to the single carrier frequency response is defined.

Relative to the ideal OFDM, the single carrier part of the DSSS-OFDM signal shall have the pulse shape established herein. In a particular implementation, it is acceptable to deviate from this ideal but only in a manner that is common to both the single carrier signal and the multicarrier signal across the passband of the OFDM signal. This requirement provides the required frequency response coherency.

The frequency response of the single carrier pulse is patterned after the OFDM that is transmitted in tandem. The single carrier pulse is derived from a time-windowed sinc function as shown in Figure 19-11 and Equation (19-3). The sinc function is the time response of an ideal brickwall filter. The brickwall filter is set equal to the bandwidth of an ideal OFDM signal. In particular, the bandwidth of the brickwall filter has been set to 52 times the Clause 18 subcarrier spacing of 20/64 MHz, or 312.5 KHz

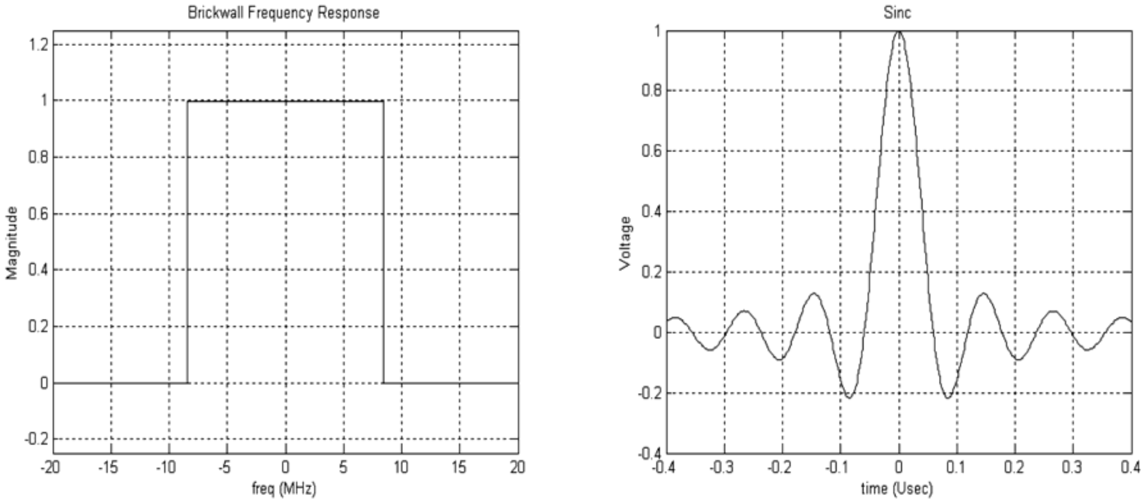


Figure 19-11—Foundational brickwall filter

$$h_{IdealBW}(t) = f_W \frac{\sin(\pi f_W t)}{\pi f_W t} = f_W \mathrm{sinc}(f_W t) \tag{19-3}$$

where

$f_W = 52(20/64)$ MHz

The infinite duration impulse response of the brickwall filter should be windowed to something practical. A continuous time version of the Hanning window may be used. The Hanning window and an overlay of the sinc function are shown in Equation (19-4) and Figure 19-12.

$$h_{\mathrm{Window}}(t) = 0.5\left[1 + \cos\left(2\pi \frac{t}{t_{\mathrm{SPAN}}}\right)\right] \tag{19-4}$$

where

$T_{\mathrm{SPAN}} = 0.8\ \mu\mathrm{s}$

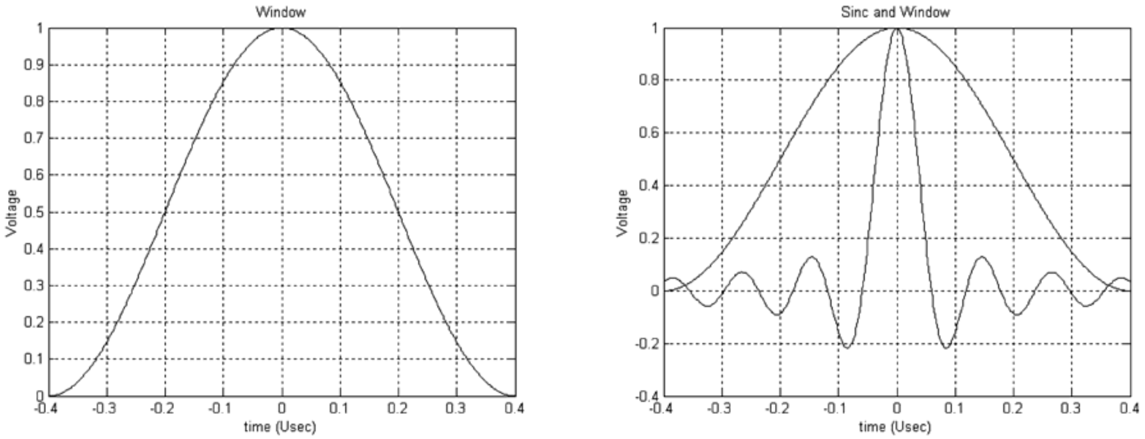


Figure 19-12—Continuous time Hanning window

The pulse specified for use with the single carrier packet segment is obtained by application of the window as shown in Equation (19-5) and Figure 19-13. Notice that its duration is equal to a Clause 18 short sync cycle, only 0.8 μs.

$$p(t) = h_{\text{Window}} h_{\text{IdealBW}}(t) \qquad (19\text{-}5)$$

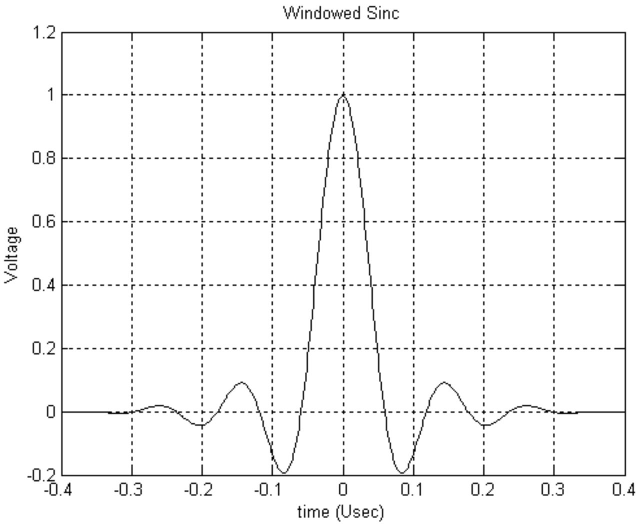


Figure 19-13—Specified pulse

The frequency response of the derived pulse is shown in Figure 19-14. This pulse generates a single carrier signal that has a spectrum nearly equal to that of the OFDM signal. This means that the receiver experiences essentially no change in receive signal power behavior even in the presence of multipath. At the point of the outermost subcarrier in the OFDM signal, the single carrier spectrum is down only about 4 dB. This is deemed adequate because the single carrier preamble-header is long in duration compared to the Clause 18 sync duration. Plenty of time is available to generate channel impulse response information that is sufficiently accurate.

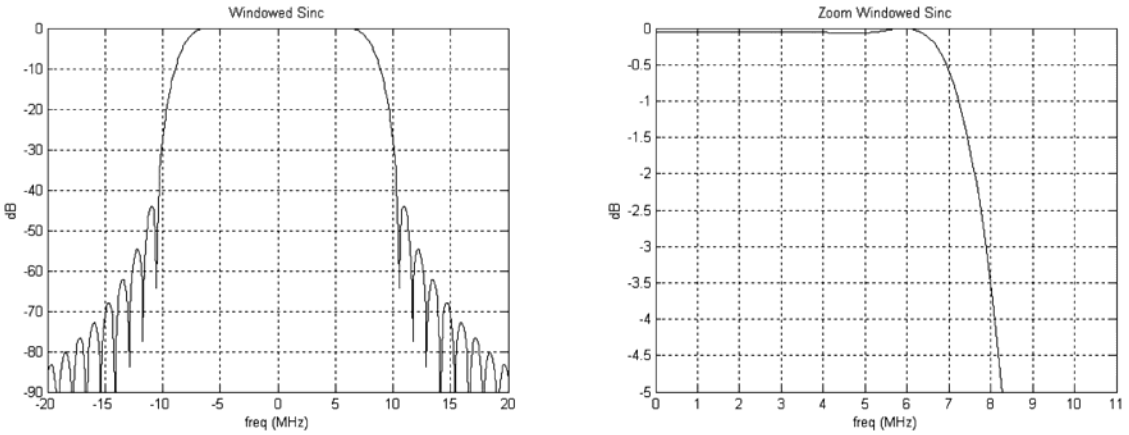


Figure 19-14—Single carrier frequency response

In summary, the specified single carrier pulse provides frequency response coherency between the single carrier and multicarrier segments of the packet. This does not mean that the spectrums are identical between segments. Rather, it means the ideal frequency responses of both are known. Beyond this, all linear distortion is common to both. It is not necessary to use this single carrier pulse during Clause 16 or Clause 17 packet transmissions.

19.7.3.3 Sample-power matching requirement

The transmit signal power shall be equal for the single carrier and multicarrier signal segments. The point of comparison is shown in Figure 19-15. The power measurement shall be over the single carrier header and over the OFDM data symbols.

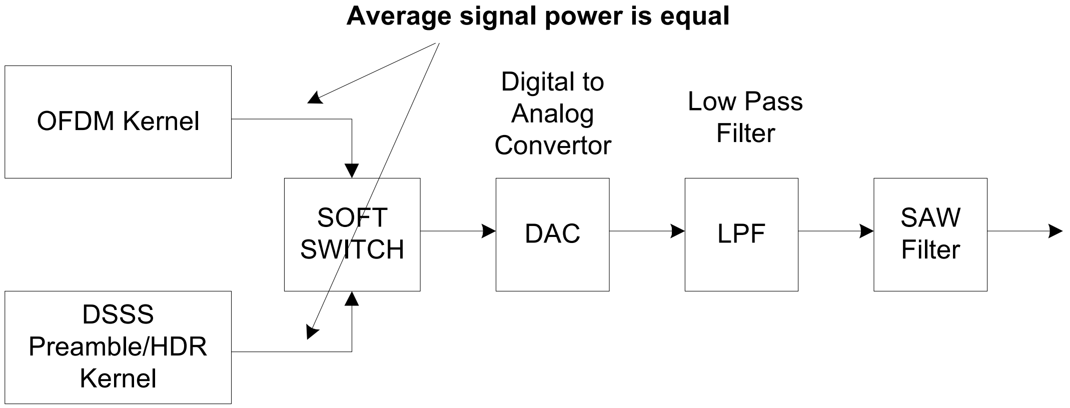


Figure 19-15—Comparing signal power

19.7.3.4 Transition time alignment

This subclause describes how the single carrier signal and the multicarrier signal are time aligned. The single carrier signal uses a chip rate of 11 MHz. The OFDM signal uses a fundamental sample rate of 20 MHz. The signals are easily aligned by first aligning the 11 MHz clock and the 20 MHz clock on 1 µs boundaries as shown in Figure 19-16.

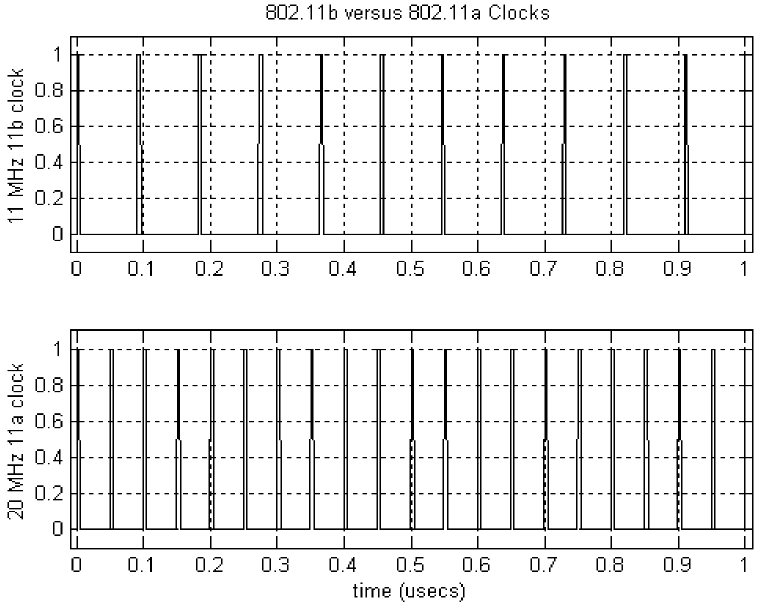


Figure 19-16—Aligning the 11 MHz and 20 MHz clocks

The 11 Barker chips of the preamble and header are transmitted aligned with this timing epoch. The first Barker chip is transmitted synchronous to the epoch, and then the remaining 10 chips follow. This is repeated over the duration of the preamble and header.

The peak of the continuous-time single carrier pulse shall be aligned to this epoch as shown in Figure 19-17.

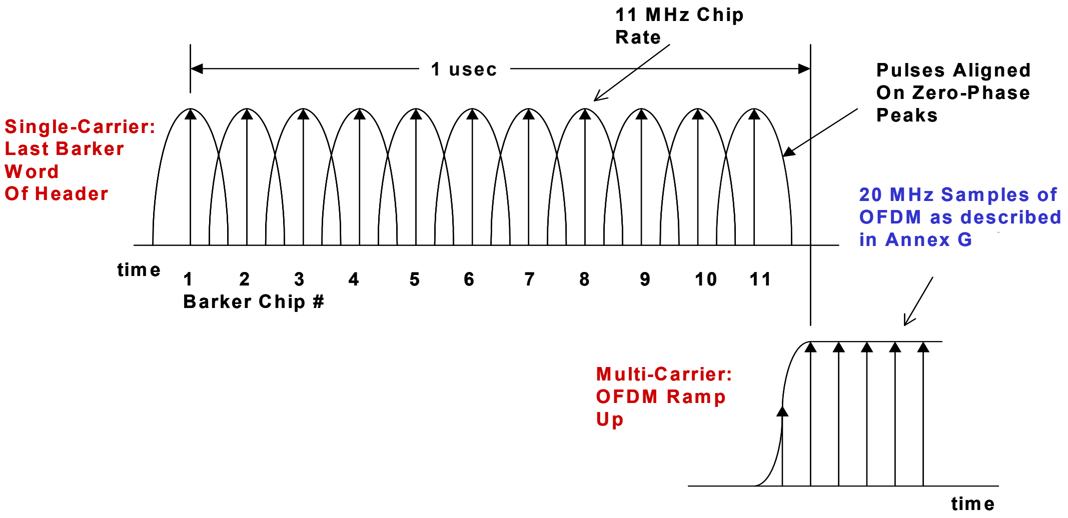

11 MHz Chip Rate
1 usec
Pulses Aligned On Zero-Phase Peaks
Single-Carrier: Last Barker Word Of Header
20 MHz Samples of OFDM as described in Annex G
time
1
2
3
4
5
6
7
8
9
10
11
Barker Chip #
Multi-Carrier: OFDM Ramp Up
time

Figure 19-17—Single carrier to OFDM time alignment

The first full-strength OFDM sample is sent on the 1 µs epoch boundary, as illustrated in Figure 19-17. Tapering may precede this. The peak corresponds to the first full-strength sample described in Annex L.

19.7.3.5 Single carrier termination

The single carrier segment of a packet should terminate in nominally 0.1 μs with the same type shaping described for Clause 18. This is depicted in Figure 19-18. It is not necessary to completely flush the single carrier pulse-shaping filter. This minimizes the transition time overhead. This is informative as the basic requirement is to meet the spectral mask defined in 18.3.9.3.

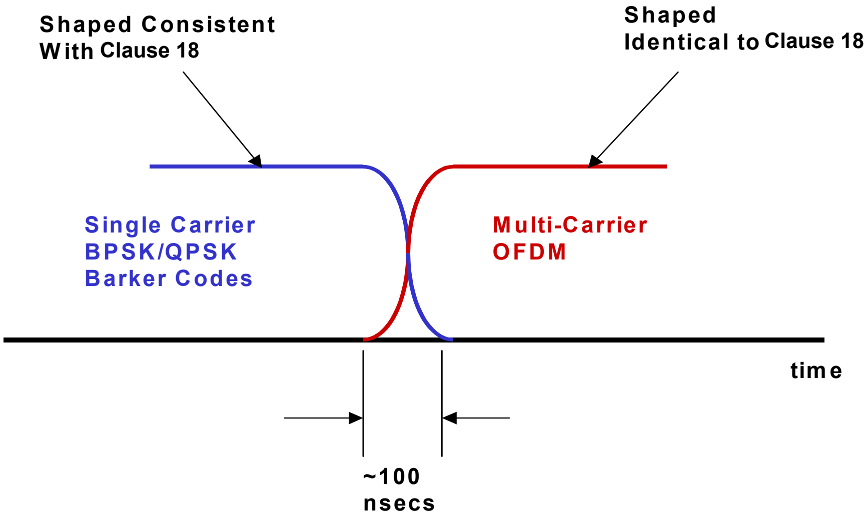


Figure 19-18—Single carrier termination requirement

This termination may be performed explicitly in the baseband processor, or it may be provided by filters in the transmit radio.

19.7.3.6 Transition carrier frequency requirement

The carrier frequency shall be coherent across the packet segments. This effect is depicted in Figure 19-19.

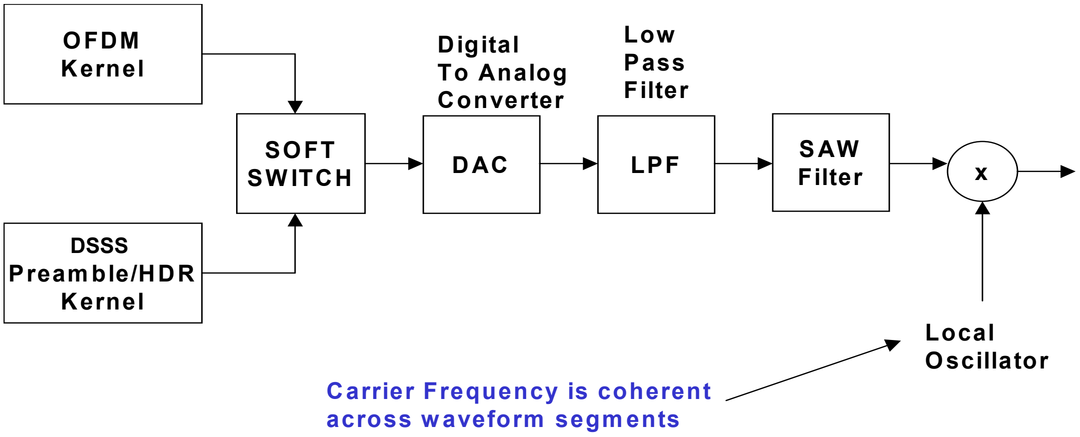


Figure 19-19—Carrier frequency coherency shall be maintained

19.7.3.7 Transition carrier phase requirement

The carrier phase shall be coherent across the single carrier to multicarrier transition. This coherency shall be differentially established relative to the phase of the last Barker symbol transmitted (the last 11 single carrier chips). The OFDM segment symbols shall be transmitted with one of four phases relative to the phase of OFDM symbols as described in Clause 18. These phases include 0, 90, 180, or 270 degrees, depending on the phase of the last Barker symbol. The phase of the first OFDM symbol (as referenced by the pilot tones) shall be 45 degrees more than the phase of the last Barker symbol. "More than" implies a clockwise rotation as shown in Figure 19-20.

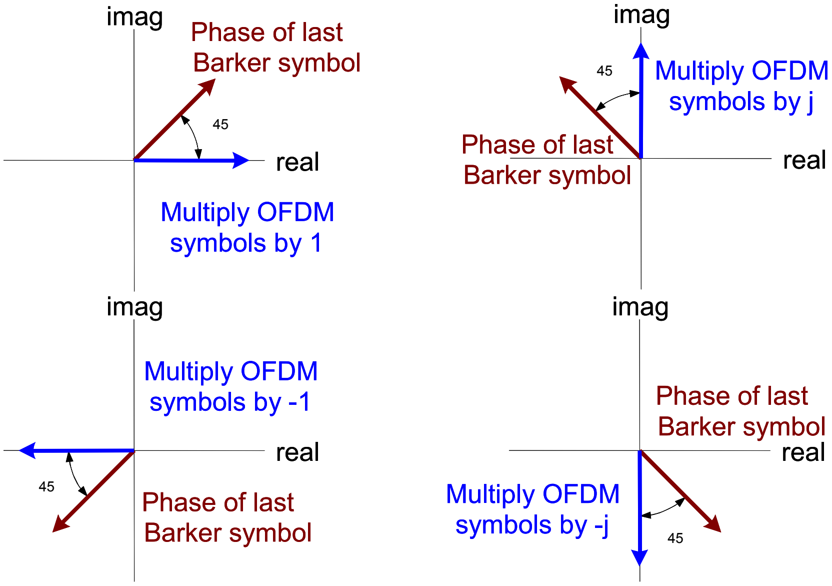


Figure 19-20—The phase of the first OFDM segment symbol is established by the last Barker symbol

In a transmit implementation using I/Q signaling, it is common to maximally energize in the I-and-Q channels concurrently for BPSK or QPSK signaling. The analog stages of the transmit radio tend to perform best with this configuration. To achieve this effect, typically the BPSK or QPSK I-and-Q alignment of the Barker symbols are at 45 degrees, 135 degrees, –135 degrees, and –45 degrees, as shown in Figure 19-21. This Barker symbol alignment is used to establish the phase of the OFDM signal.

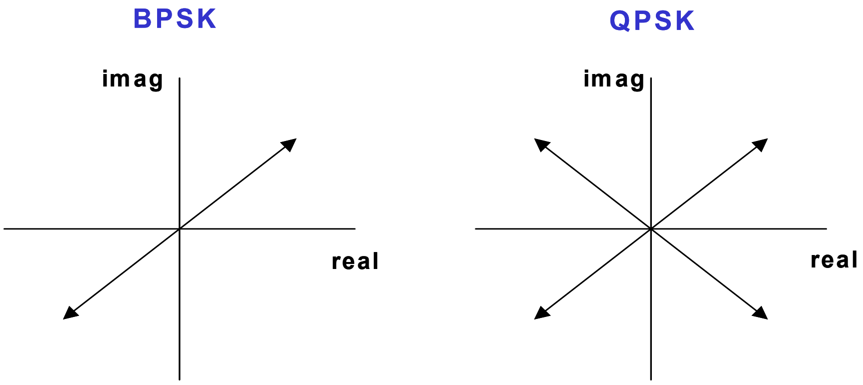


Figure 19-21—BPSK and QPSK signaling with the I/Q channels maximally energized

Figure 19-20 is a series of diagrams illustrating the phase relationship between the last Barker symbol (not the last chip) in the header and subsequent OFDM symbols. For example, if the phase of the last Barker symbol is in the first quadrant at 45 degrees, then the phase of the OFDM symbols is as described in Annex L unmodified. However, if the phase of the last Barker symbol is in the second quadrant (135 degree phase), then the phase of the OFDM symbols is rotated by +90 degrees relative to the phase of the samples in Annex L. If the phase of the last Barker symbol is in the third quadrant (–135 degree phase), then the phase of the OFDM symbols is rotated by +180 degrees relative to the phase of the samples in Annex L. If the phase of the last Barker symbol is in the fourth quadrant (–45 degree phase), then the phase of the OFDM symbols is rotated by +270 degrees relative to the phase of the samples in Annex L.

If the transmitter generates the Barker symbols at some other angular relationship to the I/Q axes, then the OFDM symbols shall be transmitted at a phase 45 degrees more than the phase of the last 11-chip Barker symbol.

19.7.3.8 Transmit modulation accuracy requirement

The preceding subclauses establish transmit modulation requirements without mention of required accuracy. The accuracy is as described in 18.3.9.8.

The required accuracy for a given transmit packet is data rate dependent. The packet accuracy is set by the data rate of the OFDM portion of the packet. The preamble and header are transmitted with the same fidelity requirement as the fidelity requirement levied on the OFDM portion of the packet. For the single carrier portion of the packet, the EVM is interpreted as normalized mean-squared error.

19.8 ERP PLME

19.8.1 PLME SAP

Table 19-7 lists the additional MIB attributes that may be accessed by the PHY sublayer entities and the intralayer of higher LMEs. These attributes are accessed via the PLME_GET, PLME_SET, PLME_RESET, and PLME_CHARACTERISTICS primitives defined in 6.5.

19.8.2 MIB

High Rate PHY MIB attributes are defined in Annex C with additions from this supplement and with specific values defined in Table 19-7.

Table 19-7—MIB attribute default values/ranges

Managed object	Default value/range	Operational semantics
dot11 PHY Operation Table		
dot11PHYtype	ERP (X'06')	Static
dot11CurrentRegDomain	Implementation dependent	Static
dot11 PHY Antenna Table		
dot11CurrentTxAntenna	Implementation dependent	Dynamic
dot11DiversitySupportImplemented	Implementation dependent	Static
dot11CurrentRxAntenna	Implementation dependent	Dynamic

Table 19-7—MIB attribute default values/ranges *(continued)*

Managed object	Default value/range	Operational semantics
dot11 PHY Tx Power Table		
dot11NumberSupportedPowerLevelsImplemented	Implementation dependent	Static
dot11TxPowerLevel1	Implementation dependent	Static
dot11TxPowerLevel2	Implementation dependent	Static
dot11TxPowerLevel3	Implementation dependent	Static
dot11TxPowerLevel4	Implementation dependent	Static
dot11TxPowerLevel5	Implementation dependent	Static
dot11TxPowerLevel6	Implementation dependent	Static
dot11TxPowerLevel7	Implementation dependent	Static
dot11TxPowerLevel8	Implementation dependent	Static
dot11CurrentTxPowerLevel	Implementation dependent	Dynamic
dot11 Phy DSSS Table		
dot11CurrentChannel	Implementation dependent	Dynamic
dot11 Reg Domains Supported Table		
dot11RegDomainsImplementedValue(s)	Implementation dependent	Static
dot11 PHY Antennas List Table		
dot11TxAntennaImplemented	Implementation dependent	Static
dot11RxAntennaImplemented	Implementation dependent	Static
dot11DiversitySelectionRxImplemented	Implementation dependent	Dynamic
dot11 Supported Data Rates Tx Table		
dot11SupportedDataratesTxValue	X'02' = 1 Mb/s X'04' = 2 Mb/s X'0B' = 5.5 Mb/s X'16' = 11 Mb/s X'0C' = 6 Mb/s X'12' = 9 Mb/s X'18' = 12 Mb/s X'24' = 18 Mb/s X'2C = 22 Mb/s X'30' = 24 Mb/s X'42 = 33 Mb/s X'48' = 36 Mb/s X'60' = 48 Mb/s X'6C' = 54 Mb/s	Static

Table 19-7—MIB attribute default values/ranges *(continued)*

Managed object	Default value/range	Operational semantics
dot11 Supported Data Rates Rx Table		
dot11ImplementedDataRatesRxValue	X'02' = 1 Mb/s X'04' = 2 Mb/s X'0B' = 5.5 Mb/s X'16' = 11 Mb/s X'0C' = 6 Mb/s X'12' = 9 Mb/s X'18' = 12 Mb/s X'24' = 18 Mb/s X'2C = 22 Mb/s X'30' = 24 Mb/s X'42 = 33 Mb/s X'48' = 36 Mb/s X'60' = 48 Mb/s X'6C' = 54 Mb/s	Static
dot11 HRDSSS PHY Table		
dot11ShortPreambleOptionImplemented	true/Boolean	Static
dot11PBCCOptionImplemented	Implementation dependent	Static
dot11ChannelAgilityPresent	Implementation dependent	Static
dot11ChannelAgilityActivated	false/Boolean	Dynamic
dot11 PHY ERP Table		
dot11ERPPBCCOptionImplemented	false/Boolean	Static
dot11DSSSOFDMOptionImplemented	false/Boolean	Static
dot11DSSSOFDMOptionActivated	false/Boolean	Dynamic
dot11ShortSlotTimeOptionImplemented	false/Boolean	Static
dot11ShortSlotTimeOptionActivated	false/Boolean	Dynamic

19.8.3 TXTIME

19.8.3.1 General

The value of TXTIME is calculated for each modulation type based on parameters in the TXVECTOR. For the 1, 2, 5.5, and 11 Mb/s modes with DSSS, CCK, and PBCC modulation formats, the value shall be calculated as described in 17.3.4.

19.8.3.2 ERP-OFDM TXTIME calculations

The value of the TXTIME parameter returned by the PLME_TXTIME.confirm primitive shall be calculated using the ERP-OFDM TXTIME calculation as shown in Equation (19-6).

$$\text{TXTIME} = T_{PREAMBLE} + T_{SIGNAL} + T_{SYM} \times \text{Ceiling}\left((16 + 8 \times \text{LENGTH} + 6)/N_{DBPS}\right) + \text{Signal Extension} \qquad (19\text{-}6)$$

where

$T_{PREAMBLE}$, T_{SIGNAL}, and T_{SYM} are defined in Table 18-5 in 18.3.2.4

N_{DBPS} is the number of data bits per symbol and is derived from the DATARATE parameter in Table 18-4 in 18.3.2.3

Ceiling is a function that returns the smallest integer value greater than or equal to its argument value

Signal Extension is 6 μs

19.8.3.3 ERP-PBCC TXTIME calculations

The value of the TXTIME parameter returned by the PLME_TXTIME.confirm primitive shall be calculated according to the following:

For PBCC 5.5 Mb/s and 11 Mb/s, see 17.3.4.

For ERP-PBCC-22 Mb/s, use Equation (19-7).

$$\text{TXTIME} = \text{PreambleLength} + \text{PLCPHeaderTime} + \text{Ceiling}(((\text{LENGTH}+\text{PBCC}) \times 8) / \text{DATARATE}) \tag{19-7}$$

For ERP-PBCC-33 Mb/s, use Equation (19-8).

$$\text{TXTIME} = \text{PreambleLength} + \text{PLCPHeaderTime} + \text{Ceiling}(((\text{LENGTH}+\text{PBCC}) \times 8) / \text{DATARATE}) + \text{ClkSwitchTime} \tag{19-8}$$

where

LENGTH and DATARATE are values from the TXVECTOR parameter of the corresponding PLME_TXTIME request primitive

PBCC has a value of 1 if the SIGNAL value from the TXVECTOR parameter specifies ERP-PBCC and has a value of 0 otherwise

PreambleLength is 144 μs if the PREAMBLE_TYPE value from the TXVECTOR parameter indicates “LONGPREAMBLE” or 72 μs if the PREAMBLE_TYPE value from the TXVECTOR parameter indicates “SHORTPREAMBLE”

PLCPHeaderTime is 48 μs if the PREAMBLE_TYPE value from the TXVECTOR parameter indicates “LONGPREAMBLE” or 24 μs if the PREAMBLE_TYPE value from the TXVECTOR parameter indicates “SHORTPREAMBLE”

LENGTH is in units of octets

DATARATE is in units of Mb/s

ClkSwitchTime is defined as 1 μs

Ceiling is a function that returns the smallest integer value greater than or equal to its argument value

19.8.3.4 DSSS-OFDM TXTIME calculations

The value of the TXTIME parameter returned by the PLME_TXTIME.confirm primitive shall be calculated according to Equation (19-9):

$$\text{TXTIME} = \text{PreambleLengthDSSS} + \text{PLCPHeaderTimeDSSS} + \text{PreambleLengthOFDM} + \text{PLCPSignalOFDM} + 4 \times \text{Ceiling}((\text{PLCPServiceBits} + 8 \times (\text{NumberOfOctets}) + \text{PadBits}) / N_{DBPS}) + \text{SignalExtension} \tag{19-9}$$

where

PreambleLengthDSSS is 144 μs if the PREAMBLE_TYPE value from the TXVECTOR parameter indicates “LONGPREAMBLE,” or 72 μs if the PREAMBLE_TYPE value from the TXVECTOR parameter indicates “SHORTPREAMBLE”

PLCPHeaderTimeDSSS is 48 µs if the PREAMBLE_TYPE value from the TXVECTOR parameter indicates “LONGPREAMBLE,” or 24 µs if the PREAMBLE_TYPE value from the TXVECTOR parameter indicates “SHORTPREAMBLE”
Ceiling is a function that returns the smallest integer value greater than or equal to its argument value
PreambleLengthOFDM is 8 µs
PLCPSignalOFDM is 4 µs
PLCPServiceBits is 16 bits
NumberOfOctets is the number of data octets in the PSDU
PadBits is 6 bits
SignalExtension is 6 µs
N_{DBPS} is the number of data bits per OFDM symbol

19.8.4 ERP-OFDM PLCP PSDU definition

The DSSS PHY characteristics in Table 19-8 shall be used for the ERP for the purposes of MAC timing calculations.

Table 19-8—ERP characteristics

Characteristic	Value
aSlotTime	Long = 20 µs, short = 9 µs
aSIFSTime	10 µs
aCCATime	<15 µs for long slot time or <4 µs for Short Slot Time, see 19.4.7
aPHY-RX-START-Delay	24 µs for ERP-OFDM, 192 µs for ERP-DSSS/CCK with long preamble, and 96 µs for ERP-DSSS/CCK with short preamble
aRxTxTurnaroundTime	<5 µs
aTxRxTurnaroundTime	<10 µs
aTxPLCPDelay	Implementation dependent as long as the requirements of aRxTxTurnaroundTime are met.
aRxPLCPDelay	Implementation dependent as long as the requirements of aSIFSTime and aCCATime are met.
aRxTxSwitchTime	<<1 µs
aTxRampOnTime	Implementation dependent as long as the requirements of aRxTxTurnaroundTime are met.
aTxRampOffTime	Implementation dependent as long as the requirements of aSIFSTime are met.
ATxRFDelay	Implementation dependent as long as the requirements of aRxTxTurnaroundTime are met.
ARxRFDelay	Implementation dependent as long as the requirements of aSIFSTime and aCCATime are met.
aAirPropagationTime	<<1 µs
aMACProcessingDelay	<2 µs
aPreambleLength	20 µs
aPLCPHeaderLength	4 µs
aMPDUMaxLength	4095
aCWmin(0)	31

Table 19-8—ERP characteristics *(continued)*

Characteristic	Value
aCWmin(1)	15
ACWmin	The set aCWmin()
aCWmax	1023

The slot time shall be 20 µs, unless the BSS consists only of ERP STAs that support the Short Slot Time option. STAs indicate support for a short slot time by setting the Short Slot Time subfield to 1 when transmitting Association Request and Reassociation Request MMPDUs. If the BSS consists of only ERP STAs that support the Short Slot Time option, an optional 9 µs slot time may be used. APs indicate usage of a 9 µs slot time by setting the Short Slot Time subfield to 1 in all Beacon, Probe Response, Association Response, and Reassociation MMPDU transmissions as described in 8.4.1.4. STAs shall use short slot if the BSS indicates short slot.

19.9 Extended rate PMD sublayer

19.9.1 Scope and field of application

This subclause describes the PMD services provided to the PLCP for the ERP.

19.9.2 Overview of service

The ERP sublayer accepts PLCP sublayer service primitives and provides the actual means by which data are transmitted or received from the medium. The combined functions of the Extended Rate PMD sublayer primitives and parameters for the receive function result in a data stream, timing information, and associated received signal parameters being delivered to the PLCP sublayer. A similar functionality is provided for data transmission.

19.9.3 Overview of Interactions

The primitives associated with the PLCP sublayer to the ERP fall into two basic categories, as follows:

a) Service primitives that support PLCP peer-to-peer interactions

b) Service primitives that have local significance and that support sublayer-to-sublayer interactions

19.9.4 Basic service and options

19.9.4.1 General

All of the service primitives described in this subclause are considered mandatory, unless otherwise specified.

19.9.4.2 PMD_SAP peer-to-peer service primitives

Table 19-9 indicates the primitives for peer-to-peer interactions.

Table 19-9—PMD_SAP peer-to-peer services

Primitive	Request	Indicate	Confirm	Response
PMD_Data	X	X		

19.9.4.3 PMD_SAP sublayer-to-sublayer service primitives

Table 19-10 indicates the primitives for sublayer-to-sublayer interactions.

Table 19-10—PMD_SAP sublayer-to-sublayer services

Primitive	Request	Indicate	Confirm	Response
PMD_TXSTART	X			
PMD_TXEND	X			
PMD_ANTSEL	X			
PMD_TXPWRLVL	X			
PMD_MODULATION	X			
PMD_PREAMBLE	X			
PMD_RATE	X			
PMD_RSSI		X		
PMD_SQ		X		
PMD_CS		X		
PMD_ED		X		
PMD_RCPI		X		

19.9.4.4 PMD_SAP service primitive parameters

Table 19-11 shows the parameters used by one or more of the PMD_SAP service primitives.

Table 19-11—List of parameters for the PMD primitives

Parameter	Associated primitive	Value	Description
TXD_UNIT	PMD_DATA.request	0 to $2^n - 1$, where n is the number of bits per symbol for the modulation and rate specified in PMD_ MODULATION.request and PMD_RATE.request primitives.	This parameter represents a single block of data, which, in turn, is used by the PMD to be encoded into a transmitted symbol.

Table 19-11—List of parameters for the PMD primitives *(continued)*

Parameter	Associated primitive	Value	Description
RXD_UNIT	PMD_DATA.indication	0 to $2^n - 1$, where n is the number of bits per symbol for the modulation and rate specified in PMD_ MODULATION.request and PMD_RATE.request primitives.	This parameter represents a single symbol that has been demodulated by the PMD entity.
MODULATION	PMD_MODULATION.request	ERP-DSSS, ERP-CCK, PBCC, ERP-PBCC, ERP-OFDM, DSSS-OFDM	The MODULATION parameter specifies to the PMD layer, which ERP modulation format is for transmission of the PSDU portion of the PPDU.
PREAMBLE	PMD_PREAMBLE.request	0 for long, 1 for short	PREAMBLE selects which of the ERP preamble types is used for PLCP transmission, when applicable. It is not applicable to ERP-OFDM format.
ANT_STATE	PMD_ANTSEL.request	1 to 256	ANT_STATE selects which of the available antennas is used for transmission. The number of available antennas is determined from the MIB table parameters.
TXPWR_LEVEL	PMD_TXPWRLVL.request	1–8 (max of 8 levels)	TXPWR_LEVEL selects which of the optional transmit power levels should be used for the current PPDU transmission. The number of available power levels is determined from the MIB table parameters.
RATE	PMD_RATE.request	X'0A' for 1 Mb/s X'14' for 2 Mb/s X'37' for 5.5 Mb/s X'6E' for 11 Mb/s X'DC' for 22 Mb/s X'21' for 33 Mb/s X'75' for 12 Mb/s BPSK X'E7' for 24 Mb/s QPSK X'4B' for 48 Mb/s 16 QAM X'AA' for 72 Mb/s 64QAM	RATE selects which of the ERP data rates is used for PSDU transmission. Note that the OFDM rates are the raw, uncoded rates as in 18.3.7 and 18.5.5 and represent the rates existing at this interface.
RSSI	PMD_RSSI.indication	8 bits of RSSI (256 levels)	The RSSI is a measure of the RF energy received. Mapping of the RSSI values to actual received power is implementation dependent. See 19.9.5.11.
SQ	PMD_SQ.indication	8 bits of SQ	This parameter is a measure of the signal quality received by the ERP during the PLCP preamble and header. It is not applicable to ERP-OFDM format. See 19.9.5.12.

Table 19-11—List of parameters for the PMD primitives *(continued)*

Parameter	Associated primitive	Value	Description
CS	PMD_CS.indication	0 for DISABLED1 for ENABLED	The PMD_CS (preamble detect) primitive, in conjunction with the PMD_ED, provides the CCA status through the PLCP layer PHY-CCA primitive. PMD_CS indicates a binary status of ENABLED or DISABLED. PMD_CS is ENABLED upon detection of Barker code or OFDM sync signals. PMD_CS is DISABLED otherwise.
ED	PMD_ED.indication	0 for DISABLED1 for ENABLED	The PMD_ED primitive, along with the PMD_SQ, provides CCA status at the PLCP layer through the PHY-CCA primitive. PMD_ED indicates a binary status of ENABLED or DISABLED. PMD_ED is ENABLED when the RSSI indicated in the PMD_RSSI is greater than the detection threshold. PMD_ED is DISABLED otherwise.
RCPI	PMD_RCPI.indication	0–255	The RCPI is a measure of the received channel power. See 19.9.5.15.

19.9.5 PMD_SAP detailed service specification

19.9.5.1 Introduction

Subclauses 19.9.5.2 to 19.9.5.14 describe the services provided by each PMD primitive.

19.9.5.2 PMD_DATA.request

This primitive is the same as that defined in 18.5.5.2 and 17.4.5.2 except that the parameter TXD_UNIT is expanded in scope to reflect the supported modulation formats of ERP as defined in 19.9.4.4.

19.9.5.3 PMD_DATA.indication

This primitive is the same as that defined in 18.5.5.3 and 17.4.5.3 except that the parameter RXD_UNIT is expanded in scope to reflect the supported modulation formats of ERP as defined in 19.9.4.4.

19.9.5.4 PMD_MODULATION.request

This primitive is the same as that defined in 17.4.5.4 except that the parameter MODULATION is expanded in scope to reflect the supported modulation formats of ERP as defined in 19.9.4.4.

19.9.5.5 PMD_PREAMBLE.request

This primitive is the same as that defined in 17.4.5.5, including the definition of the parameter PREAMBLE. This primitive is not used in association with transmission of ERP-OFDM modulations.

19.9.5.6 PMD_TXSTART.request

This primitive is the same as that defined in 18.5.5.4 and 17.4.5.7.

19.9.5.7 PMD_TXEND.request

This primitive is the same as that defined in 18.5.5.5 and 17.4.5.8.

19.9.5.8 PMD_ANTSEL.request

This primitive is the same as that defined in 17.4.5.9, including the definition of the parameter ANT_STATE.

19.9.5.9 PMD_TXPRWLVL.request

This primitive is the same as that defined in 18.5.5.6, including the definition of the parameter TXPWR_LEVEL.

19.9.5.10 PMD_RATE.request

This primitive is the same as that defined in 18.5.5.7 and 17.4.5.11, except that the parameter RATE is expanded in scope to reflect the supported ERP transmission rates as defined in 19.9.4.4.

19.9.5.11 PMD_RSSI.indication

This primitive is the same as that defined in 18.5.5.8 and 17.4.5.12, including the parameter RSSI. This primitive is used to aid in link optimization algorithms such as roaming decisions.

19.9.5.12 PMD_SQ.indication

This primitive is the same as that defined in 17.4.5.13, including the parameter SQ. This primitive is not used in association with reception of ERP-OFDM modulations. This primitive is used to aid in link optimization algorithms such as roaming decisions.

19.9.5.13 PMD_CS.indication

This primitive is the same as that defined in 17.4.5.14, except that its use is expanded for use with all ERP modulation types as described in 19.3.5.

19.9.5.14 PMD_ED.indication

This primitive is the same as that defined in 17.4.5.15, except that its use is expanded for use with all ERP modulation types as described in 19.3.5.

19.9.5.15 PMD_RCPI.indication

This primitive is the same as that defined in 18.5.5.9 and 17.4.5.17, including the parameter RCPI. This primitive is used for radio measurement purposes and to aid in link optimization algorithms such as roaming decisions.

20. High Throughput (HT) PHY specification

20.1 Introduction

20.1.1 Introduction to the HT PHY

Clause 20 specifies the PHY entity for a high throughput (HT) orthogonal frequency division multiplexing (OFDM) system.

In addition to the requirements found in Clause 20, an HT STA shall be capable of transmitting and receiving frames that are compliant with the mandatory PHY specifications defined as follows:

— In Clause 18 when the HT STA is operating in a 20 MHz channel width in the 5 GHz band

— In Clause 17 and Clause 19 when the HT STA is operating in a 20 MHz channel width in the 2.4 GHz band

The HT PHY is based on the OFDM PHY defined in Clause 18, with extensibility up to four spatial streams, operating in 20 MHz bandwidth. Additionally, transmission using one to four spatial streams is defined for operation in 40 MHz bandwidth. These features are capable of supporting data rates up to 600 Mb/s (four spatial streams, 40 MHz bandwidth).

The HT PHY data subcarriers are modulated using binary phase shift keying (BPSK), quadrature phase shift keying (QPSK), 16-quadrature amplitude modulation (16-QAM), or 64-QAM. Forward error correction (FEC) coding (convolutional coding) is used with a coding rate of 1/2, 2/3, 3/4, or 5/6. LDPC codes are added as an optional feature.

Other optional features at both transmit and receive sides are 400 ns short guard interval (GI), transmit beamforming, HT-greenfield format, and STBC.

An HT non-AP STA shall support all equal modulation (EQM) rates for one spatial stream (MCSs 0 to 7) using 20 MHz channel width. An HT AP shall support all EQM rates for one and two spatial streams (MCSs 0 to 15) using 20 MHz channel width.

The maximum HT PSDU length is 65 535 octets.

20.1.2 Scope

The services provided to the MAC by the HT PHY consist of two protocol functions, defined as follows:

a) A PHY convergence function, which adapts the capabilities of the physical medium dependent (PMD) system to the PHY service. This function is supported by the physical layer convergence procedure (PLCP), which defines a method of mapping the PSDUs into a framing format (PPDU) suitable for sending and receiving PSDUs between two or more STAs using the associated PMD system.

b) A PMD system whose function defines the characteristics and method of transmitting and receiving data through a wireless medium between two or more STAs. Depending on the PPDU format, these STAs support a mixture of HT PHY and Clause 16, Clause 18, Clause 17, or Clause 19 PHYs.

20.1.3 HT PHY functions

20.1.3.1 General

The HT PHY contains three functional entities: the PHY convergence function (i.e., the PLCP), the PMD function, and the layer management function (i.e., the PLME). Each of these functions is described in detail in 20.3 through 20.5.

The HT PHY service is provided to the MAC through the PHY service primitives defined in Clause 7.

20.1.3.2 HT PLCP sublayer

In order to allow the MAC to operate with minimum dependence on the PMD sublayer, a PHY convergence sublayer is defined (i.e., the PLCP). The PLCP sublayer simplifies the PHY service interface to the MAC services.

20.1.3.3 HT PMD sublayer

The HT PMD sublayer provides a means to send and receive data between two or more STAs. This clause is concerned with the 2.4 GHz and 5 GHz frequency bands using HT OFDM modulation.

20.1.3.4 PHY management entity (PLME)

The PLME performs management of the local PHY functions in conjunction with the MLME.

20.1.3.5 Service specification method

The models represented by figures and state diagrams are intended to be illustrations of the functions provided. It is important to distinguish between a model and a real implementation. The models are optimized for simplicity and clarity of presentation; the actual method of implementation is left to the discretion of the HT-PHY-compliant developer. The service of a layer or sublayer is the set of capabilities that it offers to a user in the next higher layer (or sublayer). Abstract services are specified here by describing the service primitives and parameters that characterize each service. This definition is independent of any particular implementation.

20.1.4 PPDU formats

The structure of the PPDU transmitted by an HT STA is determined by the TXVECTOR FORMAT, CH_BANDWIDTH, CH_OFFSET, and MCS parameters as defined in Table 20-1. The effect of the CH_BANDWIDTH, CH_OFFSET, and MCS parameters on PPDU format is described in 20.2.3.

The FORMAT parameter determines the overall structure of the PPDU as follows:

- *Non-HT format* **(NON_HT):** Packets of this format are structured according to the Clause 18 (OFDM) or Clause 19 (ERP) specification. Support for non-HT format is mandatory.
- *HT-mixed format* **(HT_MF):** Packets of this format contain a preamble compatible with Clause 18 and Clause 19 receivers. The non-HT-STF (L-STF), the non-HT-LTF (L-LTF), and the non-HT SIGNAL field (L-SIG) are defined so they can be decoded by non-HT Clause 18 and Clause 19 STAs. The rest of the packet cannot be decoded by Clause 18 or Clause 19 STAs. Support for HT-mixed format is mandatory.
- *HT-greenfield format* **(HT_GF):** HT packets of this format do not contain a non-HT compatible part. Support for HT-greenfield format is optional. An HT STA that does not support the reception of an HT-greenfield format packet shall be able to detect that an HT-greenfield format packet is an HT transmission (as opposed to a non-HT transmission). In this case, the receiver shall decode the HT-SIG and determine whether the HT-SIG cyclic redundancy check (CRC) passes.

20.2 HT PHY service interface

20.2.1 Introduction

The PHY interfaces to the MAC through the TXVECTOR, TXSTATUS, RXVECTOR, and PHYCONFIG_VECTOR. The TXVECTOR supplies the PHY with per-packet transmit parameters. Status of the transmission is reported from PHY to MAC by parameters within TXSTATUS. Using the RXVECTOR, the PHY informs the MAC of the received packet parameters. Using the PHYCONFIG_VECTOR, the MAC configures the PHY for operation, independent of frame transmission or reception.

This interface is an extension of the generic PHY service interface defined in 7.3.4.

20.2.2 TXVECTOR and RXVECTOR parameters

The parameters in Table 20-1 are defined as part of the TXVECTOR parameter list in the PHY-TXSTART.request primitive and/or as part of the RXVECTOR parameter list in the PHY-RXSTART.indication primitive.

Table 20-1—TXVECTOR and RXVECTOR parameters

Parameter	**Condition**	**Value**	**TXVECTOR**	**RXVECTOR**
			See NOTE 1	
FORMAT		Determines the format of the PPDU. Enumerated type: NON_HT indicates Clause 16, Clause 18, Clause 17, or Clause 19 PPDU formats or non-HT duplicated PPDU format. In this case, the modulation is determined by the NON_HT_MODULATION parameter. HT_MF indicates HT-mixed format. HT_GF indicates HT-greenfield format.	Y	Y
NON_HT_MODULATION	FORMAT is NON_HT	Enumerated type: ERP-DSSS ERP-CCK ERP-OFDM ERP-PBCC DSSS-OFDM OFDM NON_HT_DUP_OFDM	Y	Y
	Otherwise	Not present		
L_LENGTH	FORMAT is NON_HT	Indicates the length of the PSDU in octets in the range of 1 to 4095. This value is used by the PHY to determine the number of octet transfers that occur between the MAC and the PHY.	Y	Y
	FORMAT is HT_MF	Indicates the value in the Length field of the L-SIG in the range of 1 to 4095. This use is defined in 9.23.4. This parameter may be used for the protection of more than one PPDU as described in 9.23.5.	Y	Y
	FORMAT is HT_GF	Not present	N	N

Table 20-1—TXVECTOR and RXVECTOR parameters *(continued)*

Parameter	Condition	Value	TXVECTOR	RXVECTOR
			See NOTE 1	
L_DATARATE	FORMAT is NON_HT	Indicates the rate used to transmit the PSDU in megabits per second. Allowed values depend on the value of the NON_HT_MODULATION parameter as follows: ERP-DSSS: 1 and 2 ERP-CCK: 5.5 and 11 ERP-PBCC: 5.5, 11, 22, and 33 DSSS-OFDM, ERP-OFDM, NON_HT_DUP_OFDM: 6, 9, 12, 18, 24, 36, 48, and 54 OFDM: 6, 9, 12, 18, 24, 36, 48, and 54	Y	Y
	FORMAT is HT_MF	Indicates the data rate value that is in the L-SIG. This use is defined in 9.23.4.	Y	Y
	FORMAT is HT_GF	Not present	N	N
LSIGVALID	FORMAT is HT_MF	True if L-SIG Parity is valid False if L-SIG Parity is not valid	N	Y
	Otherwise	Not present	N	N
SERVICE	FORMAT is NON_HT and NON_HT_MODULATION is one of — DSSS-OFDM — ERP-OFDM — OFDM	Scrambler initialization, 7 null bits + 9 reserved null bits	Y	N
	FORMAT is HT_MF or HT_GF	Scrambler initialization, 7 null bits + 9 reserved null bits	Y	N
	Otherwise	Not present	N	N
TXPWR_LEVEL		The allowed values for the TXPWR_LEVEL parameter are in the range from 1 to 8. This parameter is used to indicate which of the available TxPowerLevel attributes defined in the MIB shall be used for the current transmission.	Y	N
RSSI		The allowed values for the RSSI parameter are in the range from 0 to RSSI maximum. This parameter is a measure by the PHY of the power observed at the antennas used to receive the current PPDU. RSSI shall be measured during the reception of the PLCP preamble. In HT-mixed format, the reported RSSI shall be measured during the reception of the HT-LTFs. RSSI is intended to be used in a relative manner, and it shall be a monotonically increasing function of the received power.	N	Y

Table 20-1—TXVECTOR and RXVECTOR parameters *(continued)*

Parameter	Condition	Value	TXVECTOR	RXVECTOR
			See NOTE 1	
PREAMBLE_TYPE	FORMAT is NON_HT and NON_HT_MODULATION is one of — ERP-DSSS — ERP-CCK — ERP-PBCC — DSSS-OFDM	Enumerated type: SHORTPREAMBLE LONGPREAMBLE	Y	Y
	Otherwise	Not present	N	N
MCS	FORMAT is HT_MF or HT_GF	Selects the modulation and coding scheme used in the transmission of the packet. The value used in each MCS is the index defined in 20.6. Integer: range 0 to 76. Values of 77 to 127 are reserved. The interpretation of the MCS index is defined in 20.6.	Y	Y
	Otherwise	Not present	N	N
REC_MCS	FORMAT is HT_MF or HT_GF	Indicates the MCS that the STA's receiver recommends.	N	O
	Otherwise	Not present	N	N
CH_BANDWIDTH	FORMAT is HT_MF or HT_GF	Indicates whether the packet is transmitted using 40 MHz or 20 MHz channel width. Enumerated type: HT_CBW20 for 20 MHz and 40 MHz upper and 40 MHz lower modes HT_CBW40 for 40 MHz	Y	Y
	FORMAT is NON_HT	Enumerated type: NON_HT_CBW40 for non-HT duplicate format NON_HT_CBW20 for all other non-HT formats	Y	Y
CH_OFFSET		Indicates which portion of the channel is used for transmission. Refer to Table 20-2 for valid combinations of CH_OFFSET and CH_BANDWIDTH. Enumerated type: CH_OFF_20 indicates the use of a 20 MHz channel (that is not part of a 40 MHz channel). CH_OFF_40 indicates the entire 40 MHz channel. CH_OFF_20U indicates the upper 20 MHz of the 40 MHz channel CH_OFF_20L indicates the lower 20 MHz of the 40 MHz channel.	Y	N
LENGTH	FORMAT is HT_MF or HT_GF	Indicates the length of an HT PSDU in the range of 0 to 65 535 octets. A value of 0 indicates a NDP that contains no data symbols after the HT preamble (see 20.3.9).	Y	Y
	Otherwise	Not present	N	N

Table 20-1—TXVECTOR and RXVECTOR parameters *(continued)*

Parameter	Condition	Value	TXVECTOR	RXVECTOR
			See NOTE 1	
SMOOTHING	FORMAT is HT_MF or HT_GF	Indicates whether frequency-domain smoothing is recommended as part of channel estimation. (See NOTE 2.) Enumerated type: SMOOTHING_REC indicates that smoothing is recommended. SMOOTHING_NOT_REC indicates that smoothing is not recommended.	Y	Y
	Otherwise	Not present	N	N
SOUNDING	FORMAT is HT_MF or HT_GF	Indicates whether this packet is a sounding packet. Enumerated type: SOUNDING indicates this is a sounding packet. NOT_SOUNDING indicates this is not a sounding packet.	Y	Y
	Otherwise	Not present	N	N
AGGREGATION	FORMAT is HT_MF or HT_GF	Indicates whether the PSDU contains an A-MPDU. Enumerated type: AGGREGATED indicates this packet has A-MPDU aggregation. NOT_AGGREGATED indicates this packet does not have A-MPDU aggregation.	Y	Y
	Otherwise	Not present	N	N
STBC	FORMAT is HT_MF or HT_GF	Indicates the difference between the number of space-time streams (N_{STS}) and the number of spatial streams (N_{SS}) indicated by the MCS as follows: 0 indicates no STBC ($N_{STS} = N_{SS}$). 1 indicates $N_{STS} - N_{SS} = 1$. 2 indicates $N_{STS} - N_{SS} = 2$. Value of 3 is reserved.	Y	Y
	Otherwise	Not present	N	N
FEC_CODING	FORMAT is HT_MF or HT_GF	Indicates which FEC encoding is used. Enumerated type: BCC_CODING indicates binary convolutional code. LDPC_CODING indicates low-density parity check code.	Y	Y
	Otherwise	Not present	N	N
GI_TYPE	FORMAT is HT_MF or HT_GF	Indicates whether a short guard interval is used in the transmission of the packet. Enumerated type: LONG_GI indicates short GI is not used in the packet. SHORT_GI indicates short GI is used in the packet.	Y	Y
	Otherwise	Not present	N	N

Table 20-1—TXVECTOR and RXVECTOR parameters *(continued)*

Parameter	Condition	Value	TXVECTOR	RXVECTOR
			See NOTE 1	
NUM_EXTEN_SS	FORMAT is HT_MF or HT_GF	Indicates the number of extension spatial streams that are sounded during the extension part of the HT training in the range of 0 to 3.	Y	Y
	Otherwise	Not present	N	N
ANTENNA_SET	FORMAT is HT_MF or HT_GF	Indicates which antennas of the available antennas are used in the transmission. The length of the field is 8 bits. A 1 in bit position *n,* relative to the LSB, indicates that antenna *n* is used. At most 4 bits out of 8 may be set to 1. This field is present only if ASEL is applied.	O	N
	Otherwise	Not present	N	N
N_TX	FORMAT is HT_MF or HT_GF	The N_TX parameter indicates the number of transmit chains.	Y	N
	Otherwise	Not present	N	N
EXPANSION_MAT	EXPANSION_MAT_TYPE is COMPRESSED_SV	Contains a set of compressed beamforming feedback matrices as defined in 20.3.12.3.6. The number of elements depends on the number of spatial streams and the number of transmit chains.	Y	N
	EXPANSION_MAT_TYPE is NON_COMPRESSED_SV	Contains a set of noncompressed beamforming feedback matrices as defined in 20.3.12.3.5. The number of complex elements is $N_{ST} \times N_r \times N_c$ where N_{ST} is the total number of subcarriers, N_c is the number of columns, and N_r is the number of rows in each matrix.	Y	N
	EXPANSION_MAT_TYPE is CSI_MATRICES	Contains a set of CSI matrices as defined in 20.3.12.3.2. The number of complex elements is $N_{ST} \times N_r \times N_c$ where N_{ST} is the total number of subcarriers, N_c is the number of columns, and N_r is the number of rows in each matrix.	Y	N
	Otherwise	Not present	N	N
EXPANSION_MAT_TYPE	EXPANSION_MAT is present	Enumerated type: COMPRESSED_SV indicates that EXPANSION_MAT is a set of compressed beamforming feedback matrices. NON_COMPRESSED_SV indicates that EXPANSION_MAT is a set of noncompressed beamforming feedback matrices. CSI_MATRICES indicates that EXPANSION_MAT is a set of channel state matrices.	Y	N
	Otherwise	Not present	N	N

Table 20-1—TXVECTOR and RXVECTOR parameters *(continued)*

Parameter	Condition	Value	TXVECTOR	RXVECTOR
			See NOTE 1	
CHAN_MAT	CHAN_MAT_TYPE is COMPRESSED_SV	Contains a set of compressed beamforming feedback matrices as defined in 20.3.12.3.6 based on the channel measured during the training symbols of the received PPDU. The number of elements depends on the number of spatial streams and the number of transmit chains.	N	Y
	CHAN_MAT_TYPE is NON_COMPRESSED_SV	Contains a set of noncompressed beamforming feedback matrices as defined in 20.3.12.3.5 based on the channel measured during the training symbols of the received PPDU. The number of complex elements is $N_{ST} \times N_r \times N_c$ where N_{ST} is the total number of subcarriers, N_c is the number of columns, and N_r is the number of rows in each matrix.	N	Y
	CHAN_MAT_TYPE is CSI_MATRICES	Contains a set of CSI matrices as defined in 20.3.12.3.2 based on the channel measured during the training symbols of the received PPDU. The number of complex elements is $N_{ST} \times N_r \times N_c$ where N_{ST} is the total number of subcarriers, N_c is the number of columns, and N_r is the number of rows in each matrix.	N	Y
	Otherwise	Not present	N	N
CHAN_MAT_TYPE	FORMAT is HT_MF or HT_GF	Enumerated type: COMPRESSED_SV indicates that CHAN_MAT is a set of compressed beamforming vector matrices. NON_COMPRESSED_SV indicates that CHAN_MAT is a set of noncompressed beamforming vector matrices. CSI_MATRICES indicates that CHAN_MAT is a set of channel state matrices.	N	Y
	Otherwise	Not present	N	N
RCPI		Is a measure of the received RF power averaged over all the receive chains in the data portion of a received frame. Refer to 20.3.21.6 for the definition of RCPI.	N	Y
SNR	CHAN_MAT_TYPE is CSI_MATRICES	Is a measure of the received SNR per chain. SNR indications of 8 bits are supported. SNR shall be the decibel representation of linearly averaged values over the tones represented in each receive chain as described in 8.4.1.27	N	Y
	CHAN_MAT_TYPE is COMPRESSED_SV or NON_COMPRESSED_SV	Is a measure of the received SNR per stream. SNR indications of 8 bits are supported. SNR shall be the sum of the decibel values of SNR per tone divided by the number of tones represented in each stream as described in 8.4.1.28 and 8.4.1.29	N	Y

Table 20-1—TXVECTOR and RXVECTOR parameters *(continued)*

Parameter	Condition	Value	TXVECTOR	RXVECTOR
			See NOTE 1	
NO_SIG_EXTN	FORMAT is HT_MF or HT_GF Or FORMAT is NON_HT and NON_HT_MODULATION is ERP-OFDM, DSSS-OFDM, or NON_HT_ DUPOFDM	Indicates whether signal extension needs to be applied at the end of transmission. Boolean values: true indicates no signal extension is present. false indicates signal extension may be present depending on other TXVECTOR parameters (see 20.2.2).	Y	N
	Otherwise	Not present	N	N
TIME_OF_DEPARTURE_REQUESTED		Enumerated type: True indicates that the MAC entity requests that the PHY PLCP entity measures and reports time of departure parameters corresponding to the time when the first frame energy is sent by the transmitting port. False indicates that the MAC entity requests that the PHY PLCP entity neither measures nor reports time of departure parameters.	O	N
RX_START_OF_FRAME_OFFSET		0 to 2^{32}– 1. An estimate of the offset (in 10 ns units) from the point in time at which the start of the preamble corresponding to the incoming frame arrived at the receive antenna port to the point in time at which this primitive is issued to the MAC.	N	O

NOTE 1—In the "TXVECTOR" and "RXVECTOR" columns, the following apply:
Y = Present; N = Not present; O = Optional
NOTE 2—Setting the smoothing bit is defined in 20.3.11.11.2.

20.2.3 Effect of CH_BANDWIDTH, CH_OFFSET, and MCS parameters on PPDU format

The structure of the PPDU transmitted by an HT STA is determined by the TXVECTOR FORMAT, CH_BANDWIDTH, CH_OFFSET, and MCS parameters as defined in Table 20-1. The effect of the FORMAT parameter is described in 20.1.4.

The operation of the PHY in the frequency domain is determined by the CH_BANDWIDTH and CH_OFFSET parameters. Table 20-2 shows the combination of CH_BANDWIDTH and CH_OFFSET parameters that are supported.

Table 20-2—PPDU format as a function of CH_BANDWIDTH and CH_OFFSET parameters

CH_BANDWIDTH	CH_OFFSET
HT_CBW20	CH_OFF_20 or CH_OFFSET is not present: *20 MHz HT format*—A STA that has a 20 MHz operating channel width transmits an HT-mixed or HT-greenfield format packet of 20 MHz bandwidth with one to four spatial streams. CH_OFF_40: *Not defined* CH_OFF_20U: *40 MHz HT upper format*—The STA transmits an HT-mixed or HT-greenfield format packet of 20 MHz bandwidth with one to four spatial streams in the upper 20 MHz of a 40 MHz channel. CH_OFF_20L: *40 MHz HT lower format*—The STA transmits an HT-mixed or HT-greenfield format packet of 20 MHz bandwidth with one to four spatial streams in the lower 20 MHz of a 40 MHz channel.
HT_CBW40	Not present: *Not defined* CH_OFF_20: *Not defined* CH_OFF_40: *40 MHz HT format*—A PPDU of this format occupies a 40 MHz channel to transmit an HT-mixed or HT-greenfield format packet of 40 MHz bandwidth with one to four spatial streams. CH_OFF_20U: *Not defined* CH_OFF_20L: *Not defined*
NON_HT_CBW20	CH_OFF_20 or CH_OFFSET is not present: *20 MHz non-HT format*—A STA that has a 20 MHz operating channel width transmits a non-HT format packet according to Clause 18 or Clause 19 operation. CH_OFF_40: Not defined CH_OFF_20U: *40 MHz non-HT upper format*—The STA transmits a non-HT packet of type ERP-DSSS, ERP-CCK, ERP-OFDM, ERP-PBCC, DSSS-OFDM, or OFDM in the upper 20 MHz of a 40 MHz channel. CH_OFF_20L: *40 MHz non-HT lower format*—The STA transmits a non-HT packet of type ERP-DSSS, ERP-CCK, ERP-OFDM, ERP-PBCC, DSSS-OFDM, or OFDM in the lower 20 MHz of a 40 MHz channel.

Table 20-2—PPDU format as a function of CH_BANDWIDTH and CH_OFFSET parameters *(continued)*

CH_BANDWIDTH	CH_OFFSET
NON_HT_CBW40	Not present: *Not defined* CH_OFF_20: *Not defined* CH_OFF_40: *Non-HT duplicate format*—The STA operates in a 40 MHz channel composed of two adjacent 20 MHz channels. The packets to be sent are in the Clause 18 format in each of the 20 MHz channels. The upper channel (higher frequency) is rotated by +90º relative to the lower channel. See 20.3.11.12. CH_OFF_20U: *Not defined* CH_OFF_20L: *Not defined*

NOTE—Support of 20 MHz non-HT format and 20 MHz HT format with one and two spatial streams is mandatory at APs. Support of 20 MHz non-HT format and 20 MHz HT format with one spatial stream is mandatory at non-AP STAs.

20.2.4 Support for NON_HT formats

When the FORMAT parameter is equal to NON_HT, the behavior of the HT PHY is defined in other clauses as shown in Table 20-3, dependent on the operational band. In this case, the PHY-TXSTART.request primitive is handled by mapping the TXVECTOR parameters as defined in Table 20-3 and following the operation as defined in the referenced clause. Likewise the PHY-RXSTART.indication primitive emitted when a NON_HT PPDU is received is defined in the referenced clauses, with mapping of RXVECTOR parameters as defined in Table 20-3.

Table 20-3—Mapping of the HT PHY parameters for NON_HT operation

HT PHY parameter	2.4 GHz operation defined by Clause 16	2.4 GHz operation defined by Clause 17	2.4 GHz operation defined by Clause 19	5.0 GHz operation defined by Clause 18
L_LENGTH	LENGTH	LENGTH	LENGTH	LENGTH
L_DATARATE	DATARATE	DATARATE	DATARATE	DATARATE
LSIGVALID	—	—	—	—
TXPWR_LEVEL	TXPWR_LEVEL	TXPWR_LEVEL	TXPWR_LEVEL	TXPWR_LEVEL
RSSI	RSSI	RSSI	RSSI	RSSI
FORMAT	—	—	—	—
PREAMBLE_TYPE	—	—	PREAMBLE_TYPE	—
NON_HT_MODULATION	—	MODULATION	MODULATION	—
SERVICE	SERVICE	SERVICE	SERVICE	SERVICE
MCS	—	—	—	—
CH_BANDWIDTH	—	—	—	—
CH_OFFSET	—	—	—	—
LENGTH	—	—	—	—

Table 20-3—Mapping of the HT PHY parameters for NON_HT operation *(continued)*

HT PHY parameter	2.4 GHz operation defined by Clause 16	2.4 GHz operation defined by Clause 17	2.4 GHz operation defined by Clause 19	5.0 GHz operation defined by Clause 18
SMOOTHING	—	—	—	—
SOUNDING	—	—	—	—
AGGREGATION	—	—	—	—
STBC	—	—	—	—
FEC_CODING	—	—	—	—
GI_TYPE	—	—	—	—
NUM_EXTEN_SS	—	—	—	—
ANTENNA_SET	—	—	—	—
EXPANSION_MAT	—	—	—	—
EXPANSION_MAT_TYPE	—	—	—	—
CHAN_MAT	—	—	—	—
CHAN_MAT_TYPE	—	—	—	—
N_TX	—	—	—	—
RCPI	RCPI	RCPI	RCPI	RCPI
REC_MCS	—	—	—	—
NO_SIG_EXTN	—	—	—	—
TIME_OF_DEPARTURE_REQUESTED	TIME_OF_DEPARTURE_REQUESTED	TIME_OF_DEPARTURE_REQUESTED	TIME_OF_DEPARTURE_REQUESTED	TIME_OF_DEPARTURE_REQUESTED
NOTE—A dash (—) in an entry above indicates that the related parameter is not present.				

Non-HT format PPDUs structured according to Clause 16, Clause 18, Clause 17, or Clause 19 are transmitted

- Within the limits of the transmit spectrum mask specified in the respective clauses, or
- As non-HT duplicate PPDUs within the limits of the 40 MHz transmit spectrum mask defined in 20.3.20.1, or
- As 20 MHz format non-HT PPDUs, within the limits of the 40 MHz transmit spectrum mask defined in 20.3.20.1, in the upper (CH_BANDWIDTH of value NON_HT_CBW20 and CH_OFFSET of value CH_OFF_20U) or lower (CH_BANDWIDTH of value NON_HT_CBW20 and CH_OFFSET of value CH_OFF_20U) 20 MHz of the 40 MHz channel

Non-HT PPDUs transmitted using the 40 MHz transmit spectrum mask are referred to as 40 MHz mask non-HT PPDUs. Refer to 10.15.9 for CCA sensing rules for transmission of 40 MHz mask non-HT PPDUs.

20.2.5 TXSTATUS parameters

The parameters listed in Table 20-4 are defined as part of the TXSTATUS parameter list in the PHY-TXSTART.confirm(TXSTATUS) service primitive.

Table 20-4—TXSTATUS parameter

Parameter	Value
TIME_OF_DEPARTURE	0 to 2^{32}– 1. The locally measured time when the first frame energy is sent by the transmitting port, in units equal to 1/ TIME_OF_DEPARTURE_ClockRate. This parameter is present only if TIME_OF_DEPARTURE_REQUESTED is true in the corresponding request.
TIME_OF_DEPARTURE_ClockRate	0 to 2^{16}– 1. The clock rate, in units of MHz, is used to generate the TIME_OF_DEPARTURE value. This parameter is present only if TIME_OF_DEPARTURE_REQUESTED is true in the corresponding request.
TX_START_OF_FRAME_OFFSET	0 to 2^{32}– 1. An estimate of the offset (in 10 ns units) from the point in time at which the start of the preamble corresponding to the frame was transmitted at the transmit antenna port to the point in time at which this primitive is issued to the MAC.

20.3 HT PLCP sublayer

20.3.1 Introduction

A convergence procedure, in which PSDUs are converted to and from PPDUs, is provided for the HT PHY in 20.3. During transmission, the PSDU is processed (i.e., scrambled and coded) and appended to the PLCP preamble to create the PPDU. At the receiver, the PLCP preamble is processed to aid in demodulation and delivery of the PSDU.

Two preamble formats are defined. For HT-mixed format operation, the preamble has a non-HT portion and an HT portion. The non-HT portion of the HT-mixed format preamble enables detection of the PPDU and acquisition of carrier frequency and timing by both HT STAs and STAs that are compliant with Clause 18 and/or Clause 19. The non-HT portion of the HT-mixed format preamble also consists of the SIGNAL field defined in Clause 18 and is thus decodable by STAs compliant with Clause 18 and Clause 19 as well as HT STAs.

The HT portion of the HT-mixed format preamble enables estimation of the MIMO channel to support demodulation of the HT data by HT STAs. The HT portion of the HT-mixed format preamble also includes the HT-SIG field, which supports HT operation. The SERVICE field is prepended to the PSDU.

For HT-greenfield operation, compatibility with Clause 18 and Clause 19 STAs is not required. Therefore, the non-HT portions of the preamble are not included in the HT-greenfield format preamble.

20.3.2 PPDU format

Two formats are defined for the PLCP: HT-mixed format and HT-greenfield format. These two formats are called *HT formats*. Figure 20-1 shows the non-HT format[42] and the HT formats. There is also an MCS 32 format (specified in 20.3.11.11.5) used for MCS 32 that provides the lowest rate in a 40 MHz channel. In

[42] The non-HT format is shown related to the terminology of this subclause. The non-HT PPDU format is defined in 18.3.3 and 18.3.2.

addition to the HT formats, there is a non-HT duplicate format (specified in 20.3.11.12) that duplicates the 20 MHz non-HT packet in two 20 MHz halves of a 40 MHz channel.

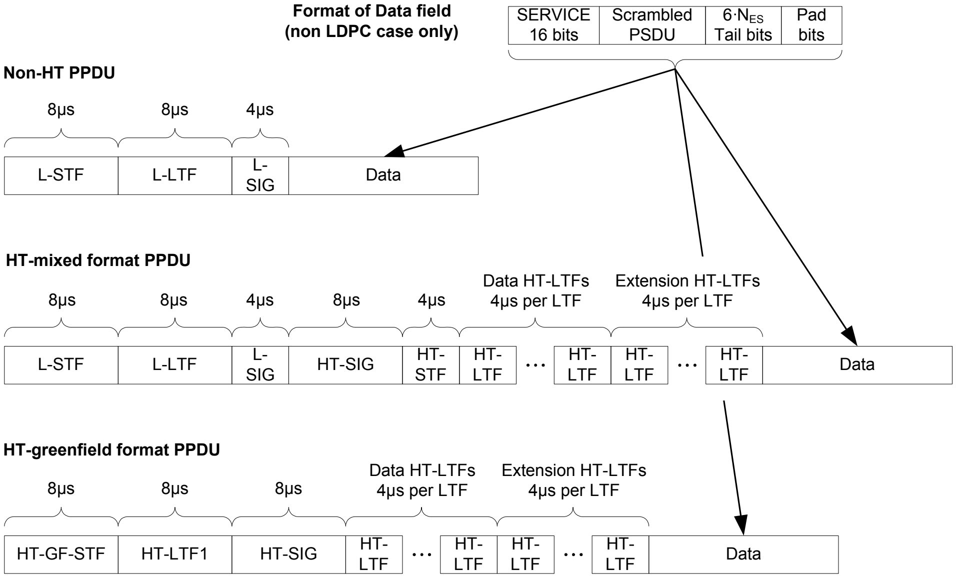


Figure 20-1—PPDU format

The elements of the PLCP packet are summarized in Table 20-5.

Table 20-5—Elements of the HT PLCP packet

Element	Description
L-STF	Non-HT Short Training field
L-LTF	Non-HT Long Training field
L-SIG	Non-HT SIGNAL field
HT-SIG	HT SIGNAL field
HT-STF	HT Short Training field
HT-GF-STF	HT-Greenfield Short Training field
HT-LTF1	First HT Long Training field (Data)
HT-LTFs	Additional HT Long Training fields (Data and Extension)
Data	The Data field includes the PSDU

The HT-SIG, HT-STF, HT-GF-STF, HT-LTF1, and HT-LTFs exist only in HT packets. In non-HT and non-HT duplicate formats only the L-STF, L-LTF, L-SIG, and Data fields exist.

In both HT-mixed format and HT-greenfield format frames, there are two types of HT-LTFs: Data HT-LTFs (HT-DLTFs) and Extension HT-LTFs (HT-ELTFs). HT-DLTFs are always included in HT PPDUs to provide the necessary reference for the receiver to form a channel estimate that allows it to demodulate the data portion of the frame. The number of HT-DLTFs, N_{HTDLTF}, may be 1, 2, or 4 and is determined by the number of space-time streams being transmitted in the frame (see Table 20-13). HT-ELTFs provide additional reference in sounding PPDUs so that the receiver can form an estimate of additional dimensions of the channel beyond those that are used by the data portion of the frame. The number of HT-ELTFs, N_{HTELTF}, may be 0, 1, 2, or 4 (see Table 20-14). PLCP preambles in which HT-DLTFs are followed by HT-ELTFs are referred to as staggered preambles. The HT-mixed format and HT-greenfield format frames shown in Figure 20-1 both contain staggered preambles for illustrative purposes.

Transmissions of frames with TXVECTOR parameter NO_SIG_EXTN equal to false are followed by a period of no transmission for a duration of aSignalExtension µs. See 9.3.8.

A Signal Extension shall be present in a transmitted PPDU, based on the parameters of the TXVECTOR, when the NO_SIG_EXTN parameter is equal to false and either of the following is true:

— The FORMAT parameter is equal to HT_MF or HT_GF.
— The FORMAT parameter is equal to NON_HT, and the NON_HT_MODULATION parameter is equal to ERP-OFDM, DSSS-OFDM, or NON_HT_DUPOFDM.

A Signal Extension shall be assumed to be present (for the purpose of timing of PHY-RXEND.indication and PHY-CCA.indication primitives, as described below and in 20.3.23) in a received PPDU when either of the following is true, based on the determined parameter values of the RXVECTOR:

— The FORMAT parameter is equal to HT_MF or HT_GF.
— The FORMAT parameter is equal to NON_HT, and the NON_HT_MODULATION parameter is equal to ERP-OFDM, DSSS-OFDM, or NON_HT_DUPOFDM.

A PPDU containing a Signal Extension is called a *signal extended PPDU*. When transmitting a signal extended PPDU, the PHY-TXEND.indication primitive shall be emitted a period of aSignalExtension µs after the end of the last symbol of the PPDU. When receiving a signal extended PPDU, the PHY-RXEND.indication primitive shall be emitted a period of aSignalExtension µs after the end of the last symbol of the PPDU.

20.3.3 Transmitter block diagram

HT-mixed format and HT-greenfield format transmissions can be generated using a transmitter consisting of the following blocks:

a) *Scrambler* scrambles the data to reduce the probability of long sequences of 0s or 1s; see 20.3.11.3.
b) *Encoder parser*, if BCC encoding is to be used, demultiplexes the scrambled bits among N_{ES} (number of BCC encoders for the Data field) BCC encoders, in a round robin manner.
c) *FEC encoders* encode the data to enable error correction. An FEC encoder may include a binary convolutional encoder followed by a puncturing device, or it may include an LDPC encoder.
d) *Stream parser* divides the outputs of the encoders into blocks that are sent to different interleaver and mapping devices. The sequence of the bits sent to an interleaver is called a *spatial stream*.
e) *Interleaver* interleaves the bits of each spatial stream (changes order of bits) to prevent long sequences of adjacent noisy bits from entering the BCC decoder. Interleaving is applied only when BCC encoding is used.
f) *Constellation mapper* maps the sequence of bits in each spatial stream to constellation points (complex numbers).
g) *STBC* encoder spreads constellation points from N_{SS} spatial streams into N_{STS} space-time streams using a space-time block code. STBC is used only when $N_{SS} < N_{STS}$; see 20.3.11.9.2.

h) *Spatial mapper* maps space-time streams to transmit chains. This may include one of the following:
   1) *Direct mapping*: Constellation points from each space-time stream are mapped directly onto the transmit chains (one-to-one mapping).
   2) *Spatial expansion*: Vectors of constellation points from all the space-time streams are expanded via matrix multiplication to produce the input to all the transmit chains.
   3) *Beamforming*: Similar to spatial expansion, each vector of constellation points from all the space-time streams is multiplied by a matrix of steering vectors to produce the input to the transmit chains.

i) *Inverse discrete Fourier transform (IDFT)* converts a block of constellation points to a time domain block.

j) *Cyclic shift (CSD) insertion* is where the insertion of the cyclic shifts prevents unintentional beamforming. CSD insertion may occur before or after the IDFT. There are three cyclic shift types as follows:
   1) A cyclic shift specified per transmitter chain with the values defined in Table 20-9 (a possible implementation is shown in Figure 20-2).
   2) A cyclic shift specified per space-time stream with the values defined in Table 20-10 (a possible implementation is shown in Figure 20-3).
   3) A cyclic shift $M_{CSD}(k)$ that may be applied as a part of the spatial mapper; see 20.3.11.11.2.

k) *GI insertion* prepends to the symbol a circular extension of itself.

l) *Windowing* optionally smooths the edges of each symbol to increase spectral decay.

Figure 20-2 and Figure 20-3 show example transmitter block diagrams. In particular, Figure 20-2 shows the transmitter blocks used to generate the HT-SIG of the HT-mixed format PPDU. These transmitter blocks are also used to generate the non-HT portion of the HT-mixed format PPDU, except that the BCC encoder and interleaver are not used when generating the L-STF and L-LTFs. Figure 20-3 shows the transmitter blocks used to generate the Data field of the HT-mixed format and HT-greenfield format PPDUs. A subset of these transmitter blocks consisting of the constellation mapper and CSD blocks, as well as the blocks to the right of, and including, the spatial mapping block, are also used to generate the HT-STF, HT-GF-STF, and HT-LTFs. The HT-greenfield format SIGNAL field is generated using the transmitter blocks shown in Figure 20-2, augmented by additional CSD and spatial mapping blocks.

20.3.4 Overview of the PPDU encoding process

The encoding process is composed of the steps described below. The following overview is intended to facilitate an understanding of the details of the convergence procedure:

a) Determine the number of transmit chains, N_{TX}, from the N_TX field of the TXVECTOR. Produce the PLCP preamble training fields for each of the N_{TX} transmit chains based on the FORMAT, NUM_EXTEN_SS, CH_BANDWIDTH, and MCS parameters of the TXVECTOR. The format and relative placement of the PLCP preamble training fields vary depending on the frame format being used, as indicated by these parameters. Apply cyclic shifts. Determine spatial mapping to be used for HT-STF and HT-LTFs in HT-mixed format frame and HT-GF-STF and HT-LTFs in HT-greenfield format frame from the EXPANSION_MAT parameter of the TXVECTOR. Refer to 20.3.9 for details.

b) Construct the PLCP preamble SIGNAL fields from the appropriate fields of the TXVECTOR by adding tail bits, applying convolutional coding, formatting into one or more OFDM symbols, applying cyclic shifts, applying spatial processing, calculating an inverse Fourier transform for each OFDM symbol and transmit chain, and prepending a cyclic prefix or GI to each OFDM symbol in each transmit chain. The number and placement of the PLCP preamble SIGNAL fields depend on the frame format being used. Refer to 20.3.9.3.5, 20.3.9.4.3, and 20.3.9.5.4.

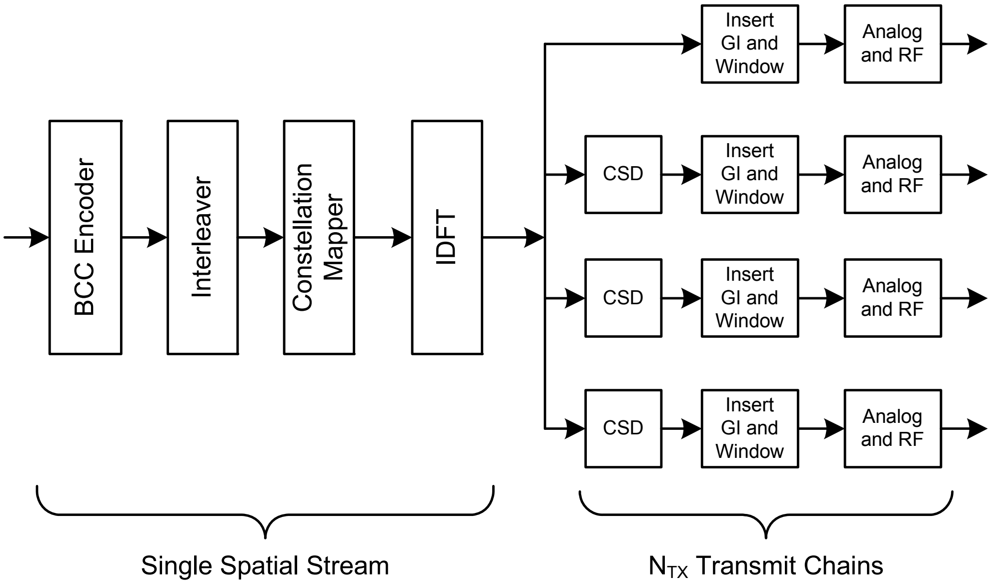
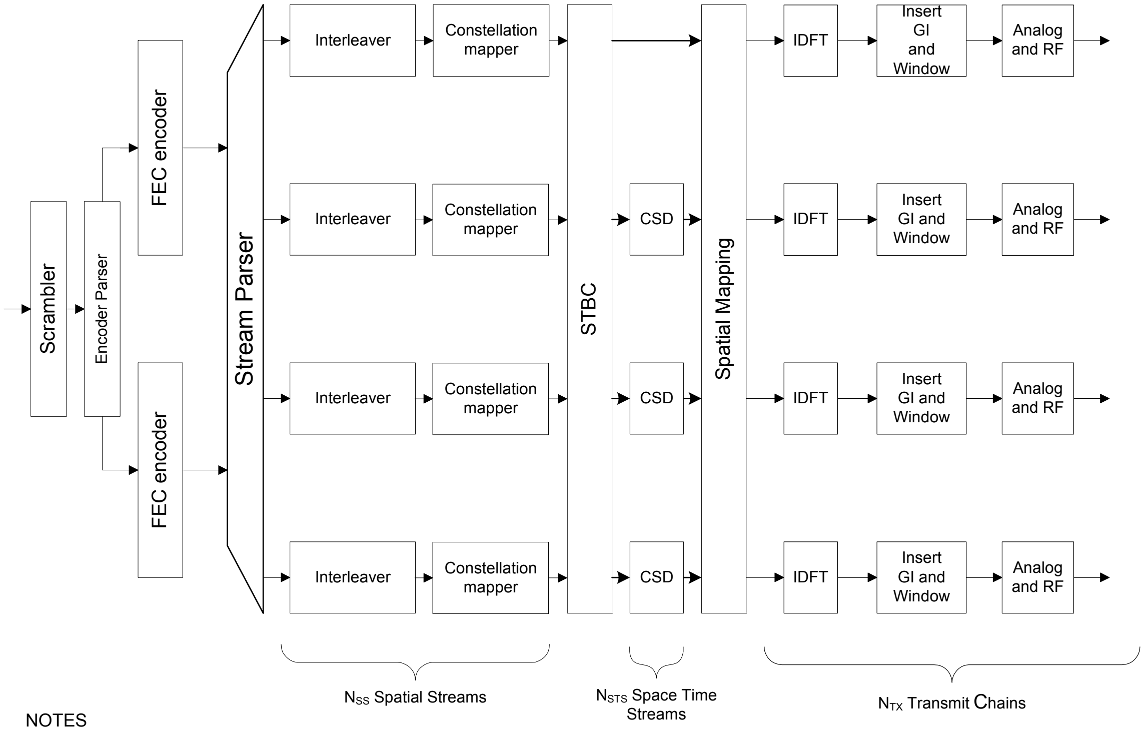


Figure 20-2—Transmitter block diagram 1



NOTES

—There might be 1 or 2 FEC encoders when BCC encoding is used.
—The stream parser might have 1, 2, 3 or 4 outputs.
—When LDPC encoding is used, the interleavers are not used
—When STBC is used, the STBC block has more outputs than inputs.
—When spatial mapping is used, there might be more transmit chains than space time streams.
—The number of inputs to the spatial mapper might be 1, 2, 3, or 4.

Figure 20-3—Transmitter block diagram 2

c) Concatenate the PLCP preamble training and SIGNAL fields for each transmit chain one field after another, in the appropriate order, as described in 20.3.2 and 20.3.7**.**

d) Use the MCS and CH_BANDWIDTH parameters of the TXVECTOR to determine the number of data bits per OFDM symbol (N_{DBPS}), the coding rate (R), the number of coded bits in each OFDM subcarrier (N_{BPSC}), and the number of coded bits per OFDM symbol (N_{CBPS}). Determine the number of encoding streams (N_{ES}) from the MCS, CH_BANDWIDTH, and FEC_CODING parameters of the TXVECTOR. Refer to 20.3.11.4 for details.

e) Append the PSDU to the SERVICE field (see 20.3.11.2). If BCC encoding is to be used, as indicated by the FEC_CODING parameter of the TXVECTOR, tail bits are appended to the PSDU**.** If a single BCC encoder is used (i.e., when the value of N_{ES} is 1), the bit string is extended by 6 zero bits. If two BCC encoders are used (i.e., when the value of N_{ES} is 2), the bit string is extended by 12 zero bits. The number of symbols, N_{SYM}, is calculated according to Equation (20-32), and if necessary, the bit string is further extended with zero bits so that the resulting length is a multiple of $N_{SYM} \times N_{DBPS}$, as described in 20.3.11. If LDPC encoding is to be used, as indicated by the FEC_CODING parameter of the TXVECTOR, the resulting bit string is padded, if needed, by repeating coded bits rather than using zero bits, as given in the encoding procedure of 20.3.11.7.5. The number of resulting symbols is given by Equation (20-41), and the number of repeated coded bits used for padding is given by Equation (20-42). The resulting bit string constitutes the DATA part of the packet.

f) Initiate the scrambler with a pseudorandom nonzero seed, generate a scrambling sequence, and exclusive-OR (XOR) it with the string of data bits, as described in 18.3.5.5.

g) If BCC encoding is to be used, replace the scrambled zero bits that served as tail bits (6 bits if the value of N_{ES} is 1, or 12 bits if the value of N_{ES} is 2) following the data with the same number of nonscrambled zero bits, as described in 18.3.5.3. (These bits return the convolutional encoder to the zero state.)

h) If BCC encoding is to be used and the value of N_{ES} is 2, divide the scrambled data bits between two BCC encoders by sending alternating bits to the two different encoders, as described in 20.3.11.5.

i) If BCC encoding is to be used, encode the extended, scrambled data string with a rate 1/2 convolutional encoder (see 18.3.5.6). Omit (puncture) some of the encoder output string (chosen according to puncturing pattern) to reach the desired coding rate, *R*. Refer to 20.3.11.6 for details. If LDPC encoding is to be used, encode the scrambled data stream according to 20.3.11.7.5.

j) Parse the coded bit stream that results from the BCC encoding or LDPC encoding into N_{SS} spatial streams, where the value of N_{SS} is determined from the MCS parameter of the TXVECTOR. See 20.3.11.8.2 for details.

k) Divide each of the N_{SS} encoded and parsed spatial streams of bits into groups of $N_{CBPSS}(i)$ bits. If BCC encoding is to be used, within each spatial stream and group, perform an interleaving (reordering) of the bits according to a rule corresponding to $N_{BPSCS}(i)$, where i is the index of the spatial stream. Refer to 20.3.6 for details.

l) For each of the N_{SS} encoded, parsed, and interleaved spatial streams, divide the resulting coded and interleaved data string into groups of $N_{BPSCS}(i)$ bits, where i is the index of the spatial stream. For each of the bit groups, convert the bit group into a complex number according to the modulation encoding tables. Refer to 18.3.5.8 for details.

m) Divide the complex number string for each of the resulting N_{SS} spatial streams into groups of N_{SD} complex numbers, where the value of N_{SD} is determined from the CH_OFFSET parameter of

TXVECTOR and the CH_BANDWIDTH parameter of TXVECTOR. Each such group is associated with one OFDM symbol in one spatial stream. In each group, the complex numbers are indexed 0 to $N_{SD} - 1$, and these indices have an associated one-to-one correspondence with subcarrier indices via the mapping function $M^r(k)$ as described in 20.3.11.11, 20.3.11.11.3, 20.3.11.11.4, 20.3.11.11.5, and 20.3.11.12.

n) If STBC is to be applied, as indicated by the STBC parameter in the TXVECTOR, operate on the complex number associated with each data subcarrier in sequential pairs of OFDM symbols as described in 20.3.11.9.2 to generate N_{STS} OFDM symbols for every N_{SS} OFDM symbols associated with the N_{SS} spatial streams. If STBC is not to be used, the number of space-time streams is the same as the number of spatial streams, and the sequences of OFDM symbols in each space-time stream are composed of the sequences of OFDM symbols in the corresponding spatial stream. In each group of N_{SD} resulting complex numbers in each space-time stream, the complex numbers indexed 0 to $N_{SD} - 1$ are mapped onto OFDM subcarriers via the mapping function $M^r(k)$ as described in 20.3.11.11, 20.3.11.11.3, 20.3.11.11.4, 20.3.11.11.5, and 20.3.11.12.

o) Determine whether 20 MHz or 40 MHz operation is to be used from the CH_BANDWIDTH parameter of the TXVECTOR. Specifically, when CH_BANDWIDTH is HT_CBW20 or NON_HT_CBW20, 20 MHz operation is to be used. When CH_BANDWIDTH is HT_CBW40 or NON_HT_CBW40, 40 MHz operation is to be used. For 20 MHz operation (with the exception of non-HT formats), insert four subcarriers as pilots into positions –21, –7, 7, and 21. The total number of the subcarriers, N_{ST}, is 56. For 40 MHz operation (with the exception of MCS 32 and non-HT duplicate format), insert six subcarriers as pilots into positions –53, –25, –11, 11, 25, and 53, resulting in a total of $N_{ST} = 114$ subcarriers. See 20.3.11.11.5 for pilot locations when using MCS 32 and 20.3.11.12 for pilot locations when using non-HT duplicate format. The pilots are modulated using a pseudorandom cover sequence. Refer to 20.3.11.10 for details. For 40 MHz operation, apply a +90 degree phase shift to the complex value in each OFDM subcarrier with an index greater than 0, as described in 20.3.11.11.4, 20.3.11.11.5, and 20.3.11.12.

p) Map each of the complex numbers in each of the N_{ST} subcarriers in each of the OFDM symbols in each of the N_{STS} space-time streams to the N_{TX} transmit chain inputs. For direct-mapped operation, $N_{TX} = N_{STS}$, and there is a one-to-one correspondence between space-time streams and transmit chains. In this case, the OFDM symbols associated with each space-time stream are also associated with the corresponding transmit chain. Otherwise, a spatial mapping matrix associated with each OFDM subcarrier, as indicated by the EXPANSION_MAT parameter of the TXVECTOR, is used to perform a linear transformation on the vector of N_{STS} complex numbers associated with each subcarrier in each OFDM symbol. This spatial mapping matrix maps the vector of N_{STS} complex numbers in each subcarrier into a vector of N_{TX} complex numbers in each subcarrier. The sequence of N_{ST} complex numbers associated with each transmit chain (where each of the N_{ST} complex numbers is taken from the same position in the N_{TX} vector of complex numbers across the N_{ST} subcarriers associated with an OFDM symbol) constitutes an OFDM symbol associated with the corresponding transmit chain. For details, see 20.3.11.11. Spatial mapping matrices may include cyclic shifts, as described in 20.3.11.11.2.

q) If the CH_BANDWIDTH and CH_OFFSET parameters of the TXVECTOR indicate that upper or lower 20 MHz are to be used in 40 MHz, move the complex numbers associated with subcarriers –28 to 28 in each transmit chain to carriers 4 to 60 in the upper channel or –60 to –4 in the lower channel. Note that this shifts the signal in frequency from the center of the 40 MHz channel to +10 MHz or –10 MHz offset from the center of the 40 MHz channel. The complex numbers in the other subcarriers are set to 0.

r) For each group of N_{ST} subcarriers and each of the N_{TX} transmit chains, convert the subcarriers to time domain using IDFT. Prepend to the Fourier-transformed waveform a circular extension of itself, thus forming a GI, and truncate the resulting periodic waveform to a single OFDM symbol length by applying time domain windowing. Determine the length of the GI according to the GI_TYPE parameter of the TXVECTOR. Refer to 20.3.11.11 and 20.3.11.12 for details. When beamforming is not used, it is sometimes possible to implement the cyclic shifts in the time domain.

s) Append the OFDM symbols associated with each transmit chain one after another, starting after the final field of the PLCP preamble. Refer to 20.3.2 and 20.3.7 for details.

t) Up-convert the resulting complex baseband waveform associated with each transmit chain to an RF signal according to the center frequency of the desired channel and transmit. Refer to 20.3.7 for details. The transmit chains are connected to antenna elements according to ANTENNA_SET of the TXVECTOR if ASEL is applied.

20.3.5 Modulation and coding scheme (MCS)

The MCS is a value that determines the modulation, coding, and number of spatial channels. It is a compact representation that is carried in the HT-SIG. Rate-dependent parameters for the full set of MCSs are shown in Table 20-30 through Table 20-44 (in 20.6). These tables give rate-dependent parameters for MCSs with indices 0 to 76. MCSs with indices 0 to 7 and 32 have a single spatial stream; MCSs with indices 8 to 31 have multiple spatial streams using equal modulation (EQM) on all the streams; MCSs with indices 33 to 76 have multiple spatial streams using unequal modulation (UEQM) on the spatial streams. MCS indices 77 to 127 are reserved.

Table 20-30 through Table 20-33 show rate-dependent parameters for EQM MCSs for one, two, three, and four streams for 20 MHz operation. Table 20-34 through Table 20-37 show rate-dependent parameters for EQM MCSs in one, two, three, and four streams for 40 MHz operation. The same EQM MCSs are used for 20 MHz and 40 MHz operation. Table 20-38 shows rate-dependent parameters for the 40 MHz, 6 Mb/s MCS 32 format.

The remaining tables, Table 20-39 to Table 20-44, show rate-dependent parameters for the MCSs with UEQM of the spatial streams for use with $N_{SS} > 1$, including,

— Transmit beamforming

— STBC modes for which two spatial streams (N_{SS}=2) are encoded into three space-time streams (N_{STS}=3) and three spatial streams (N_{SS}=3) are encoded into four space-time streams (N_{STS}=4). These STBC mode cases are specified in Table 20-18.

UEQM MCSs are detailed in the following tables:

— Table 20-39 through Table 20-41 are for 20 MHz operation.

— Table 20-42 through Table 20-44 are for 40 MHz operation.

MCS 0 to 15 are mandatory in 20 MHz with 800 ns GI at an AP. MCS 0 to 7 are mandatory in 20 MHz with 800 ns GI at all STAs. All other MCSs and modes are optional, specifically including transmit and receive support of 400 ns GI, operation in 40 MHz, and support of MCSs with indices 16 to 76.

20.3.6 Timing-related parameters

Table 20-6 defines the timing-related parameters.

Table 20-6—Timing-related constants

Parameter	TXVECTOR CH_BANDWIDTH			
	NON_HT_CBW20	HT_CBW_20	HT_CBW40 or NON_HT_CBW40	
			HT format	MCS 32 and non-HT duplicate
N_{SD}: Number of complex data numbers	48	52	108	48
N_{SP}: Number of pilot values	4	4	6	4
N_{ST}: Total number of subcarriers See NOTE 1	52	56	114	104
N_{SR}: Highest data subcarrier index	26	28	58	58
Δ_F : Subcarrier frequency spacing	312.5kHz (20 MHz/64)	312.5kHz	312.5kHz (40 MHz/128)	
T_{DFT}: IDFT/DFT period	3.2 µs	3.2 µs	3.2 µs	
T_{GI}: Guard interval duration	0.8 µs= $T_{DFT}/4$	0.8 µs	0.8 µs	
T_{GI2}: Double guard interval	1.6 µs	1.6 µs	1.6 µs	
T_{GIS}: Short guard interval duration	N/A	0.4 µs = $T_{DFT}/8$	0.4 µs See NOTE 2	
$T_{L\text{-}STF}$: Non-HT short training sequence duration	8 µs=10× $T_{DFT}/4$	8 µs	8 µs	
$T_{HT\text{-}GF\text{-}STF}$: HT-greenfield short training field duration	N/A	8 µs=10× $T_{DFT}/4$	8 µs See NOTE 2	
$T_{L\text{-}LTF}$: Non-HT long training field duration	8 µs=2× $T_{DFT}+T_{GI2}$	8 µs	8 µs	
T_{SYM}: Symbol interval	4 µs= $T_{DFT}+T_{GI}$	4 µs	4 µs	
T_{SYMS}: Short GI symbol interval	N/A	3.6 µs = $T_{DFT}+T_{GIS}$	3.6 µs See NOTE 2	
$T_{L\text{-}SIG}$: Non-HT SIGNAL field duration	4 µs= T_{SYM}	4 µs	4 µs	
$T_{HT\text{-}SIG}$: HT SIGNAL field duration	N/A	8 µs= $2T_{SYM}$	8 µs See NOTE 2	
$T_{HT\text{-}STF}$: HT short training field duration	N/A	4 µs	4 µs See NOTE 2	

Table 20-6—Timing-related constants *(continued)*

Parameter	TXVECTOR CH_BANDWIDTH			
	NON_HT_CBW20	HT_CBW_20	HT_CBW40 or NON_HT_CBW40	
			HT format	MCS 32 and non-HT duplicate
$T_{HT\text{-}LTF1}$: First HT long training field duration	N/A	4 µs in HT-mixed format, 8 µs in HT-greenfield format	4 µs in HT-mixed format, 8 µs in HT-greenfield format See NOTE 2	
$T_{HT\text{-}LTFs}$: Second, and subsequent, HT long training fields duration	N/A	4 µs	4 µs See NOTE 2	

NOTE 1—N_{ST}=N_{SD}+N_{SP} except in the cases of MCS 32 and non-HT duplicate, where the number of data subcarriers differs from the number of complex data numbers, and the number of pilot subcarriers differs from the number of pilot values. In those cases, data numbers and pilot values are replicated in upper and lower 20 MHz portions of 40 MHz signal to make a total of 104 subcarriers.
NOTE 2—Not applicable in non-HT formats.
NOTE 3—N/A = Not applicable.

Table 20-7 defines parameters used frequently in Clause 20.

Table 20-7—Frequently used parameters

Symbol	Explanation
N_{CBPS}	Number of coded bits per symbol
$N_{CBPSS}(i)$	Number of coded bits per symbol per the *i*-th spatial stream
N_{DBPS}	Number of data bits per symbol
N_{BPSC}	Number of coded bits per single carrier
$N_{BPSCS}(i)$	Number of coded bits per single carrier for spatial stream *i*
N_{RX}	Number of receive chains
N_{STS}	Number of space-time streams
N_{SS}	Number of spatial streams
N_{ESS}	Number of extension spatial streams
N_{TX}	Number of transmit chains
N_{ES}	Number of BCC encoders for the Data field
N_{HTLTF}	Number of HT Long Training fields (see 20.3.9.4.6)
N_{HTDLTF}	Number of Data HT Long Training fields
N_{HTELTF}	Number of Extension HT Long Training fields
R	Coding rate

20.3.7 Mathematical description of signals

For the description of the convention on mathematical description of signals, see 18.3.2.5.

In the case of either a 20 MHz non-HT format (TXVECTOR parameter FORMAT equal to NON_HT, MODULATION parameter equal to one of {DSSS-OFDM, ERP-OFDM, OFDM}) transmission or a 20 MHz HT format (TXVECTOR parameter FORMAT equal to HT_MF or HT_GF, CH_BANDWIDTH equal to HT_CBW_20) transmission, the channel is divided into 64 subcarriers. In the 20 MHz non-HT format, the signal is transmitted on subcarriers –26 to –1 and 1 to 26, with 0 being the center (dc) carrier. In the 20 MHz HT format, the signal is transmitted on subcarriers –28 to –1 and 1 to 28.

In the case of the 40 MHz HT format, a 40 MHz channel is used. The channel is divided into 128 subcarriers. The signal is transmitted on subcarriers –58 to –2 and 2 to 58.

In the case of 40 MHz HT upper format or 40 MHz HT lower format, the upper or lower 20 MHz is divided into 64 subcarriers. The signal is transmitted on subcarriers –60 to –4 in the case of a 40 MHz HT lower format transmission and on subcarriers 4 to 60 in the case of a 40 MHz HT upper format transmission.

In the case of the MCS 32 and non-HT duplicate formats, the same data are transmitted over two adjacent 20 MHz channels. In this case, the 40 MHz channel is divided into 128 subcarriers, and the data are transmitted on subcarriers –58 to –6 and 6 to 58.

The transmitted signal is described in complex baseband signal notation. The actual transmitted signal is related to the complex baseband signal by the relation shown in Equation (20-1).

$$r_{RF}(t) = \mathrm{Re}\{r(t)\exp(j2\pi f_c t)\} \tag{20-1}$$

where

$\mathrm{Re}\{.\}$ represents the real part of a complex variable

f_c is the center frequency of the carrier

The transmitted RF signal is derived by modulating the complex baseband signal, which consists of several fields. The timing boundaries for the various fields are shown in Figure 20-4.

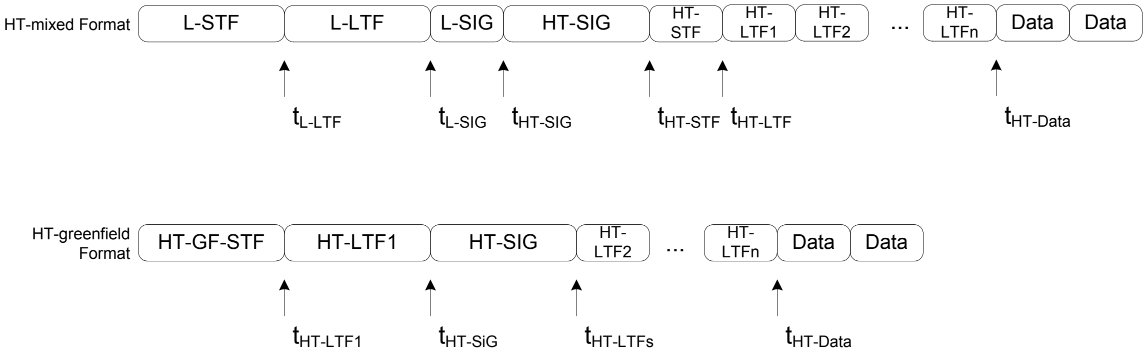


Figure 20-4—Timing boundaries for PPDU fields

The time offset, t_{Field}, determines the starting time of the corresponding field.

In HT-mixed format, the signal transmitted on transmit chain i_{TX} shall be as shown in Equation (20-2).

$$\begin{aligned} r_{PPDU}^{(i_{TX})}(t) &= r_{L\text{-}STF}^{(i_{TX})}(t) + r_{L\text{-}LTF}^{(i_{TX})}(t - t_{L\text{-}LTF}) \\ &+ r_{L\text{-}SIG}^{(i_{TX})}(t - t_{L\text{-}SIG}) + r_{HT\text{-}SIG}^{(i_{TX})}(t - t_{HT\text{-}SIG}) + r_{HT\text{-}STF}^{(i_{TX})}(t - t_{HT\text{-}STF}) \\ &+ \sum_{i_{LTF}=1}^{N_{LTF}} r_{HT\text{-}LTF}^{(i_{TX}, i_{LTF})}(t - t_{HT\text{-}LTF} - (i_{LTF} - 1)T_{HT\text{-}LTFs}) + r_{HT\text{-}DATA}^{(i_{TX})}(t - t_{HT\text{-}DATA}) \end{aligned} \tag{20-2}$$

where

$$\begin{aligned} t_{L\text{-}LTF} &= T_{L\text{-}STF} \\ t_{L\text{-}SIG} &= t_{L\text{-}LTF} + T_{L\text{-}LTF} \\ t_{HT\text{-}SIG} &= t_{L\text{-}SIG} + T_{L\text{-}SIG} \\ t_{HT\text{-}STF} &= t_{HT\text{-}SIG} + T_{HT\text{-}SIG} \\ t_{HT\text{-}LTF} &= t_{HT\text{-}STF} + T_{HT\text{-}STF} \\ t_{HT\text{-}Data} &= t_{HT\text{-}LTF} + N_{LTF} \cdot T_{HT\text{-}LTFs} \end{aligned}$$

In the case of HT-greenfield format, the transmitted signal on transmit chain i_{TX} shall be as shown in Equation (20-3).

$$\begin{aligned} r_{PPDU}^{(i_{TX})}(t) &= r_{HT-GF-STF}^{(i_{TX})}(t) + r_{HT-LTF1}^{(i_{TX})}(t - t_{HT-LTF1}) \\ &+ r_{HT-SIG}^{(i_{TX})}(t - t_{HT-SIG}) \\ &+ \sum_{i_{LTF}=2}^{N_{LTF}} r_{HT-LTF}^{(i_{TX}, i_{LTF})}(t – t_{HT-LTFs} - (i_{LTF} - 2)T_{HT-LTFs}) \\ &+ r_{HT-DATA}^{(i_{TX})}(t - t_{HT-DATA}) \end{aligned} \tag{20-3}$$

where

$$\begin{aligned} t_{HT-LTF1} &= T_{HT-GF-STF} \\ t_{HT\text{-}SIG} &= t_{HT\text{-}LTF1} + T_{HT\text{-}LTF1} \\ t_{HT\text{-}LTFs} &= t_{HT\text{-}SIG} + T_{HT\text{-}SIG} \\ t_{HT\text{-}Data} &= t_{HT\text{-}LTFs} + (N_{LTF} - 1) \cdot T_{HT\text{-}LTFs} \end{aligned}$$

Each baseband waveform, $r_{\text{Field}}^{(i_{TX})}(t)$, is defined via the discrete Fourier transform (DFT) per OFDM symbol as shown in Equation (20-4).

$$r_{\text{Field}}^{(i_{TX})}(t) = \frac{1}{\sqrt{N_{\text{Field}}^{\text{Tone}} \cdot N_{TX}}} w_{\text{T}_{\text{Field}}}(t) \sum_k \Upsilon_k X_k^{(i_{TX})} \exp(j2\pi k \Delta_F t) \tag{20-4}$$

This general representation holds for all fields. A suggested definition of the windowing function, $w_{T_{\text{Field}}}(t)$, is given in 18.3.2.5. The frequency-domain symbols $X_k^{(i_{TX})}$ represent the output of any spatial processing in subcarrier k for transmit chain i_{TX} required for the field.

The function Υ_k is used to represent a rotation of the upper tones in a 40 MHz channel as shown in Equation (20-5) and Equation (20-6).

$$\Upsilon_k = \begin{cases} 1, k \le 0, \text{ in a 40 MHz channel} \\ j, k > 0, \text{ in a 40 MHz channel} \end{cases} \tag{20-5}$$

$$\Upsilon_k = 1, \text{ in a 20 MHz channel} \tag{20-6}$$

The $1/\sqrt{N_{\text{Field}}^{\text{Tone}} \cdot N_{TX}}$ scale factor in Equation (20-4) ensures that the total power of the time domain signal as summed over all transmit chains is either 1 or lower than 1 when required. Table 20-8 summarizes the various values of $N_{\text{Field}}^{\text{Tone}}$.

Table 20-8—Value of tone scaling factor N_{Field}^{Tone}

Field	N_{Field}^{Tone}, see NOTE 1	
	20 MHz	40 MHz
L-STF	12	24
HT-GF-STF	12	24
L-LTF	52	104
L-SIG	52	104
HT-SIG	52/56, see NOTE 2	104/114, see NOTE 2
HT-STF	12	24
HT-LTF	56	114
HT-Data	56	114
MCS 32, see NOTE 3	—	104

NOTE 1—The numbers in the table refer only to the value of N_{Field}^{Tone} as it appears in Equation (20-4) and in subsequent specification of various fields. This value might be different from the actual number of tones being transmitted.

NOTE 2—The values 56 and 114 are for HT-greenfield format; the values 52 and 104 are for HT-mixed format.

NOTE 3—This is the Data field in an MCS 32 format PPDU.

20.3.8 Transmission in the upper/lower 20 MHz of a 40 MHz channel

When transmitting in the upper/lower 20 MHz portion of a 40 MHz channel, the mathematical definition of transmission shall follow that of a 20 MHz channel with f_c in Equation (20-1) replaced by $f_c \pm 10$ MHz.

This rule applies to 20 MHz HT transmission in the upper/lower 20 MHz of a 40 MHz channel (TXVECTOR primitive CH_BANDWIDTH equal to HT_CBW20 and CH_OFFSET primitive equal to CH_OFF_20U or CH_OFF_20L) and to 20 MHz non-HT transmission in the upper/lower 20 MHz of a 40 MHz channel (TXVECTOR primitive CH_BANDWIDTH equal to NON_HT_CBW20 and CH_OFFSET primitive equal to CH_OFF_20U or CH_OFF_20L).

20.3.9 HT preamble

20.3.9.1 Introduction

The HT preambles are defined in HT-mixed format and in HT-greenfield format to carry the required information to operate in a system with multiple transmit and multiple receive antennas.

In the HT-mixed format, to ensure compatibility with non-HT STAs, specific non-HT fields are defined so that they can be received by non-HT STAs compliant with Clause 18 or Clause 19 followed by the fields specific to HT STAs.

In the HT-greenfield format, all of the non-HT fields are omitted. The specific HT fields used are as follows:

- One HT-GF-STF for automatic gain control convergence, timing acquisition, and coarse frequency acquisition,
- One or several HT-LTFs, provided as a way for the receiver to estimate the channel between each spatial mapper input and receive chain. The first HT-LTFs (HT-DLTFs) are necessary for demodulation of the HT-Data portion of the PPDU and are followed, for sounding packets only, by optional HT-LTFs (HT-ELTFs) to sound extra spatial dimensions of the MIMO channel,
- HT-SIG, which provides all the information required to interpret the HT packet format.

In the case of multiple transmit chains, the HT preambles use cyclic shift techniques to prevent unintentional beamforming.

20.3.9.2 HT-mixed format preamble

In HT-mixed format frames, the preamble has fields that support compatibility with Clause 18 and Clause 19 STAs and fields that support HT operation. The non-HT portion of the HT-mixed format preamble enables detection of the PPDU and acquisition of carrier frequency and timing by both HT STAs and STAs that are compliant with Clause 18 or Clause 19. The non-HT portion of the HT-mixed format preamble contains the SIGNAL field (L-SIG) defined in Clause 18 and is thus decodable by STAs compliant with Clause 18 and Clause 19 as well as HT STAs.

The HT portion of the HT-mixed format preamble enables estimation of the MIMO channel to support demodulation of the data portion of the frame by HT STAs. The HT portion of the HT-mixed format preamble also contains the HT-SIG field that supports HT operation.

20.3.9.3 Non-HT portion of the HT-mixed format preamble

20.3.9.3.1 Introduction

The transmission of the non-HT training fields and the L-SIG as part of an HT-mixed format packet is described in 20.3.9.3.2 through 20.3.9.3.5.

20.3.9.3.2 Cyclic shift definition

The cyclic shift values defined in this subclause apply to the non-HT fields in the HT-mixed format preamble and the HT-SIG in the HT-mixed format preamble.

Cyclic shifts are used to prevent unintentional beamforming when the same signal or scalar multiples of one signal are transmitted through different spatial streams or transmit chains. A cyclic shift of duration T_{CS} on a signal $s(t)$ on interval $0 \le t \le T$ is defined as follows, where T is defined as T_{DFT} as referenced in Table 20-6.

With $T_{CS} \le 0$, replace $s(t)$ with $s(t - T_{CS})$ when $0 \le t < T + T_{CS}$ and with $s(t - T_{CS} - T)$ when $T + T_{CS} \le t \le T$. The cyclic-shifted signal is defined as shown in Equation (20-7).

$$s_{CS}(t;T_{CS})\Big|_{T_{CS} < 0} = \begin{cases} s(t - T_{CS}) & 0 \le t < T + T_{CS} \\ s(t - T_{CS} - T) & T + T_{CS} \le t \le T \end{cases} \quad (20\text{-}7)$$

The cyclic shift is applied to each OFDM symbol in the packet separately. Table 20-9 specifies the values for the cyclic shifts that are applied in the L-STF (in an HT-mixed format packet), the L-LTF, and L-SIG. It also applies to the HT-SIG in an HT-mixed format packet.

Table 20-9—Cyclic shift for non-HT portion of packet

$T_{CS}^{i_{TX}}$ values for non-HT portion of packet				
Number of transmit chains	**Cyclic shift for transmit chain 1 (ns)**	**Cyclic shift for transmit chain 2 (ns)**	**Cyclic shift for transmit chain 3 (ns)**	**Cyclic shift for transmit chain 4 (ns)**
1	0	—	—	—
2	0	–200	—	—
3	0	–100	–200	—
4	0	–50	–100	–150

With more than four transmit chains, each cyclic shift on the additional transmit chains shall be between –200 ns and 0 ns inclusive.

20.3.9.3.3 L-STF definition

The L-STF is identical to the Clause 18 short training symbol. The non-HT short training OFDM symbol in the 20 MHz channel width is shown in Equation (20-8).

$$\begin{aligned} S_{-26,26} = {} & \sqrt{1/2} \\ & \{0, 0, 1+j, 0, 0, 0, -1-j, 0, 0, 0, 1+j, 0, 0, 0, -1-j, 0, 0, 0, -1-j, 0, 0, 0, 1+j, 0, 0, 0, \\ & 0, 0, 0, 0, -1-j, 0, 0, 0, -1-j, 0, 0, 0, 1+j, 0, 0, 0, 1+j, 0, 0, 0, 1+j, 0, 0, 0, 1+j, 0, 0\} \end{aligned} \quad (20\text{-}8)$$

The normalization factor $\sqrt{1/2}$ is the QPSK normalization.

The non-HT short training OFDM symbol in a 40 MHz channel width is given by Equation (20-9), after rotating the tones in the upper subchannel (subcarriers 6–58) by +90º (see Equation (20-10)).

$$\begin{aligned} S_{-58,58} = {} & \sqrt{1/2} \\ & \{0, 0, 1+j, 0, 0, 0, -1-j, 0, 0, 0, 1+j, 0, 0, 0, -1-j, 0, 0, 0, -1-j, 0, 0, 0, 1+j, 0, 0, 0, \\ & 0, 0, 0, 0, -1-j, 0, 0, 0, -1-j, 0, 0, 0, 1+j, 0, 0, 0, 1+j, 0, 0, 0, 1+j, 0, 0, 0, 1+j, 0, 0, 0, 0, 0, \\ & 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1+j, 0, 0, 0, -1-j, 0, 0, 0, 1+j, 0, 0, 0, -1-j, 0, 0, 0, -1-j, 0, 0, 0, 1+j, \\ & 0, 0, 0, 0, 0, 0, 0, -1-j, 0, 0, 0, -1-j, 0, 0, 0, 1+j, 0, 0, 0, 1+j, 0, 0, 0, 1+j, 0, 0, 0, 1+j, 0, 0\} \end{aligned} \tag{20-9}$$

In HT-mixed format, the L-STF on transmit chain i_{TX} shall be as shown in Equation (20-10).

$$r_{L-STF}^{(i_{TX})}(t) = \frac{1}{\sqrt{N_{TX} \cdot N_{L-STF}^{Tone}}} w_{T_{L-STF}}(t) \sum_{k=-N_{SR}}^{N_{SR}} \Upsilon_k S_k \exp(j2\pi k \Delta_F (t - T_{CS}^{i_{TX}})) \tag{20-10}$$

where

$T_{CS}^{i_{TX}}$ represents the cyclic shift for transmit chain i_{TX} and takes values from Table 20-9

Υ_k is defined in Equation (20-5) and Equation (20-6)

The L-STF has a period of 0.8 μs. The entire STF includes ten such periods, with a total duration of $T_{L\text{-}STF}$ = 8 μs.

20.3.9.3.4 L-LTF definition

The non-HT long training OFDM symbol is identical to the Clause 18 long training OFDM symbol. In the 20 MHz channel width, the long training OFDM symbol is given by Equation (20-11).

$$\begin{aligned} L_{-26,26} = {} & \{1, 1, -1, -1, 1, 1, -1, 1, -1, 1, 1, 1, 1, 1, 1, -1, -1, 1, 1, -1, 1, -1, 1, 1, 1, 1, 0, \\ & 1, -1, -1, 1, 1, -1, 1, -1, 1, -1, -1, -1, -1, -1, 1, 1, -1, -1, 1, -1, 1, -1, 1, 1, 1, 1\} \end{aligned} \tag{20-11}$$

The non-HT long training OFDM symbol in a 40 MHz channel width is given by Equation (20-12), after rotating the tones in the upper subchannel (subcarriers 6–58) by +90º (see Equation (20-13)).

$$\begin{aligned} L_{-58,58} = {} & \{1, 1, -1, -1, 1, 1, -1, 1, -1, 1, 1, 1, 1, 1, 1, -1, -1, 1, 1, -1, 1, -1, 1, 1, 1, 1, 0, \\ & 1, -1, -1, 1, 1, -1, 1, -1, 1, -1, -1, -1, -1, -1, 1, 1, -1, -1, 1, -1, 1, -1, 1, 1, 1, 1, 0, 0, 0, 0, 0, \\ & 0, 0, 0, 0, 0, 0, 1, 1, -1, -1, 1, 1, -1, 1, -1, 1, 1, 1, 1, 1, 1, -1, -1, 1, 1, -1, 1, -1, 1, 1, 1, 1, 0, \\ & 1, -1, -1, 1, 1, -1, 1, -1, 1, -1, -1, -1, -1, -1, 1, 1, -1, -1, 1, -1, 1, -1, 1, 1, 1, 1\} \end{aligned} \tag{20-12}$$

The subcarriers at ± 32 in 40 MHz, which are the dc subcarriers for the non-HT 20 MHz transmission, are both nulled in the L-LTF. Such an arrangement allows proper synchronization of a 20 MHz non-HT STA.

The L-LTF waveform shall be as shown in Equation (20-13).

$$r_{L-LTF}^{(i_{TX})}(t) = \frac{1}{\sqrt{N_{TX} \cdot N_{L-LTF}^{Tone}}} w_{T_{L-LTF}}(t) \sum_{k=-N_{SR}}^{N_{SR}} \Upsilon_k L_k \exp(j2\pi k \Delta_F (t - T_{GI2} - T_{CS}^{i_{TX}})) \tag{20-13}$$

where

T_{GI2} is 1.6 μs

$T_{CS}^{i_{TX}}$ represents the cyclic shift for transmit chain i_{TX} and takes values specified in Table 20-9

Υ_k is defined in Equation (20-5) and Equation (20-6)

The entire LTF includes two 3.2 µs IDFT/DFT periods and an additional 1.6 µs double GI. The entire LTF is modulated with the L-LTF waveform.

20.3.9.3.5 L-SIG definition

The L-SIG is used to communicate rate and length information.The structure of the L-SIG is shown in Figure 20-5.

Rate (4 bits)				R	Length (12 bits)												P	Tail (6 bits)					
R1	R2	R3	R4															"0"	"0"	"0"	"0"	"0"	"0"
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23

Figure 20-5—L-SIG structure

The value in the Rate field is obtained from the L_DATARATE field of the TXVECTOR. The value in the Length field is obtained from the L_LENGTH field of the TXVECTOR. The Length field is transmitted LSB first.

The reserved bit shall be set to 0.

The parity field has the even parity of bits 0–16.

The L-SIG shall be encoded, interleaved and mapped, and it shall have pilots inserted following the steps described in 18.3.5.6, 18.3.5.7, and 18.3.5.9. The stream of 48 complex numbers generated by the steps described in 18.3.5.7 is denoted by $d_k, k = 0\ldots47$. The time domain waveform of the L-SIG in 20 MHz transmission shall be as given by Equation (20-14).

$$r_{L-SIG}^{(i_{TX})}(t) = \frac{1}{\sqrt{N_{TX}\cdot N_{L-SIG}^{Tone}}} w_{\mathrm{T}_{SYM}}(t) \sum_{k=-26}^{26} (D_k + p_0 P_k)\exp(j2\pi k\Delta_F(t - T_{GI} - T_{CS}^{i_{TX}})) \tag{20-14}$$

In a 40 MHz transmission the time domain waveform of the L-SIG shall be as given by Equation (20-15).

$$\begin{aligned} r_{L-SIG}^{(i_{TX})}(t) = {} & \frac{1}{\sqrt{N_{TX}\cdot N_{L-SIG}^{Tone}}} w_{\mathrm{T}_{SYM}}(t) \sum_{k=-26}^{26} (D_k + p_0 P_k) \\ & \cdot (\exp(j2\pi(k-32)\Delta_F(t - T_{GI} - T_{CS}^{i_{TX}})) + j\exp(j2\pi(k+32)\Delta_F(t - T_{GI} - T_{CS}^{i_{TX}}))) \end{aligned} \tag{20-15}$$

where

$$D_k = \begin{cases} 0, k = 0, \pm 7, \pm 21 \\ d_{M^r(k)}, \text{otherwise} \end{cases}$$

$$M^r(k) = \begin{cases} k + 26, -26 \le k \le -22 \\ k + 25, -20 \le k \le -8 \\ k + 24, -6 \le k \le -1 \\ k + 23, 1 \le k \le 6 \\ k + 22, 8 \le k \le 20 \\ k + 21, 22 \le k \le 26 \end{cases}$$

P_k is defined in 18.3.5.10

p_0 is the first pilot value in the sequence defined in 18.3.5.10

N_{L-SIG}^{Tone} has the value given in Table 20-8

$T_{CS}^{i_{TX}}$ represents the cyclic shift for transmit chain i_{TX} and is defined by Table 20-9 for HT-mixed format PPDUs

NOTE— D_k exists for $-N_{SR} \le k \le N_{SR}$ and takes the values from d_k that exists for $0 \le k \le N_{SD} - 1$. $M^r(k)$ is a “reverse” function of the function $M(k)$ defined in 18.3.5.10.

20.3.9.4 HT portion of HT-mixed format preamble

20.3.9.4.1 Introduction

When an HT-mixed format preamble is transmitted, the HT preamble consists of the HT-STF, the HT-LTFs, and the HT-SIG.

20.3.9.4.2 Cyclic shift definition

The cyclic shift values defined in this subclause apply to the HT-STF and HT-LTFs of the HT-mixed format preamble. The cyclic shift values defined in 20.3.9.3.2 apply to the HT-SIG in an HT-mixed format preamble.

Throughout the HT portion of an HT-mixed format preamble, cyclic shift is applied to prevent beamforming when similar signals are transmitted in different space-time streams. The same cyclic shift is applied to these streams during the transmission of the data portion of the frame. The values of the cyclic shifts to be used during the HT portion of the HT-mixed format preamble (with the exception of the HT_SIG) and the data portion of the frame are specified in Table 20-10.

Table 20-10—Cyclic shift values of HT portion of packet

$T_{CS}^{i_{STS}}$ values for HT portion of packet				
Number of space-time streams	Cyclic shift for space-time stream 1 (ns)	Cyclic shift for space-time stream 2 (ns)	Cyclic shift for space-time stream 3 (ns)	Cyclic shift for space-time stream 4 (ns)
1	0	—	—	—
2	0	–400	—	—
3	0	–400	–200	—
4	0	–400	–200	–600

20.3.9.4.3 HT-SIG definition

The HT-SIG is used to carry information required to interpret the HT packet formats. The fields of the HT-SIG are described in Table 20-11.

Table 20-11—HT-SIG fields

Field	Number of bits	Explanation and coding
Modulation and Coding Scheme	7	Index into the MCS table. See NOTE 1.
CBW 20/40	1	Set to 0 for 20 MHz or 40 MHz upper/lower. Set to 1 for 40 MHz.
HT Length	16	The number of octets of data in the PSDU in the range of 0 to 65 535. See NOTE 1 and NOTE 2.
Smoothing	1	Set to 1 indicates that channel estimate smoothing is recommended. Set to 0 indicates that only per-carrier independent (unsmoothed) channel estimate is recommended. See 20.3.11.11.2.
Not Sounding	1	Set to 0 indicates that PPDU is a sounding PPDU. Set to 1 indicates that the PPDU is not a sounding PPDU.
Reserved	1	Set to 1.
Aggregation	1	Set to 1 to indicate that the PPDU in the data portion of the packet contains an A-MPDU; otherwise, set to 0.
STBC	2	Set to a nonzero number, to indicate the difference between the number of space-time streams (N_{STS}) and the number of spatial streams (N_{SS}) indicated by the MCS. Set to 00 to indicate no STBC ($N_{STS} = N_{SS}$). See NOTE 1.
FEC coding	1	Set to 1 for LDPC. Set to 0 for BCC.
Short GI	1	Set to 1 to indicate that the short GI is used after the HT training. Set to 0 otherwise.
Number of extension spatial streams	2	Indicates the number of extension spatial streams (N_{ESS}). Set to 0 for no extension spatial stream. Set to 1 for 1 extension spatial stream. Set to 2 for 2 extension spatial streams. Set to 3 for 3 extension spatial streams. See NOTE 1.
CRC	8	CRC of bits 0–23 in HT-SIG$_1$ and bits 0–9 in HT-SIG$_2$. See 20.3.9.4.4. The first bit to be transmitted is bit C7 as explained in 20.3.9.4.4.
Tail Bits	6	Used to terminate the trellis of the convolution coder. Set to 0.

NOTE 1—Integer fields are transmitted in unsigned binary format, LSB first.
NOTE 2—A value of 0 in the HT Length field indicates a PPDU that does not include a data field, i.e., NDP. NDP transmissions are used for sounding purposes only (see 9.31.2). The packet ends after the last HT-LTF or the HT-SIG.

The structure of the HT-SIG$_1$ and HT-SIG$_2$ fields is defined in Figure 20-6.

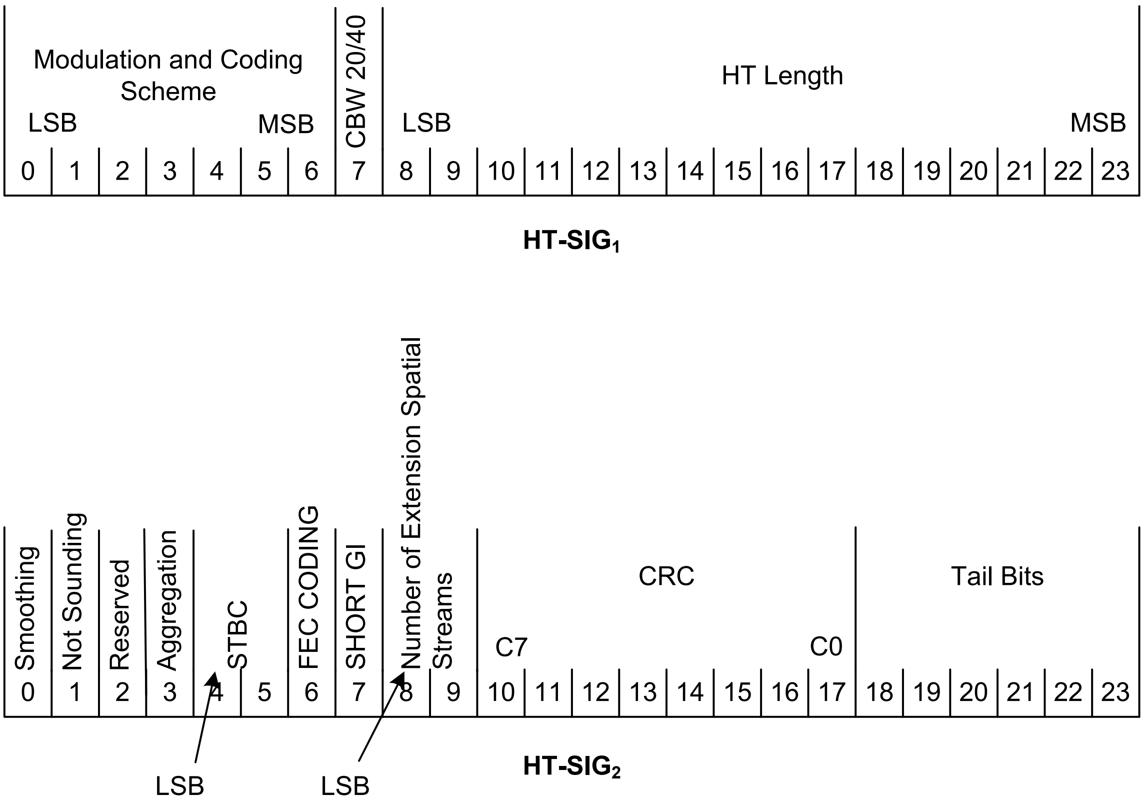


Figure 20-6—Format of HT-SIG$_1$ and HT-SIG$_2$

The HT-SIG is composed of two parts, HT-SIG$_1$ and HT-SIG$_2$, each containing 24 bits, as shown in Figure 20-6. All the fields in the HT-SIG are transmitted LSB first, and HT-SIG$_1$ is transmitted before HT-SIG$_2$.

The HT-SIG parts shall be encoded at R = 1/2, interleaved, and mapped to a BPSK constellation, and they have pilots inserted following the steps described in 18.3.5.6, 18.3.5.7, 18.3.5.8, and 18.3.5.9, respectively. The BPSK constellation is rotated by 90° relative to the L-SIG in order to accommodate detection of the start of the HT-SIG. The stream of 96 complex numbers generated by these steps is divided into two groups of 48 complex numbers: $d_{k,n}$, $0 \le k \le 47$, $n = 0, 1$. The time domain waveform for the HT-SIG in an HT-mixed format packet in a 20 MHz transmission shall be as shown in Equation (20-16).

$$r_{HT-SIG}^{i_{TX}}(t) = \frac{1}{\sqrt{N_{TX} \cdot N_{HT-SIG}^{Tone}}} \sum_{n=0}^{1} w_{\mathrm{T}_{SYM}}(t - nT_{SYM}) \cdot \sum_{k=-26}^{26} (jD_{k,n} + p_{n+1}P_k)\exp(j2\pi k\Delta_F(t - nT_{SYM} - T_{GI} - T_{CS}^{i_{TX}})) \qquad (20\text{-}16)$$

For a 40 MHz transmission, the time domain waveform shall be as shown in Equation (20-17).

$$
\begin{aligned}
r_{HT-SIG}^{i_{TX}}(t) &= \frac{1}{\sqrt{N_{TX} \cdot N_{HT-SIG}^{Tone}}} \sum_{n=0}^{1} w_{T_{SYM}}(t - nT_{SYM}) \\
&\cdot \sum_{k=-26}^{26} (jD_{k,n} + p_{n+1}P_k)(\exp(j2\pi(k-32)\Delta_F(t - nT_{SYM} - T_{GI} - T_{CS}^{i_{TX}})) \\
&+ j\exp(j2\pi(k+32)\Delta_F(t - nT_{SYM} - T_{GI} - T_{CS}^{i_{TX}})))
\end{aligned}
\qquad (20\text{-}17)
$$

where

$$
D_{k,n} = \begin{cases} 0, k = 0, \pm 7, \pm 21 \\ d_{M^r(k),n}, \text{ otherwise} \end{cases}
$$

$M^r(k)$ is defined in 20.3.9.3

P_k and p_n are defined in 18.3.5.10

N_{HT-SIG}^{Tone} has the value given in Table 20-8

$T_{CS}^{i_{TX}}$ represents the cyclic shift for transmit chain i_{TX} and is defined by Table 20-9 for HT-mixed format PPDUs.

NOTE—This definition results in a quadrature binary phase shift keying (QBPSK) modulation in which the constellation of the data tones is rotated by 90º relative to the L-SIG in HT-mixed format PPDUs and relative to the first HT-LTF in HT-greenfield format PPDUs (see Figure 20-7). In HT-mixed format PPDUs, the HT-SIG is transmitted with the same number of subcarriers and the same cyclic shifts as the preceding non-HT portion of the preamble. This is done to accommodate the estimation of channel parameters needed to robustly demodulate and decode the information contained in the HT-SIG.

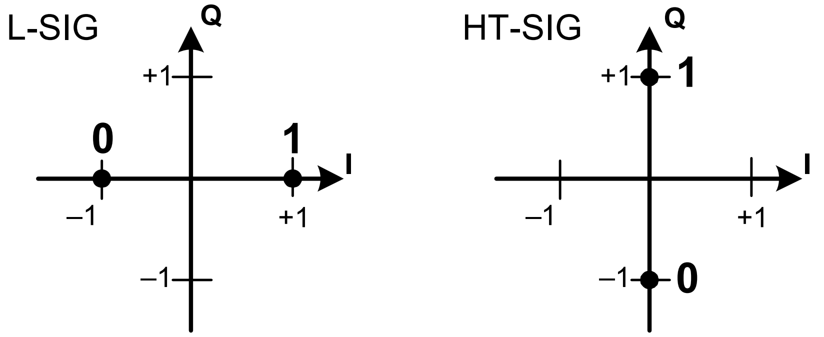


Figure 20-7—Data tone constellations in an HT-mixed format PPDU

20.3.9.4.4 CRC calculation for HT-SIG

The CRC protects bits 0–33 of the HT-SIG (bits 0–23 of HT-SIG$_1$ and bits 0–9 of HT-SIG$_2$). The value of the CRC field shall be the ones complement of

$$crc(D) = (M(D) \oplus I(D))D^8 \ modulo \ G(D) \tag{20-18}$$

where

$M(D) = m_0D^{33} + m_1D^{32} + \ldots + m_{32}D + m_{33}$ is the HT-SIG represented as a polynomial

where

m_0 is bit 0 of HT-SIG$_1$

m_{33} is bit 9 of HT-SIG$_2$

$I(D) = \sum_{i=26}^{33} D^i$ are initialization values that are added modulo 2 to the first 8 bits of HT-SIG$_1$

$G(D) = D^8 + D^2 + D + 1$ is the CRC generating polynomial

$crc(D) = c_0D^7 + c_1D^6 + \ldots + c_6D + c_7$

The CRC field is transmitted with c_7 first.

Figure 20-8 shows the operation of the CRC. First, the shift register is reset to all ones. The bits are then passed through the XOR operation at the input. When the last bit has entered, the output is generated by shifting the bits out of the shift register, C7 first, through an inverter.

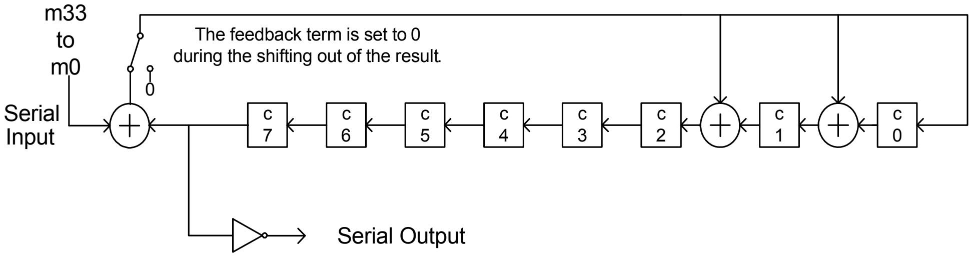

m33
to
m0
Serial
Input
The feedback term is set to 0
during the shifting out of the result
0
c 7
c 6
c 5
c 4
c 3
c 2
c 1
c 0
Serial Output

Figure 20-8—HT-SIG CRC calculation

As an example, if bits $\{m_0 \ldots m_{33}\}$ are given by {1 1 1 1 0 0 0 1 0 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0}, the output bits $\{b_7 \ldots b_0\}$, where b_7 is output first, are {1 0 1 0 1 0 0 0}.

20.3.9.4.5 HT-STF definition

The purpose of the HT-STF is to improve automatic gain control estimation in a MIMO system. The duration of the HT-STF is 4 µs. In a 20 MHz transmission, the frequency sequence used to construct the HT-STF is identical to L-STF. In a 40 MHz transmission, the HT-STF is constructed from the 20 MHz version by duplicating and frequency shifting and by rotating the upper subcarriers by 90°. The frequency sequences are shown in Equation (20-19) and Equation (20-20).

For 20 MHz:

$$\begin{aligned} HTS_{-28,28} = {} & \sqrt{1/2} \\ & \{0, 0, 0, 0, 1+j, 0, 0, 0, -1-j, 0, 0, 0, 1+j, 0, 0, 0, -1-j, 0, 0, 0, -1-j, 0, 0, 0, 1+j, 0, 0, 0, \\ & 0, 0, 0, 0, -1-j, 0, 0, 0, -1-j, 0, 0, 0, 1+j, 0, 0, 0, 1+j, 0, 0, 0, 1+j, 0, 0, 0, 1+j, 0, 0, 0, 0\} \end{aligned} \tag{20-19}$$

For 40 MHz:

$$HTS_{-58,58} = \sqrt{1/2} \tag{20-20}$$
$$\{0, 0, 1+j, 0, 0, 0, -1-j, 0, 0, 0, 1+j, 0, 0, 0, -1-j, 0, 0, 0, -1-j, 0, 0, 0, 1+j, 0, 0, 0,$$
$$0, 0, 0, 0, -1-j, 0, 0, 0, -1-j, 0, 0, 0, 1+j, 0, 0, 0, 1+j, 0, 0, 0, 1+j, 0, 0, 0, 1+j, 0, 0, 0, 0, 0,$$
$$0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1+j, 0, 0, 0, -1-j, 0, 0, 0, 1+j, 0, 0, 0, -1-j, 0, 0, 0, -1-j, 0, 0, 0, 1+j,$$
$$0, 0, 0, 0, 0, 0, 0, -1-j, 0, 0, 0, -1-j, 0, 0, 0, 1+j, 0, 0, 0, 1+j, 0, 0, 0, 1+j, 0, 0, 0, 1+j, 0, 0\}$$

The time domain representation of the transmission in transmit chain i_{TX} shall be as shown in Equation (20-21).

$$r_{HT-STF}^{i_{TX}}(t) = \frac{1}{\sqrt{N_{STS} \cdot N_{HT-STF}^{Tone}}} w_{\mathrm{T}_{HT-STF}}(t)$$
$$\sum_{k=-N_{SR}}^{N_{SR}} \sum_{i_{STS}=1}^{N_{STS}} [Q_k]_{i_{TX}, i_{STS}} \Upsilon_k HTS_k \exp(j2\pi k\Delta_F(t - T_{CS}^{i_{STS}})) \tag{20-21}$$

where

$T_{CS}^{i_{STS}}$ represents the cyclic shift for the space-time stream i_{STS} and takes the values given in Table 20-10

Q_k is defined in 20.3.11.11.2

Υ_k is defined in Equation (20-5) and Equation (20-6)

20.3.9.4.6 HT-LTF definition

The HT-LTF provides a means for the receiver to estimate the MIMO channel between the set of QAM mapper outputs (or, if STBC is applied, the STBC encoder outputs) and the receive chains. If the transmitter is providing training for exactly the space-time streams (spatial mapper inputs) used for the transmission of the PSDU, the number of training symbols, N_{LTF}, is equal to the number of space-time streams, N_{STS}, except that for three space-time streams, four training symbols are required. If the transmitter is providing training for more space-time streams (spatial mapper inputs) than the number used for the transmission of the PSDU, the number of training symbols is greater than the number of space-time streams. This latter case happens in a sounding PPDU.

The HT-LTF portion has one or two parts. The first part consists of one, two, or four HT-LTFs that are necessary for demodulation of the HT-Data portion of the PPDU. These HT-LTFs are referred to as HT-DLTFs. The optional second part consists of zero, one, two, or four HT-LTFs that may be used to sound extra spatial dimensions of the MIMO channel that are not utilized by the HT-Data portion of the PPDU. These HT-LTFs are referred to as HT-ELTFs. If a receiver has not advertised its ability to receive HT-ELTFs, it shall either issue a PHY-RXEND.indicate(UnsupportedRate) primitive upon reception of a frame that includes HT-ELTFs or decode that frame. (When an HT packet includes one or more HT-ELTFs, it is optional for a receiver that has not advertised its capability to receive HT-ELTFs to decode the data portion of the PPDU.)

The number of HT-DLTFs is denoted N_{HTDLTF}. The number of HT-ELTFs is denoted N_{HTELTF}. The total number of HT-LTFs is shown in Equation (20-22).

$$N_{HTLTF} = N_{HTDLTF} + N_{HTELTF} \tag{20-22}$$

N_{HTLTF} shall not exceed 5. Table 20-12 shows the determination of the number of space-time streams from the MCS and STBC fields in the HT-SIG. Table 20-13 shows the number of HT-DLTFs as a function of the number of space-time streams (N_{STS}). Table 20-14 shows the number of HT-ELTFs as a function of the number of extension spatial streams (N_{ESS}). N_{STS} plus N_{ESS} is less than or equal to 4. In the case where N_{STS} equals 3, N_{ESS} cannot exceed one; if N_{ESS} equals one in this case then N_{LTF} equals 5.

Table 20-12—Determining the number of space-time streams

Number of Spatial Streams (from MCS) N_{SS}	**STBC field**	**Number of space-time streams** N_{STS}
1	0	1
1	1	2
2	0	2
2	1	3
2	2	4
3	0	3
3	1	4
4	0	4

Table 20-13—Number of HT-DLTFs required for data space-time streams

N_{STS}	N_{HTDLTF}
1	1
2	2
3	4
4	4

Table 20-14—Number of HT-ELTFs required for extension spatial streams

N_{ESS}	N_{HTELTF}
0	0
1	1
2	2
3	4

The HT-LTF sequence shown in Equation (20-23) is transmitted in the case of 20 MHz operation.

$$\begin{aligned} HTLTF_{-28,28} = \{&1, 1, 1, 1, -1, -1, 1, 1, -1, 1, -1, 1, 1, 1, 1, 1, 1, -1, -1, 1, 1, -1, 1, -1, 1, 1, 1, 1, 0,\\ &1, -1, -1, 1, 1, -1, 1, -1, 1, -1, -1, -1, -1, -1, 1, 1, -1, -1, 1, -1, 1, -1, 1, 1, 1, 1, -1, -1\} \end{aligned} \tag{20-23}$$

NOTE—This sequence is an extension of the L-LTF where the four extra subcarriers are filled with +1 for negative frequencies and –1 for positive frequencies.

In 40 MHz transmissions, including MCS 32 format frames, the sequence to be transmitted is shown in Equation (20-24).

$$\begin{aligned} HTLTF_{-58,58} = \{&1, 1, -1, -1, 1, 1, -1, 1, -1, 1, 1, 1, 1, 1, 1, -1, -1, 1, 1, -1, 1, -1, 1, 1, 1, 1, 1,\\ &1, -1, -1, 1, 1, -1, 1, -1, 1, -1, -1, -1, -1, -1, 1, 1, -1, -1, 1, -1, 1, -1, 1, 1, 1, 1, -1, -1, -1, 1, 0,\\ &0, 0, -1, 1, 1, -1, 1, 1, -1, -1, 1, 1, -1, 1, -1, 1, 1, 1, 1, 1, 1, -1, -1, 1, 1, -1, 1, -1, 1, 1, 1, 1, 1,\\ &1, -1, -1, 1, 1, -1, 1, -1, 1, -1, -1, -1, -1, -1, 1, 1, -1, -1, 1, -1, 1, -1, 1, 1, 1, 1\} \end{aligned} \tag{20-24}$$

NOTE—This sequence is also constructed by extending the L-LTF in the following way: first, the L-LTF is duplicated and shifted as explained in 20.3.9.3.4 for the non-HT duplicate format; then the missing subcarriers [–32, –5, –4, –3, –2, 2, 3, 4, 5, 32] are filled with the values [1, –1, –1, –1, 1, –1, 1, 1, –1, 1], respectively.

This sequence, occupying 114 tones, is used even if the data portion is transmitted with MCS 32 format, which uses 104 tones.

NOTE—This sequence uses 114 tones when MCS 32 format is used to retain consistency with other 40 MHz formats and to facilitate channel estimation for beamforming and link adaptation.

In an HT-mixed format preamble, each HT-LTF consists of a single occurrence of the sequence plus a GI insertion and has a duration of 4 μs. In case of multiple space-time streams, cyclic shift is invoked as specified in Table 20-10.

The generation of HT-DLTFs is shown in Figure 20-9. The generation of HT-ELTFs is shown in Figure 20-10. In these figures, and in the following text, the following notational conventions are used:

— $[X]_{m,n}$ indicates the element in row m and column n of matrix X

— $[X]_N$ indicates a matrix consisting of the first N columns of matrix X

— $[X]_{M:N}$ indicates a matrix consisting of columns M to N of matrix X

where

$M \le N$

X is either Q_k or P_{HTLTF}

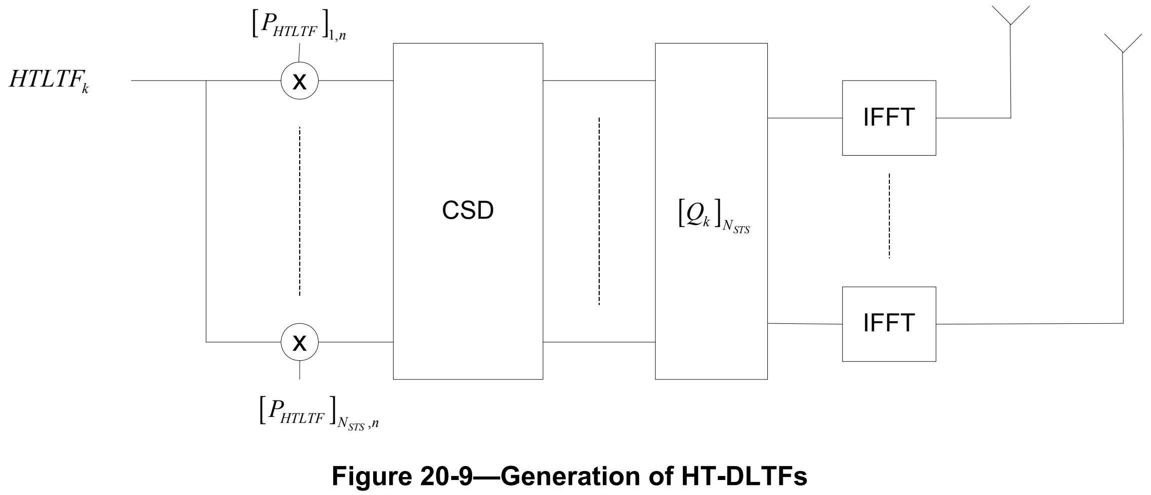


Figure 20-9—Generation of HT-DLTFs

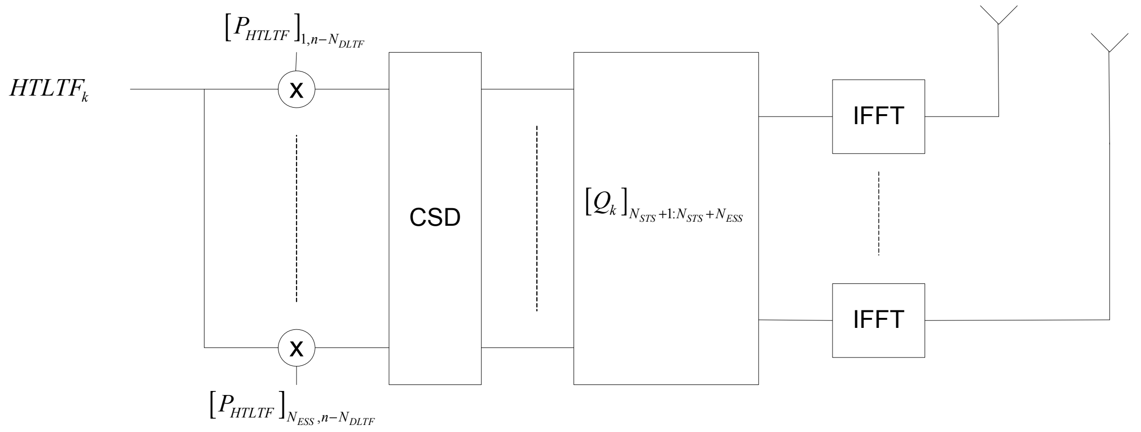

$[P_{HTLTF}]_{1,n-N_{DLTF}}$
$HTLTF_k$
X
CSD
$[Q_k]_{N_{STS}+1:N_{STS}+N_{ESS}}$
IFFT
IFFT
X
$[P_{HTLTF}]_{N_{ESS},n-N_{DLTF}}$

Figure 20-10—Generation of HT-ELTFs

The mapping between space-time streams and transmit chains is defined by the columns of an antenna map matrix Q_k for subcarrier k. The first N_{STS} columns define the space-time streams used for data transmission, and the next N_{ESS} columns (up to $N_{TX}-N_{STS}$ columns) define the extension spatial streams. Thus, for the purpose of defining HT-LTFs, Q_k is an $N_{TX}\times(N_{STS}+N_{ESS})$ dimension matrix. Columns $1\ldots N_{STS}$ of Q_k are excited by the HT-DLTFs, and columns $N_{STS}+1\ldots N_{STS}+N_{ESS}$ are excited by the HT-ELTFs, where $N_{STS}+N_{ESS}\leq N_{TX}$ is the total number of spatial streams being probed by the HT-LTFs.

Possible forms of Q_k and other limitations on Q_k are specified in 20.3.11.11.2. P_{HTLTF} is defined in Equation (20-27).

The time domain representation of the waveform transmitted on transmit chain i_{TX} during HT-DLTF n, where $1\leq n\leq N_{HTDLTF}$, shall be as shown in Equation (20-25).

$$r^{n,i_{TX}}_{HT-LTF}(t) = \frac{1}{\sqrt{N_{STS}\cdot N^{Tone}_{HT-LTF}}}w_{\mathrm{T}_{HT-LTFs}}(t) \cdot \sum_{k=-N_{SR}}^{N_{SR}}\sum_{i_{STS}=1}^{N_{STS}}[Q_k]_{i_{TX},i_{STS}}[P_{HTLTF}]_{i_{STS},n}\Upsilon_k HTLTF_k\exp(j2\pi k\Delta_F(t-T_{GI}-T_{CS}^{i_{STS}})) \qquad (20\text{-}25)$$

For the HT-ELTFs $(N_{HTDLTF} < n \le N_{HTLTF})$, it shall be as shown in Equation (20-26).

$$r_{HT-LTF}^{n,i_{TX}}(t) = \frac{1}{\sqrt{N_{HT-LTF}^{Tone} \cdot N_{ESS}}} w_{T_{HT-LTFs}}(t) \cdot \sum_{k=-N_{SR}}^{N_{SR}} \sum_{i_{ESS}=1}^{N_{ESS}} \Big([Q_k]_{i_{TX}, N_{STS}+i_{ESS}} [P_{HTLTF}]_{i_{ESS}, n-N_{HTDLTF}} \Upsilon_k HTLTF_k \cdot \exp\left(j2\pi k\Delta_F(t - T_{GI} - T_{CS}^{i_{ESS}})\right)\Big) \quad (20\text{-}26)$$

where

$T_{CS}^{i_{STS}}$ cyclic shift values are given in Table 20-10

$T_{CS}^{i_{ESS}}$ cyclic shift values are given in Table 20-10 with $i_{ESS} = i_{STS}$

Q_k is defined in 20.3.11.11.2

Υ_k is defined in Equation (20-5) and Equation (20-6)

P_{HTLTF} the HT-LTF mapping matrix, is given by Equation (20-27)

$$P_{HTLTF} = \begin{bmatrix} 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \\ -1 & 1 & 1 & 1 \end{bmatrix} \quad (20\text{-}27)$$

20.3.9.5 HT-greenfield format preamble

20.3.9.5.1 General

For HT-greenfield operation, compatibility with Clause 18 and Clause 19 STAs is not required. Therefore, the portions of the preamble that are compatible with Clause 18 and Clause 19 STAs are not included. The result is a shorter and more efficient PLCP frame format that includes a STF, LTF(s), and an HT-SIG.

20.3.9.5.2 Cyclic shift definition for HT-greenfield format preamble

Throughout the HT-greenfield format preamble, cyclic shift is applied to prevent beamforming when similar signals are transmitted on different spatial streams. The same cyclic shift is applied to these streams during the transmission of the data portion of the frame. The values of the cyclic shift to be used during the HT-greenfield format preamble, as well as the data portion of the HT-greenfield format frame, are specified in Table 20-10.

20.3.9.5.3 HT-GF-STF definition

The HT-GF-STF is placed at the beginning of an HT-greenfield format frame. The time domain waveform for the HT-GF-STF on transmit chain i_{TX} shall be as shown in Equation (20-28).

$$r_{HT-GF-STF}^{(i_{TX})}(t) = \frac{1}{\sqrt{N_{STS} \cdot N_{HT-GF-STF}^{Tone}}} w_{\mathrm{T}_{HT-GF-STF}}(t) \cdot \sum_{k=-N_{SR}}^{N_{SR}} \sum_{i_{STS}=1}^{N_{STS}} [Q_k]_{i_{TX}, i_{STS}} [P_{HTLTF}]_{i_{STS}, 1} \Upsilon_k S_k \exp(j2\pi k \Delta_F (t - T_{CS}^{i_{STS}})) \tag{20-28}$$

where

$T_{CS}^{i_{STS}}$ represents the cyclic shift for the space-time stream i_{STS} and takes values from Table 20-10

Q_k is defined in 20.3.11.11.2

P_{HTLTF} is defined in Equation (20-27)

S_k is defined in non-HT-STF (L-STF), Equation (20-8) for 20 MHz operation and Equation (20-9) for 40 MHz operation

Υ_k is defined in Equation (20-5) and Equation (20-6)

The waveform defined by Equation (20-28) has a period of 0.8 µs, and the HT-GF-STF includes ten such periods, with a total duration of $\mathrm{T}_{HT-GF-STF}$ = 8 µs.

20.3.9.5.4 HT-greenfield format HT-SIG

The content and format of the HT-SIG of an HT-greenfield format frame is identical to the HT-SIG in an HT-mixed format frame, as described in 20.3.9.4.3. The placement of the HT-SIG in an HT-greenfield format frame is shown in Figure 20-1. In HT-greenfield format frames, the HT-SIG is transmitted with the same cyclic shifts and the same spatial mapping as the preceding portions of the preamble. This use of the same cyclic shifts and spatial mapping is done to accommodate the estimation of channel parameters needed to robustly demodulate and decode the information contained in the HT-SIG.

The time domain waveform for the HT-SIG on transmit chain i_{TX} with 20 MHz operation shall be as shown in Equation (20-29).

$$r_{HT-SIG}^{i_{TX}}(t) = \frac{1}{\sqrt{N_{STS} \cdot N_{HT-SIG}^{Tone}}} \sum_{n=0}^{1} w_{\mathrm{T}_{SYM}}(t - nT_{SYM}) \cdot \sum_{k=-26}^{26} \sum_{i_{STS}=1}^{N_{STS}} [Q_k]_{i_{TX}, i_{STS}} [P_{HTLTF}]_{i_{STS}, 1} (jD_{k,n} + p_n P_k) \cdot \exp(j2\pi k \Delta_F (t - nT_{SYM} - T_{GI} - T_{CS}^{i_{STS}})) \tag{20-29}$$

where

P_k and p_n are defined in 18.3.5.10

$D_{k,n}$ is defined in 20.3.9.4.3

$T_{CS}^{i_{STS}}$ represents the cyclic shift for space-time stream i_{STS} and takes values from Table 20-10

Q_k is defined in 20.3.11.11.2

P_{HTLTF} is defined in Equation (20-27)

The time domain waveform for the HT-SIG on transmit chain i_{TX} with 40 MHz operation shall be as shown in Equation (20-30).

$$\begin{aligned} r_{HT-SIG}^{i_{TX}}(t) &= \frac{1}{\sqrt{N_{STS}\cdot N_{HT-SIG}^{Tone}}}\sum_{n=0}^{1} w_{T_{SYM}}(t-nT_{SYM}) \\ &\cdot \sum_{k=-26}^{26}\sum_{i_{STS}=1}^{N_{STS}} [P_{HTLTF}]_{i_{STS},1}(jD_{k,n}+p_nP_k) \\ &\cdot ([Q_{k-32}]_{i_{TX},i_{STS}}\exp(j2\pi(k-32)\Delta_F(t-nT_{SYM}-T_{GI}-T_{CS}^{i_{STS}})) \\ &+ j[Q_{k+32}]_{i_{TX},i_{STS}}\exp(j2\pi(k+32)\Delta_F(t-nT_{SYM}-T_{GI}-T_{CS}^{i_{STS}}))) \end{aligned} \tag{20-30}$$

where

p_n and P_k are defined in 18.3.5.10

$D_{k,n}$ is defined in 20.3.9.4.3

$T_{CS}^{i_{STS}}$ represents the cyclic shift for space-time stream i_{STS} and takes values from Table 20-10

Q_k is defined in 20.3.11.11.2

P_{HTLTF} is defined in Equation (20-27)

20.3.9.5.5 HT-greenfield format LTF

The format of the LTF portion of the preamble in an HT-greenfield format frame is similar to that of the HT-LTF in an HT-mixed format frame, as described in 20.3.9.4.6, with the difference that the first HT-LTF (HT-LTF1) is twice as long (8 µs) as the other HT-LTFs. The time domain waveform for the long training symbol on transmit chain i_{TX} for the first HT-LTF in an HT-greenfield format frame shall be as shown in Equation (20-31).

$$\begin{aligned} r_{HT-LTF}^{1,i_{TX}}(t) &= \frac{1}{\sqrt{N_{STS}\cdot N_{HT-LTF}^{Tone}}} w_{T_{HT-LTF1}}(t) \\ &\cdot \sum_{k=-N_{SR}}^{N_{SR}}\sum_{i_{STS}=1}^{N_{STS}} [Q_k]_{i_{TX},i_{STS}}[P_{HTLTF}]_{i_{STS},1}\Upsilon_k HTLTF(k)\exp(j2\pi k\Delta_F(t-T_{GI2}-T_{CS}^{i_{STS}})) \end{aligned} \tag{20-31}$$

where

$T_{CS}^{i_{STS}}$ represents the cyclic shift for space-time stream i_{STS} and takes values from Table 20-10

Q_k is defined in 20.3.11.11.2

P_{HTLTF} is defined in Equation (20-27)

The first HT-LTF (HT-LTF1) consists of two periods of the long training symbol, preceded by a double-length (1.6 µs) cyclic prefix. The placement of the first and subsequent HT-LTFs in an HT-greenfield format frame is shown in Figure 20-1.

20.3.10 Transmission of NON_HT format PPDUs with more than one antenna

When an HT device transmits a NON_HT format PPDU with the MODULATION parameter equal to OFDM or ERP-OFDM using more than one transmit chain, it shall apply the cyclic shifts defined in Table 20-9 to the transmission in each chain.

20.3.11 Data field

20.3.11.1 General

When BCC encoding is used, the Data field consists of the 16-bit SERVICE field, the PSDU, either six or twelve tail bits, depending on whether one or two encoding streams are represented, and pad bits. When LDPC encoding is used, the Data field consists of the 16-bit SERVICE field and the PSDU, processed by the procedure in 20.3.11.7.5.

The number of OFDM symbols in the data field when BCC encoding is used is computed as shown in Equation (20-32).

$$N_{SYM} = m_{STBC}\left\lceil \frac{8 \cdot length + 16 + 6 \cdot N_{ES}}{m_{STBC} \cdot N_{DBPS}} \right\rceil \tag{20-32}$$

where

m_{STBC} is 2 if STBC is used and 1 otherwise (making sure that the number of symbols is even when STBC is used)

length is the value of the HT Length field in the HT-SIG field defined in Table 20-11

N_{DBPS} takes the values defined in Table 20-30 through Table 20-44

$\lceil x \rceil$ denotes the smallest integer greater than or equal to x

The number of “zero” pad bits is thus $N_{SYM} \times N_{DBPS} - 8 \times length - 16 - 6 \times N_{ES}$. The number of symbols in the data field when LDPC encoding is used is described in 20.3.11.7.

For LDPC encoding, the number of encoded data bits, N_{avbits}, is given by Equation (20-39); the number of OFDM symbols, N_{SYM}, is given by Equation (20-41); and the number of repeated encoded bits for padding, N_{rep}, is given by Equation (20-42), in 20.3.11.7.5.

20.3.11.2 SERVICE field

The SERVICE field is used for scrambler initialization. The SERVICE field is composed of 16 bits, all set to 0 before scrambling. In non-HT PPDUs, the SERVICE field is the same as in 18.3.5.2. In HT PPDUs, the SERVICE field is composed of 16 zero bits, scrambled by the scrambler, as defined in 20.3.11.3.

20.3.11.3 Scrambler

The data field shall be scrambled by the scrambler defined in 18.3.5.5 and initialized with a pseudorandom nonzero seed.

20.3.11.4 Coding

The Data field shall be encoded using either the BCC defined in 18.3.5.6 or the LDPC code defined in 20.3.11.7. The encoder is selected by the FEC coding field in the HT-SIG, as described in 20.3.9.4.3. A single FEC encoder is always used when LDPC coding is used. When the BCC FEC encoder is used, a single encoder is used, except that two encoders are used when the selected MCS has a PHY rate greater than 300 Mb/s (see 20.6). To determine whether to use one or two BCC FEC encoders, the rate is calculated based on the use of an 800 ns GI. The operation of the BCC FEC is described in 20.3.11.6. The operation of the LDPC coder is described in 20.3.11.7.

Support for the reception of BCC-encoded Data field frames is mandatory.

20.3.11.5 Encoder parsing operation for two BCC FEC encoders

If two BCC encoders are used, the scrambled data bits are divided between the encoders by sending alternating bits to different encoders. Bit with index i sent to the encoder j, denoted $x_i^{(j)}$, is shown in Equation (20-33).

$$x_i^{(j)} = b_{N_{ES} \cdot i + j} \qquad ; \qquad 0 \le j \le N_{ES} - 1 \tag{20-33}$$

Following the parsing operation, 6 scrambled “zero” bits following the end of the message bits in each BCC input sequence are replaced by unscrambled “zero” bits, as described in 18.3.5.3.

The replaced bits are shown in Equation (20-34).

$$x_i^{(j)} \qquad : \qquad 0 \le j \le N_{ES} - 1 \qquad ; \qquad \frac{length \cdot 8 + 16}{N_{ES}} \le i \le \frac{length \cdot 8 + 16}{N_{ES}} + 5 \tag{20-34}$$

20.3.11.6 Binary convolutional coding and puncturing

When BCC encoding is used, the encoder parser output sequences $\{x_i^0\}$, and $\{x_i^1\}$ where applicable, are each encoded by the rate 1/2 convolutional encoder defined in 18.3.5.6. After encoding, the encoded data shall be punctured by the method defined in 18.3.5.7 to achieve the rate selected by the MCS.

If rate 5/6 coding is selected, the puncturing scheme is defined in Figure 20-11.

20.3.11.7 LDPC codes

20.3.11.7.1 Introduction

HT LDPC codes are described in 20.3.11.7.2 through 20.3.11.7.6. These codes are optionally used in the HT system as a high-performance error correcting code instead of the convolutional code (20.3.11.6). The LDPC encoder shall use the rate-dependent parameters in Table 20-30 through Table 20-44, with the exception of the N_{ES} parameter.

Support for LDPC codes is optional.

20.3.11.7.2 LDPC coding rates and codeword block lengths

The supported coding rates, information block lengths, and codeword block lengths are described in Table 20-15.

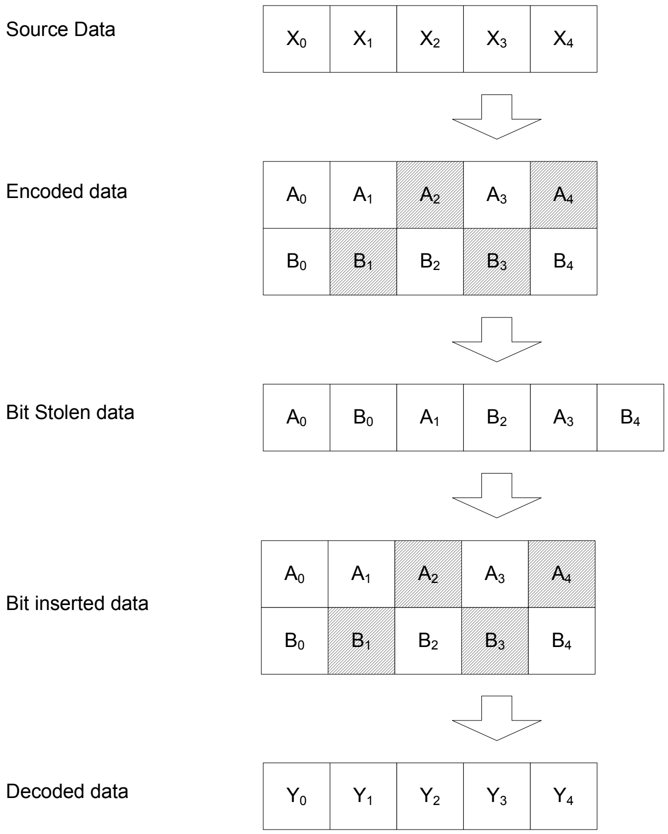

Source Data
X_0 X_1 X_2 X_3 X_4
Encoded data
A_0 A_1 A_2 A_3 A_4
B_0 B_1 B_2 B_3 B_4
Bit Stolen data
A_0 B_0 A_1 B_2 A_3 B_4
Bit inserted data
A_0 A_1 A_2 A_3 A_4
B_0 B_1 B_2 B_3 B_4
Decoded data
Y_0 Y_1 Y_2 Y_3 Y_4

Figure 20-11—Puncturing at rate 5/6

Table 20-15—LDPC parameters

Coding rate (R)	LDPC information block length (bits)	LDPC codeword block length (bits)
1/2	972	1944
1/2	648	1296
1/2	324	648
2/3	1296	1944
2/3	864	1296
2/3	432	648
3/4	1458	1944
3/4	972	1296
3/4	486	648
5/6	1620	1944
5/6	1080	1296
5/6	540	648

20.3.11.7.3 LDPC encoder

For each of the three available codeword block lengths, the LDPC encoder supports rate 1/2, rate 2/3, rate 3/4, and rate 5/6 encoding. The LDPC encoder is systematic, i.e., it encodes an information block, c=(i_0,i_1,..., $i_{(k-1)}$), of size *k,* into a codeword, **c,** of size *n,* **c**=(i_0,i_1,... $i_{(k-1)}$, p_0, p_1,..., $p_{(n-k-1)}$), by adding *n–k* parity bits obtained so that $\mathbf{H}\times\mathbf{c}^T = \mathbf{0}$, where **H** is an $(n-k)\times n$ parity-check matrix. The selection of the codeword block length (*n*) is achieved via the LDPC PPDU encoding process described in 20.3.11.7.5.

20.3.11.7.4 Parity-check matrices

Each of the parity-check matrices is partitioned into square subblocks (submatrices) of size $Z \times Z$. These submatrices are either cyclic-permutations of the identity matrix or null submatrices.

The cyclic-permutation matrix P_i is obtained from the $Z \times Z$ identity matrix by cyclically shifting the columns to the right by i elements. The matrix P_0 is the $Z \times Z$ identity matrix. Figure 20-12 illustrates examples (for a subblock size of 8 × 8) of cyclic-permutation matrices P_i.

$$P_0 = \begin{bmatrix} 1&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&0&0\\ 0&0&0&0&1&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&1 \end{bmatrix}, P_1 = \begin{bmatrix} 0&1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&0&0\\ 0&0&0&0&1&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&1\\ 1&0&0&0&0&0&0&0 \end{bmatrix}, P_5 = \begin{bmatrix} 0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&1\\ 1&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&0&0\\ 0&0&0&0&1&0&0&0 \end{bmatrix}$$

Figure 20-12—Examples of cyclic-permutation matrices with *Z*=8

Table F-1 displays the “matrix prototypes” of parity-check matrices for all four coding rates at block length *n*=648 bits. The integer *i* denotes the cyclic-permutation matrix P_i, as illustrated in Figure 20-12. Vacant entries of the table denote null (zero) submatrices.

Table F-2 displays the matrix prototypes of parity-check matrices for block length *n*=1296 bits, in the same fashion.

Table F-3 displays the matrix prototypes of parity-check matrices for block length *n*=1944 bits, in the same fashion.

20.3.11.7.5 LDPC PPDU encoding process

To encode an LDPC PPDU, step a) through step g) shall be performed in sequence:

a) Compute the number of available bits, N_{avbits}, in the minimum number of OFDM symbols in which the Data field of the packet may fit.

$$N_{pld} = length \times 8 + 16 \tag{20-35}$$

$$N_{avbits} = N_{CBPS} \times m_{STBC} \times \left\lceil \frac{N_{pld}}{N_{CBPS} \times R \times m_{STBC}} \right\rceil \tag{20-36}$$

where

m_{STBC} is 2 if STBC is used and 1 otherwise
length is the value of the HT Length field in the HT-SIG field defined in Table 20-11
$\lceil x \rceil$ denotes the smallest integer greater than or equal to *x*
N_{pld} is the number of bits in the PSDU and SERVICE field

b) Compute the integer number of LDPC codewords to be transmitted, N_{CW}, and the length of the codewords to be used, L_{LDPC} from Table 20-16.

Table 20-16—PPDU encoding parameters

Range of N_{avbits} (bits)	**Number of LDPC codewords (N_{CW})**	**LDPC codeword length L_{LDPC} (bits)**
$N_{avbits} \le 648$	1	1296, if $N_{avbits} \ge N_{pld} + 912 \times (1\text{-}R)$ 648, otherwise
$648 < N_{avbits} \le 1296$	1	1944, if $N_{avbits} \ge N_{pld} + 1464 \times (1\text{-}R)$ 1296, otherwise
$1296 < N_{avbits} \le 1944$	1	1944
$1944 < N_{avbits} \le 2592$	2	1944, if $N_{avbits} \ge N_{pld} + 2916 \times (1\text{-}R)$ 1296, otherwise
$2592 < N_{avbits}$	$\left\lceil \dfrac{N_{pld}}{1944 \cdot R} \right\rceil$	1944

c) Compute the number of shortening bits, N_{shrt}, to be padded to the N_{pld} data bits before encoding, as shown in Equation (20-37).

$$N_{shrt} = \max(0, (N_{CW} \times L_{LDPC} \times R) - N_{pld}) \tag{20-37}$$

When $N_{shrt} = 0$, shortening is not performed. (Note that N_{shrt} is inherently restricted to be non-negative due to the codeword length and count selection of 20-16). When $N_{shrt} > 0$, shortening bits shall be equally distributed over all N_{CW} codewords with the first $\mathrm{rem}(N_{shrt}, N_{CW})$ codewords shortened 1 bit more than the remaining codewords. Define $N_{spcw} = \lfloor N_{shrt} / N_{CW} \rfloor$. Then, when $N_{shrt} > 0$, the shortening is performed by setting information bits $i_{k - N_{spcw} - 1}, \ldots, i_{k-1}$ to 0 in the first $\mathrm{rem}(N_{shrt}, N_{CW})$ codewords and setting information bits $i_{k - N_{spcw}}, \ldots, i_{k-1}$ to 0 in the remaining codewords. For all values of N_{shrt}, encode each of the N_{CW} codewords using the LDPC encoding technique described in 20.3.11.7.2 through 20.3.11.7.4. When $N_{shrt} > 0$, the shortened bits shall be discarded after encoding.

d) Compute the number of bits to be punctured, N_{punc}, from the codewords after encoding, as shown in Equation (20-38).

$$N_{punc} = \max(0, (N_{CW} \times L_{LDPC}) - N_{avbits} - N_{shrt}) \tag{20-38}$$

If $\left((N_{punc} > 0.1 \times N_{CW} \times L_{LDPC} \times (1-R)) \text{ AND } \left(N_{shrt} < 1.2 \times N_{punc} \times \frac{R}{1-R}\right)\right)$ is true OR if $(N_{punc} > 0.3 \times N_{CW} \times L_{LDPC} \times (1-R))$ is true, increment N_{avbits} and recompute N_{punc} by the following two equations once:

$$N_{avbits} = N_{avbits} + N_{CBPS} \times m_{STBC} \tag{20-39}$$

$$N_{punc} = \max(0, (N_{CW} \times L_{LDPC}) - N_{avbits} - N_{shrt}) \tag{20-40}$$

The punctured bits shall be equally distributed over all N_{CW} codewords with the first $\text{rem}(N_{punc}, N_{CW})$ codewords punctured 1 bit more than the remaining codewords. Define $N_{\text{ppcw}} = \lfloor N_{\text{punc}} / N_{\text{CW}} \rfloor$. When $N_{\text{ppcw}} > 0$, the puncturing is performed by discarding parity bits $p_{n-k-N_{ppcw}-1}, \ldots, p_{n-k-1}$ of the first $\text{rem}(N_{punc}, N_{CW})$ codewords and discarding parity bits $(p_{n-k-N_{ppcw}}, \ldots, p_{n-k-1})$ of the remaining codewords after encoding. The number of OFDM symbols to be transmitted in the PPDU is computed as shown in Equation (20-41).

$$N_{SYM} = N_{avbits} / N_{CBPS} \tag{20-41}$$

e) Compute the number of coded bits to be repeated, N_{rep}, as shown in Equation (20-42).

$$N_{rep} = \max(0, N_{avbits} - N_{CW} \times L_{LDPC} \times (1-R) - N_{pld}) \tag{20-42}$$

The number of coded bits to be repeated shall be equally distributed over all N_{CW} codewords with one more bit repeated for the first $\text{rem}(N_{rep}, N_{CW})$ codewords than for the remaining codewords.

NOTE—When puncturing occurs, the coded bits are not repeated, and vice versa.

The coded bits to be repeated for any codeword shall be copied only from that codeword itself, starting from information bit i_0 and continuing sequentially through the information bits and, when necessary, into the parity bits, until the required number of repeated bits is obtained for that codeword. Note that these repeated bits are copied from the codeword after the shortening bits have been removed. If for a codeword the required number of repeated bits are not obtained in this manner (i.e., repeating the codeword once), the procedure is repeated until the required number is achieved. These repeated bits are then concatenated to the codeword after the parity bits in their same order. This process is illustrated in Figure 20-13. In this figure, the outlined arrows indicate the encoding procedure steps, while the solid arrows indicate the direction of puncturing and padding with repeated bits.

f) For each of the N_{CW} codewords, process the data using the number of shortening bits per codeword as computed in step c) for encoding, and puncture or repeat bits per codeword as computed per step d) and step e), as illustrated in Figure 20-13.

g) Aggregate all codewords and parse as defined in 20.3.11.7.6.

20.3.11.7.6 LDPC parser

The succession of LDPC codewords that result from the encoding process of 20.3.11.7.5 shall be converted into a bitstream in sequential fashion. Within each codeword, bit i_0 is ordered first. The parsing of this encoded data stream into spatial streams shall follow exactly the parsing rules defined for the BCC encoder, as defined in 20.3.11.8.1. However, the frequency interleaver of 20.3.11.8.3 is bypassed.

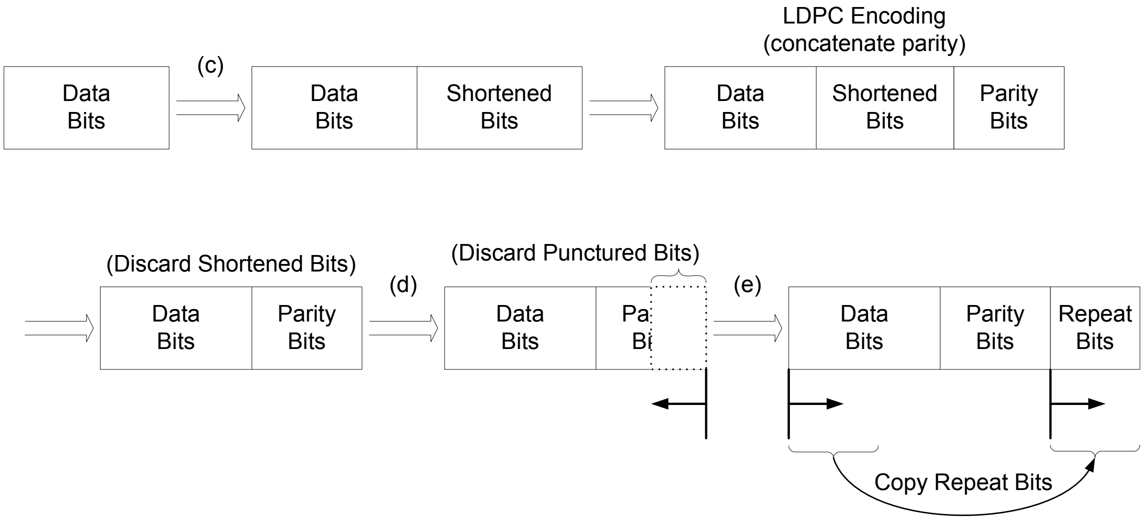


Figure 20-13—LDPC PPDU encoding padding and puncturing of a single codeword

20.3.11.8 Data interleaver

20.3.11.8.1 Overview

After coding and puncturing, the data bit streams at the output of the FEC encoders are rearranged into blocks of $N_{CBPSS}(i_{SS})$ bits, where $i_{SS} = 1, 2, \ldots N_{SS}$ is the spatial stream index. This operation is referred to as *stream parsing* and is described in 20.3.11.8.2. If BCC encoding was used, each of these blocks is then interleaved by an interleaver that is a modification of the Clause 18 interleaver.

20.3.11.8.2 Stream parser

The number of bits assigned to a single axis (real or imaginary) in a constellation point in spatial stream i_{SS} is denoted by Equation (20-43).

$$s(i_{SS}) = \max\left\{1, \frac{N_{BPSCS}(i_{SS})}{2}\right\} \tag{20-43}$$

The sum of these over all streams is $S = \sum_{i_{SS}=1}^{N_{SS}} s(i_{SS})$.

NOTE—If equal MCS is used for all spatial streams, this sum becomes $N_{SS} \cdot s$, where *s* is the number of bits for an axis common to all streams.

Consecutive blocks of $s(i_{SS})$ bits are assigned to different spatial streams in a round robin fashion.

If two encoders are present, the output of each encoder is used alternately for each round robin cycle, i.e., at the beginning *S* bits from the output of first encoder are fed into all spatial streams, and then *S* bits from the output of second encoder are used, and so on.

Input *k* to spatial stream i_{SS} shall be $y_i^{(j)}$, which is output bit *i* of the encoder *j*,

where

$$j = \left\lfloor \frac{k}{s(i_{SS})} \right\rfloor \bmod N_{ES} \qquad (20\text{-}44)$$

$$i = \sum_{i'=1}^{i_{SS}-1} s(i') + S \cdot \left\lfloor \frac{k}{N_{ES} \cdot s(i_{SS})} \right\rfloor + k \bmod s(i_{SS}) \qquad (20\text{-}45)$$

$1 \le i_{SS} \le N_{SS}$

$\lfloor x \rfloor$ is the largest integer less than or equal to x

z mod t is the remainder resulting from the division of integer z by integer t

For $i_{SS} = 1$, the first term in Equation (20-45) has a value of 0.

20.3.11.8.3 Frequency interleaver

MCS 32 interleaving shall be as defined in 18.3.5.7. Interleaving for all other MCSs is defined in this subclause.

The bits at the output of the stream parser are divided into blocks of $N_{CBPSS}(i_{SS})$, $i_{SS} = 1, 2, \ldots N_{SS}$ bits; and if BCC encoding was used, each block shall be interleaved by an interleaver based on the Clause 18 interleaver. This interleaver, which is based on entering the data in rows and reading them out in columns, has a different number of columns and rows depending on whether a 20 MHz channel or a 40 MHz channel is used. Table 20-17 defines the interleaver parameters. If LDPC encoding was used, no frequency interleaving is performed; hence the parsed streams are immediately mapped to symbols as defined in 20.3.11.9.

Table 20-17—Number of rows and columns in the interleaver

Parameter	20 MHz	40 MHz
N_{COL}	13	18
N_{ROW}	$4 \times N_{BPSCS}(i_{SS})$	$6 \times N_{BPSCS}(i_{SS})$
N_{ROT}	11	29

If more than one spatial stream exists after the operations based on the Clause 18 interleaver have been applied, a third operation called *frequency rotation* shall be applied to the additional spatial streams. The parameter for the frequency rotation is N_{ROT}.

The interleaving is defined using three permutations. The first permutation is defined by the rule shown in Equation (20-46).

$$i = N_{ROW}(k \bmod N_{COL}) + \mathrm{Floor}(k/N_{COL}) \qquad k = 0, 1, \ldots, N_{CBPSS}(i_{SS}) - 1 \qquad (20\text{-}46)$$

The second permutation is defined by the rule shown in Equation (20-47).

$$j = s(i_{SS}) \times \text{Floor}(i / s(i_{SS})) + (i + N_{CBPSS}(i_{SS}) - \text{Floor}(N_{COL} \times i / N_{CBPSS}(i_{SS}))) \bmod s(i_{SS}); \quad i = 0, 1, \ldots, N_{CBPSS}(i_{SS}) - 1 \tag{20-47}$$

The value of $s(i_{SS})$ is determined by the number of coded bits per subcarrier as shown in Equation (20-48).

$$s(i_{SS}) = \max(N_{BPSCS}(i_{SS})/2, 1) \tag{20-48}$$

If more than one spatial stream exists, a frequency rotation is applied to the output of the second permutation as shown in Equation (20-49).

$$r = \left(j - \left(((i_{SS} - 1) \times 2) \bmod 3 + 3 \times \text{Floor}\left(\frac{i_{SS} - 1}{3}\right)\right) \times N_{ROT} \times N_{BPSCS}(i_{SS})\right) \bmod N_{CBPSS}(i_{SS}); \quad j = 0, 1, \ldots, N_{CBPSS}(i_{SS}) - 1 \tag{20-49}$$

where

$i_{SS} = 1, 2, \ldots, N_{SS}$ is the index of the spatial stream on which this interleaver is operating

The deinterleaver uses the following operations to perform the inverse rotation. The index of the bit in the received block (per spatial stream) is denoted by *r*. The first permutation reverses the third (frequency rotation) permutation of the interleaver as shown in Equation (20-50).

$$j = \left(r + \left(((i_{SS} - 1) \times 2) \bmod 3 + 3 \times \text{Floor}\left(\frac{i_{SS} - 1}{3}\right)\right) \times N_{ROT} \times N_{BPSCS}(i_{SS})\right) \bmod N_{CBPSS}(i_{SS}); \quad r = 0, 1, \ldots, N_{CBPSS}(i_{SS}) - 1 \tag{20-50}$$

The second permutation reverses the second permutation in the interleaver as shown in Equation (20-51).

$$i = s(i_{SS}) \times \text{Floor}(j / s(i_{SS})) + (j + \text{Floor}(N_{COL} \times j / N_{CBPSS}(i_{SS}))) \bmod s(i_{SS}); \quad j = 0, 1, \ldots, N_{CBPSS}(i_{SS}) - 1 \tag{20-51}$$

where $s(i_{SS})$ is defined in Equation (20-48).

The third permutation reverses the first permutation of the interleaver as shown in Equation (20-52).

$$k = N_{COL} \times i - (N_{CBPSS}(i_{SS}) - 1) \times \text{Floor}(i / N_{ROW}) \qquad i = 0, 1, \ldots, N_{CBPSS}(i_{SS}) - 1 \tag{20-52}$$

20.3.11.9 Constellation mapping

20.3.11.9.1 General

The mapping between bits at the output of the interleaver and complex constellation points for BPSK, QPSK, 16-QAM, and 64-QAM follows the rules defined in 18.3.5.8.

The streams of complex numbers are denoted as shown in Equation (20-53).

$$d_{k,l,n}, 0 \le k \le N_{SD} - 1; 1 \le l \le N_{SS}; 0 \le n \le N_{SYM} - 1 \tag{20-53}$$

20.3.11.9.2 Space-time block coding (STBC)

This subclause defines a set of optional robust transmission formats that are applicable only when N_{STS} is greater than N_{SS}. In this case, N_{SS} spatial streams are mapped to N_{STS} space-time streams, which are mapped to N_{TX} transmit chains. These formats are based on STBC. When the use of STBC is indicated in the STBC field of the HT-SIG, a symbol operation shall occur between the constellation mapper and the spatial mapper (see Figure 20-3) as defined in this subclause.

If STBC is applied, the stream of complex numbers, $d_{k,i,n};k = 0\ldots N_{SD}-1;i = 1\ldots N_{SS};n = 0\ldots N_{SYM}-1$, generated by the constellation mapper, is the input of the STBC encoder, which produces as output the stream of complex numbers $\tilde{d}_{k,i,n};k = 0\ldots N_{SD}-1;i = 1\ldots N_{STS};n = 0\ldots N_{SYM}-1$. For given values of k and i, STBC processing operates on the complex modulation symbols in sequential pairs of OFDM symbols so that the value of $\tilde{d}_{k,i,2m}$ depends on $d_{k,i,2m}$ and $d_{k,i,2m+1}$, and $\tilde{d}_{k,i,2m+1}$ also depends on $d_{k,i,2m}$ and $d_{k,i,2m+1}$, as defined in Table 20-18.

Table 20-18—Constellation mapper output to spatial mapper input for STBC

N_{STS}	HT-SIG MCS field (bits 0–6 in HT-SIG$_1$)	N_{SS}	HT-SIG STBC field (bits 4–5 in HT-SIG$_2$)	i_{STS}	$\tilde{d}_{k,i,2m}$	$\tilde{d}_{k,i,2m+1}$
2	0–7	1	1	1	$d_{k,1,2m}$	$d_{k,1,2m+1}$
				2	$-d^*_{k,1,2m+1}$	$d^*_{k,1,2m}$
3	8–15, 33–38	2	1	1	$d_{k,1,2m}$	$d_{k,1,2m+1}$
				2	$-d^*_{k,1,2m+1}$	$d^*_{k,1,2m}$
				3	$d_{k,2,2m}$	$d_{k,2,2m+1}$
4	8–15	2	2	1	$d_{k,1,2m}$	$d_{k,1,2m+1}$
				2	$-d^*_{k,1,2m+1}$	$d^*_{k,1,2m}$
				3	$d_{k,2,2m}$	$d_{k,2,2m+1}$
				4	$-d^*_{k,2,2m+1}$	$d^*_{k,2,2m}$

Table 20-18—Constellation mapper output to spatial mapper input for STBC *(continued)*

N_{STS}	HT-SIG MCS field (bits 0–6 in HT-SIG$_1$)	N_{SS}	HT-SIG STBC field (bits 4–5 in HT-SIG$_2$)	i_{STS}	$\tilde{d}_{k,\,i,\,2m}$	$\tilde{d}_{k,\,i,\,2m+1}$
4	16–23, 39, 41, 43, 46, 48, 50	3	1	1	$d_{k,\,1,\,2m}$	$d_{k,\,1,\,2m+1}$
				2	$-d^{*}_{k,\,1,\,2m+1}$	$d^{*}_{k,\,1,\,2m}$
				3	$d_{k,\,2,\,2m}$	$d_{k,\,2,\,2m+1}$
				4	$d_{k,\,3,\,2m}$	$d_{k,\,3,\,2m+1}$
NOTE—the '*' operator represents the complex conjugate.						

If STBC is not applied, $\tilde{d}_{k,\,i,\,n} = d_{k,\,i,\,n}$ and $N_{SS} = N_{STS}$.

NOTE 1—The specific STBC schemes for single spatial streams ($N_{SS} = 1$) with $N_{TX} \geq 3$ are not detailed in this subclause since they are covered through the use of spatial expansion as detailed in 20.3.11.11.2.

NOTE 2—STBC is applied only for the HT-SIG MCS field values specified in Table 20-18 and is not used with MCS 32.

20.3.11.10 Pilot subcarriers

For a 20 MHz transmission, four pilot tones shall be inserted in the same subcarriers used in Clause 18, i.e., in subcarriers –21, –7, 7, and 21. The pilot sequence for the n^{th} symbols and $i_{STS}{}^{\text{th}}$ space-time stream shall be as shown in Equation (20-54).

$$
\begin{aligned}
P^{-28,28}_{(i_{STS},\,n)} = \Big\{ & 0, 0, 0, 0, 0, 0, 0, \Psi^{(N_{STS})}_{i_{STS},\,n \oplus 4}, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, \Psi^{(N_{STS})}_{i_{STS},\,(n+1) \oplus 4}, 0, 0, 0, 0, 0, 0, 0, \\
& 0, 0, 0, 0, 0, 0, \Psi^{(N_{STS})}_{i_{STS},\,(n+2) \oplus 4}, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, \Psi^{(N_{STS})}_{i_{STS},\,(n+3) \oplus 4}, 0, 0, 0, 0, 0, 0, 0 \Big\}
\end{aligned}
\tag{20-54}
$$

For a 40 MHz transmission (excluding MCS 32; see 20.3.11.11.5), pilot signals shall be inserted in subcarriers –53, –25, –11, 11, 25, and 53. The pilot sequence for symbol n and space-time stream i_{STS} shall be as shown in Equation (20-55).

$$
\begin{aligned}
P^{-58,58}_{(i_{STS},\,n)} = \Big\{ & 0, 0, 0, 0, 0, \Psi^{(N_{STS})}_{i_{STS},\,n \oplus 6}, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, \\
& 0, 0, \Psi^{(N_{STS})}_{i_{STS},\,(n+1) \oplus 6}, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, \Psi^{(N_{STS})}_{i_{STS},\,(n+2) \oplus 6}, 0, 0, 0, 0, 0, 0, 0, 0, 0, \\
& 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, \Psi^{(N_{STS})}_{i_{STS},\,(n+3) \oplus 6}, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, \Psi^{(N_{STS})}_{i_{STS},\,(n+4) \oplus 6}, \\
& 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, \Psi^{(N_{STS})}_{i_{STS},\,(n+5) \oplus 6}, 0, 0, 0, 0, 0 \Big\}
\end{aligned}
\tag{20-55}
$$

where $n \oplus a$ indicates symbol number modulo integer a and the patterns $\Psi_{i_{STS},n}^{(N_{STS})}$ are defined in Table 20-19 and Table 20-20.

NOTE—For each space-time stream, there is a different pilot pattern, and the pilot patterns are cyclically rotated over symbols.

The basic patterns are also different according to the total number of space-time streams for the packet.

Table 20-19—Pilot values for 20 MHz transmission

N_{STS}	i_{STS}	$\Psi_{i_{STS},0}^{(N_{STS})}$	$\Psi_{i_{STS},1}^{(N_{STS})}$	$\Psi_{i_{STS},2}^{(N_{STS})}$	$\Psi_{i_{STS},3}^{(N_{STS})}$
1	1	1	1	1	–1
2	1	1	1	–1	–1
2	2	1	–1	–1	1
3	1	1	1	–1	–1
3	2	1	–1	1	–1
3	3	–1	1	1	–1
4	1	1	1	1	–1
4	2	1	1	–1	1
4	3	1	–1	1	1
4	4	–1	1	1	1

Table 20-20—Pilots values for 40 MHz transmission (excluding MCS 32)

N_{STS}	i_{STS}	$\Psi_{i_{STS},0}^{(N_{STS})}$	$\Psi_{i_{STS},1}^{(N_{STS})}$	$\Psi_{i_{STS},2}^{(N_{STS})}$	$\Psi_{i_{STS},3}^{(N_{STS})}$	$\Psi_{i_{STS},4}^{(N_{STS})}$	$\Psi_{i_{STS},5}^{(N_{STS})}$
1	1	1	1	1	–1	–1	1
2	1	1	1	–1	–1	–1	–1
2	2	1	1	1	–1	1	1
3	1	1	1	–1	–1	–1	–1
3	2	1	1	1	–1	1	1
3	3	1	–1	1	–1	–1	1
4	1	1	1	–1	–1	–1	–1
4	2	1	1	1	–1	1	1
4	3	1	–1	1	–1	–1	1
4	4	–1	1	1	1	–1	1

20.3.11.11 OFDM modulation

20.3.11.11.1 General

The time domain signal is composed from the stream of complex numbers as shown in Equation (20-56)

$$\tilde{d}_{k,\,l,\,n},\ 0 \le k \le N_{SD} - 1;1 \le l \le N_{STS};0 \le n \le N_{SYM} - 1 \tag{20-56}$$

and from the pilot signals. For a 40 MHz transmission, the upper subcarriers are rotated 90° relative to the lower subcarriers.

20.3.11.11.2 Spatial mapping

The transmitter may choose to rotate and/or scale the constellation mapper output vector (or the space-time block coder output, if applicable). This rotation and/or scaling is useful in the following cases:

— When there are more transmit chains than space-time streams, $N_{STS} < N_{TX}$

— As part of (an optional) sounding packet

— As part of (an optional) calibration procedure

— When the packet is transmitted using one of the (optional) beamforming techniques

If the data to be transmitted on subcarrier k on space-time stream i_{STS} are $X_k^{(i_{STS})}$, the transmitted data on the transmit chain i_{TX} shall be as shown in Equation (20-57).

$$r_{Field}^{(i_{TX})} = \frac{1}{\sqrt{N_{STS} \cdot N_{Field}^{Tone}}} w_{T_{Field}}(t) \sum_{k = -N_{SR}}^{N_{SR}} \sum_{i_{STS} = 1}^{N_{STS}} [Q_k]_{i_{TX},\, i_{STS}} X_k^{(i_{STS})} \exp(j2\pi k \Delta_F (t - T_{CS}^{i_{STS}})) \tag{20-57}$$

where

$[Q_k]_{i_{TX},\, i_{STS}}$ is the element in row i_{TX} and column i_{STS} in a matrix Q_k with N_{TX} rows and N_{STS} columns; Q_k may be frequency dependent

Field is any field, as defined in 20.3.7, excluding L-STF, L-LTF, L-SIG, and HT-SIG in HT_MF format PPDU

Below are examples of spatial mapping matrices that might be used. There exist many other alternatives; implementation is not restricted to the spatial mapping matrices shown. The examples are:

a) *Direct mapping*: Q_k is a diagonal matrix of unit magnitude complex values that takes one of two forms:

   1) $Q_k = \mathbf{I}$, the identity matrix

   2) A CSD matrix in which the diagonal elements represent cyclic shifts in the time domain: $[Q_k]_{i,\, i} = \exp(-j2\pi k \Delta_F \tau_{CS}^i)$, where τ_{CS}^i, $i = 1, \ldots, N_{TX}$ represents the CSD applied.

b) *Indirect mapping:* Q_k may be the product of a CSD matrix and a unitary matrix such as the Hadamard matrix or the Fourier matrix.

c) *Spatial expansion:* Q_k is the product of a CSD matrix and a square matrix formed of orthogonal columns. As an illustration:

1) The spatial expansion may be performed by duplicating some of the N_{STS} streams to form the N_{TX} streams, with each stream being scaled by the normalization factor $\sqrt{N_{STS}/N_{TX}}$. The spatial expansion may be performed by using, for instance, one of the following matrices, denoted *D*, left-multiplied by a CSD matrix, denoted $M_{CSD}(k)$, and/or possibly multiplied by any unitary matrix. The resulting spatial mapping matrix is then $Q_k = M_{CSD}(k) \cdot D$, where *D* may take on one of the following values:

   i) N_{TX}=2, N_{STS}=1, $D = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \end{bmatrix}^T$

   ii) N_{TX}=3, N_{STS}=1, $D = \frac{1}{\sqrt{3}}\begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^T$

   iii) N_{TX}=4, N_{STS}=1, $D = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}^T$

   iv) N_{TX}=3, N_{STS}=2, $D = \sqrt{\frac{2}{3}}\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}$

   v) N_{TX}=4, N_{STS}=2, $D = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$

   vi) N_{TX}=4, N_{STS}=3, $D = \frac{\sqrt{3}}{2}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$

2) Different spatial expansion over subcarriers should be used in HT-mixed format only and with the smoothing bit equal to 0:

   i) N_{TX}=2, N_{STS}=1, $[Q_k]_{N_{STS}} = \begin{bmatrix} 1 & 0 \end{bmatrix}^T$ or $[Q_k]_{N_{STS}} = \begin{bmatrix} 0 & 1 \end{bmatrix}^T$

   ii) N_{TX}=3, N_{STS}=2, $[Q_k]_{N_{STS}} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$ or $\begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$

   iii) N_{TX}=4, N_{STS}=2, $[Q_k]_{N_{STS}} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$ or $\begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$

   iv) N_{TX}=4, N_{STS}=3, $[Q_k]_{N_{STS}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ or $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

d) *Beamforming steering matrix:* Q_k is any matrix that improves the reception in the receiver based on some knowledge of the channel between the transmitter and the receiver. With transmit beamforming with explicit feedback, the steering matrix Q_k is determined using either H_{eff} for CSI feedback or V_k for noncompressed and compressed matrices feedback from the STA to which the beamformed packet is addressed.

When there are fewer space-time streams than transmit chains, the first N_{STS} columns of the matrices above that are square might be used.

The same matrix Q_k shall be applied to subcarrier *k* during all parts of the packet in HT-greenfield format and all parts of the packet following and including the HT-STF field in an HT-mixed format packet. This operation is transparent to the receiver.

If 95% of the sum of the energy from all impulse responses of the time domain channels between all space-time streams and all transmit chain inputs, induced by the CSD added according to Table 20-10 and the frequency-dependence in the matrix Q_k, is contained within 800 ns, the smoothing bit should be set to 1. Otherwise, it shall be set to 0.

The CSD of Table 20-10 shall be applied at the input of the spatial mapper.

For the identity matrix direct mapping, the smoothing bit should be set to 1.

If no spatial mapping is applied, the matrix Q_k is equal to the identity matrix and $N_{STS} = N_{TX}$.

Sounding PPDUs using spatial expansion shall use an orthonormal column matrix Q_k. When the number of rows and columns is equal, the orthonormal column matrix becomes a unitary matrix.

20.3.11.11.3 Transmission in 20 MHz HT format

For 20 MHz HT transmissions, the signal from transmit chain i_{TX}, $1 \le i_{TX} \le N_{TX}$ shall be as shown in Equation (20-58).

$$
\begin{aligned}
r_{HT-DATA}^{i_{TX}}(t) &= \frac{1}{\sqrt{N_{STS} \cdot N_{HT-DATA}^{Tone}}} \sum_{n=0}^{N_{SYM}-1} w_{T_{SYM}}(t - nT_{SYM}) \\
&\cdot \sum_{k=-N_{SR}}^{N_{SR}} \sum_{i_{STS}=1}^{N_{STS}} ([Q_k]_{i_{TX}, i_{STS}} (\tilde{D}_{k, i_{STS}, n} + p_{n+z} P_{(i_{STS}, n)}^{k}) \\
&\cdot \exp(j2\pi k\Delta_F(t - nT_{SYM} - T_{GI} - T_{CS}^{i_{STS}})))
\end{aligned}
\qquad (20\text{-}58)
$$

where

z is 3 in an HT-mixed format packet and 2 in an HT-greenfield format packet

p_n is defined in 18.3.5.10

$$
\tilde{D}_{k, i_{STS}, n} = \begin{cases} 0, k = 0, \pm 7, \pm 21 \\ \tilde{d}_{M^r(k), i_{STS}, n}, \text{ otherwise} \end{cases}
$$

$$M^r(k) = \begin{cases} k+28, -28 \le k \le -22 \\ k+27, -20 \le k \le -8 \\ k+26, -6 \le k \le -1 \\ k+25, 1 \le k \le 6 \\ k+24, 8 \le k \le 20 \\ k+23, 22 \le k \le 28 \end{cases}$$

$P^k_{(i_{STS},\, n)}$ is defined in Equation (20-54)

20.3.11.11.4 Transmission in 40 MHz HT format

For 40 MHz HT transmissions, the signal from transmit chain i_{TX} shall be as shown in Equation (20-59).

$$r^{i_{TX}}_{HT-DATA}(t) = \frac{1}{\sqrt{N_{STS} \cdot N^{Tone}_{HT-DATA}}} \sum_{n=0}^{N_{SYM}-1} w_{T_{SYM}}(t - nT_{SYM})$$

$$\cdot \sum_{k=-N_{SR}}^{N_{SR}} \sum_{i_{STS}=1}^{N_{STS}} ([Q_k]_{i_{TX},\, i_{STS}}(\tilde{D}_{k,\, i_{STS},\, n} + p_{n+z}P^k_{(i_{STS},\, n)})\Upsilon_k \tag{20-59}$$

$$\cdot \exp(j2\pi k\Delta_F(t - nT_{SYM} - T_{GI} - T^{i_{STS}}_{CS})))$$

where

z is 3 in an HT-mixed format packet and 2 in an HT-greenfield format packet

p_n is defined in 18.3.5.10

$$\tilde{D}_{k,\, i_{STS},\, n} = \begin{cases} 0, k = 0, \pm 1, \pm 11, \pm 25, \pm 53 \\ \tilde{d}_{M^r(k),\, i_{STS},\, n}, \text{ otherwise} \end{cases}$$

$$M^r(k) = \begin{cases} k+58, -58 \le k \le -54 \\ k+57, -52 \le k \le -26 \\ k+56, -24 \le k \le -12 \\ k+55, -10 \le k \le -2 \\ k+52, 2 \le k \le 10 \\ k+51, 12 \le k \le 24 \\ k+50, 26 \le k \le 52 \\ k+49, 54 \le k \le 58 \end{cases}$$

$P^k_{(i_{STS},\, n)}$ is defined in Equation (20-55)

NOTE—The 90º rotation that is applied to the upper part of the 40 MHz channel is applied in the same way to the HT-STF, HT-LTF, and HT-SIG. The rotation applies to both pilots and the data in the upper part of the 40 MHz channel.

20.3.11.11.5 Transmission in MCS 32 format

MCS 32 format provides the lowest transmission rate in 40 MHz. It is used only for one spatial stream and only with BPSK modulation and rate 1/2 coding.

In the MCS 32 format, the signal shall be as shown in Equation (20-60).

$$
\begin{aligned}
r_{HT-DATA}^{i_{TX}}(t) = {} & \frac{1}{\sqrt{N_{HT-Duplicate}^{Tone}}} \sum_{n=0}^{N_{SYM}-1} w_{\mathrm{T}_{SYM}}(t - nT_{SYM}) \\
& \cdot \sum_{k=-N_{SR}}^{N_{SR}} (D_{k,n} + p_{n+z}P_k)([Q_{k-32}]_{i_{TX},1} \exp(j2\pi(k-32)\Delta_F(t - nT_{SYM} - T_{GI})) \\
& + j[Q_{k+32}]_{i_{TX},1} \exp(j2\pi(k+32)\Delta_F(t - nT_{SYM} - T_{GI})))
\end{aligned}
\qquad (20\text{-}60)
$$

where

z is defined in 20.3.11.11.3

P_k and p_n are defined in 18.3.5.10

$D_{k,n}$ is defined in 20.3.9.4.3

N_{SR} has the value defined for non-HT 20 MHz transmission

$[Q_k]_{i_{TX},1}$ is an element from a vector of length N_{TX}, which may be frequency dependent

$N_{HT-Duplicate}^{Tone}$ is defined in Table 20-8

The rules of spatial expansion CSD limitation, as specified in 20.3.11.11.2, shall apply to $[Q_k]_{i_{TX},1}$.

20.3.11.11.6 Transmission with a short GI

Short GI is used in the data field of the packet when the Short GI field in the HT-SIG is equal to 1. When it is used, the same formula for the formation of the signal shall be used as in 20.3.11.11.3, 20.3.11.11.4, and 20.3.11.11.5, with T_{GI} replaced by T_{GIS} and T_{SYM} replaced by T_{SYMS}.

NOTE—Short GI is not used in HT-greenfield format with one spatial stream, in which case the HT-SIG is immediately followed by data. It is very difficult to parse the HT-SIG in time to demodulate these data with the correct GI length if the GI length is not known in advance.

20.3.11.12 Non-HT duplicate transmission

Non-HT duplicate transmission is used to transmit to Clause 18 STAs, Clause 19 STAs, and Clause 20 STAs that may be present in either the upper or lower halves of the 40 MHz channel. The L-STF, L-LTF, and L-SIG shall be transmitted in the same way as in the HT 40 MHz transmission. The HT-SIG, HT-STF, and HT-LTF are not transmitted. Data transmission shall be as defined in Equation (20-61).

$$r^{i_{TX}}_{LEG-DUP}(t) = \frac{1}{\sqrt{N^{Tone}_{HT-Duplicate}}} \sum_{n=0}^{N_{SYM}-1} w_{T_{SYM}}(t-nT_{SYM})$$

$$\cdot \sum_{k=-26}^{26} (D_{k,n} + p_{n+1}P_k)(\exp(j2\pi(k-32)\Delta_F(t-nT_{SYM}-T_{GI}-T^{i_{TX}}_{CS})) \quad (20\text{-}61)$$

$$+ j\exp(j2\pi(k+32)\Delta_F(t-nT_{SYM}-T_{GI}-T^{i_{TX}}_{CS})))$$

where

P_k and p_n are defined in 18.3.5.10

$D_{k,n}$ is defined in 20.3.9.4.3

$T^{i_{TX}}_{CS}$ represents the cyclic shift of the transmit chain i_{TX} and is defined in Table 20-9

$N^{Tone}_{HT-Duplicate}$ is defined in Table 20-8

20.3.12 Beamforming

20.3.12.1 General

Beamforming is a technique in which the beamformer utilizes the knowledge of the MIMO channel to generate a steering matrix Q_k that improves reception in the beamformee.

The equivalent complex baseband MIMO channel model is one in which, when a vector $\mathbf{x}_k = [x_1, x_2, \ldots x_{N_{TX}}]^T$ is transmitted in subcarrier *k*, the received vector $\mathbf{y}_k = [y_1, y_2, \ldots y_{N_{RX}}]^T$ is modeled as shown in Equation (20-62).

$$\mathbf{y}_k = H_k\mathbf{x}_k + \mathbf{n} \quad (20\text{-}62)$$

where

H_k is channel matrix of dimensions $N_{RX} \times N_{TX}$

$\mathbf{n}$ is white (spatially and temporally) Gaussian noise as illustrated in Figure 20-14

When beamforming is used, the beamformer replaces $\mathbf{x}_k$, which in this case has $N_{STS} \le N_{TX}$ elements, with $Q_k\mathbf{x}_k$, where Q_k has N_{TX} rows and N_{STS} columns, so that the received vector is as shown in Equation (20-63).

$$\mathbf{y}_k = H_kQ_k\mathbf{x}_k + \mathbf{n} \quad (20\text{-}63)$$

The beamforming steering matrix that is computed (or updated) from a new channel measurement replaces the existing Q_k for the next beamformed data transmission. There are several methods of beamforming, differing in the way the beamformer acquires the knowledge of the channel matrices H_k and on whether the beamformer generates Q_k or the beamformee provides feedback information for the beamformer to generate Q_k.

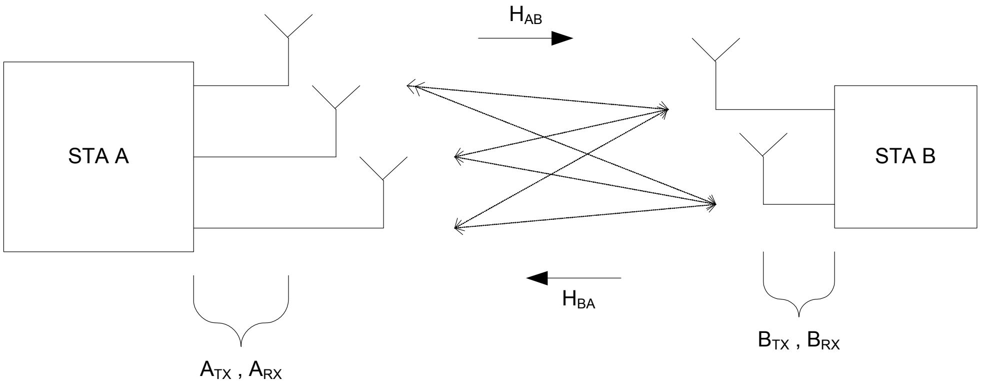


Figure 20-14—Beamforming MIMO channel model (3x2 example)

20.3.12.2 Implicit feedback beamforming

Implicit feedback beamforming is a technique that relies on reciprocity in the time division duplex channel to estimate the channel over which a device is transmitting based on the MIMO reference that is received from the device to which it plans to transmit. This technique allows the transmitting device to calculate a set of transmit steering matrices, Q_k, one for each subcarrier, which are intended to optimize the performance of the link.

Referring to Figure 20-14, beamforming transmissions from STA A to STA B using implicit techniques are enabled when STA B sends STA A a sounding PPDU, the reception of which allows STA A to form an estimate of the MIMO channel from STA B to STA A, for all subcarriers. For a TDD channel in which the forward and reverse channels are reciprocal, the channel from STA A to STA B in subcarrier k is the matrix transpose of the channel from STA B to STA A in subcarrier k to within a complex scaling factor, i.e., $H_{\mathrm{AB},k} = \rho[H_{\mathrm{BA},k}]^T$. Here $H_{\mathrm{AB},k}$ is the MIMO channel matrix from STA A to STA B at subcarrier k, and $H_{\mathrm{BA},k}$ is the channel matrix from STA B to STA A at subcarrier k. STA A uses this relationship to compute transmit steering matrices that are suitable for transmitting to STA B over $H_{\mathrm{AB},k}$.

NOTE—In order for the recipient of the sounding to compute steering matrices when steered or unsteered sounding is used, the steering matrices need to have the property $(H_kQ_k)(H_kQ_k)^H = H_kH_k^H$, where X^H indicates the conjugate transpose of the matrix X.

While the over-the-air channel between the antenna(s) at one STA and the antenna(s) at a second STA is reciprocal, the observed baseband-to-baseband channel used for communication may not be, as it includes the transmit and receive chains of the STAs. Differences in the amplitude and phase characteristics of the transmit and receive chains associated with individual antennas degrade the reciprocity of the over-the-air channel and cause degradation of performance of implicit beamforming techniques. The over-the-air calibration procedure described in 9.29.2.4 may be used to restore reciprocity. The procedure provides the means for calculating a set of correction matrices that can be applied at the transmit side of a STA to correct the amplitude and phase differences between the transmit and receive chains in the STA. If this correction is done at least at the STA that serves as the beamformer, there is sufficient reciprocity for implicit feedback in the baseband-to-baseband response of the forward link and reverse channel.

Figure 20-15 illustrates the observed baseband-to-baseband channel, including reciprocity correction. Spatial mapping matrices $Q_{\mathrm{A},k}$ and $Q_{\mathrm{B},k}$ are assumed to be identity matrices here for simplicity of illustration.

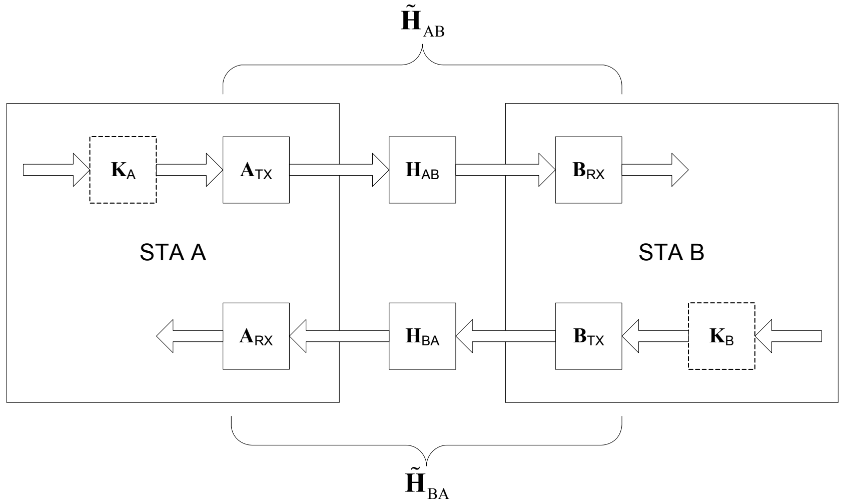


Figure 20-15—Baseband-to-baseband channel

NOTE—Spatial mapping matrix for sounding PPDUs are specified in 20.3.13.3.

The amplitude and phase responses of the transmit and receive chains can be expressed as diagonal matrices with complex valued diagonal entries, of the form $A_{TX,k}$ and $A_{RX,k}$ at STA A. The relationship between the baseband-to-baseband channel, $\tilde{H}_{\mathrm{AB},k}$, and the over-the-air channel, $H_{\mathrm{AB},k}$, is shown in Equation (20-64).

$$\tilde{H}_{\mathrm{AB},k} = B_{\mathrm{RX},k} H_{\mathrm{AB},k} A_{\mathrm{TX},k} \tag{20-64}$$

Similarly, the relationship between $\tilde{H}_{\mathrm{BA},k}$ and $H_{\mathrm{BA},k}$ is shown in Equation (20-65).

$$\tilde{H}_{\mathrm{BA},k} = A_{\mathrm{RX},k} H_{\mathrm{BA},k} B_{\mathrm{TX},k} \tag{20-65}$$

As an example, consider the case where calibration is performed at both STA A and STA B. The objective is to compute correction matrices, $K_{\mathrm{A},k}$ and $K_{\mathrm{B},k}$, that restore reciprocity so that Equation (20-66) is true.

$$\tilde{H}_{\mathrm{AB},k} K_{\mathrm{A},k} = \rho[\tilde{H}_{\mathrm{BA},k} K_{\mathrm{B},k}]^{\mathrm{T}} \tag{20-66}$$

The correction matrices are diagonal matrices with complex valued diagonal entries. The reciprocity condition in Equation (20-66) is enforced when Equation (20-67) and Equation (20-68) are true.

$$K_{\mathrm{A},k} = \alpha_{\mathrm{A},k}[A_{\mathrm{TX},k}]^{-1} A_{\mathrm{RX},k} \tag{20-67}$$

and

$$K_{\mathrm{B},k} = \alpha_{\mathrm{B},k}[B_{\mathrm{TX},k}]^{-1} B_{\mathrm{RX},k} \tag{20-68}$$

where $\alpha_{\mathrm{A},k}$ and $\alpha_{\mathrm{B},k}$ are complex valued scaling factors.

Using these expressions for the correction matrices, the calibrated baseband-to-baseband channel between STA A and STA B is expressed as shown in Equation (20-69).

$$\hat{H}_{AB,k} = \tilde{H}_{AB,k}K_{A,k} = \alpha_{A,k}B_{RX,k}H_{AB,k}A_{RX,k} \tag{20-69}$$

If both sides apply the correction matrices, the calibrated baseband-to-baseband channel between STA A and STA B is expressed as shown in Equation (20-70).

$$\hat{H}_{BA,k} = \alpha_{B,k}A_{RX,k}H_{BA,k}B_{RX,k} = \frac{\alpha_{B,k}}{\alpha_{A,k}}[\hat{H}_{AB,k}]^{T} \tag{20-70}$$

Focusing on STA A, the procedure for estimating $K_{A,k}$ is as follows:

a) STA A sends STA B a sounding PPDU, the reception of which allows STA B to estimate the channel matrices $\tilde{H}_{AB,k}$.

b) STA B sends STA A a sounding PPDU, the reception of which allows STA A to estimate the channel matrices $\tilde{H}_{BA,k}$.

c) STA B sends the quantized estimates of $\tilde{H}_{AB,k}$ to STA A.

d) STA A uses its local estimates of $\tilde{H}_{BA,k}$ and the quantized estimates of $\tilde{H}_{AB,k}$ received from STA B to compute the correction matrices $K_{A,k}$.

NOTE—When a nonidentity matrix is used for $Q_{A,k}$, STA A is responsible for accounting for the spatial mapping in its local channel estimate as well as in the quantized CSI fed back since the channel feedback received in step c) is actually $\tilde{H}_{AB,k}Q_{A,k}$ and not $\tilde{H}_{AB,k}$. Furthermore, since $Q_{B,k}$ is defined in 20.3.13.3, additional steps might be taken in STA A to remove the effect of $Q_{B,k}$ when computing the correction matrix $K_{A,k}$.

Steps a) and b) occur over a short time interval to ensure that the channel changes as little as possible between measurements. A similar procedure is used to estimate $K_{B,k}$ at STA B. The details of the computation of the correction matrices is implementation specific and beyond the scope of this standard.

20.3.12.3 Explicit feedback beamforming

20.3.12.3.1 General

In explicit beamforming, in order for STA A to transmit a beamformed packet to STA B, STA B measures the channel matrices and sends STA A either the effective channel, $H_{eff,k}$, or the beamforming feedback matrix, V_k, for STA A to determine a steering matrix, $Q_{steer,k} = Q_kV_k$, with V_k found from H_kQ_k, where Q_k is the orthonormal spatial mapping matrix that was used to transmit the sounding packet that elicited the V_k feedback. The effective channel, $H_{eff,k} = H_kQ_k$, is the product of the spatial mapping matrix used on transmit with the channel matrix. When new steering matrix $Q_{steer,k}$ is found, $Q_{steer,k}$ may replace Q_k for the next beamformed data transmission.

NOTE—$Q_{steer,k}$ is a mathematical term to update a new steering matrix for Q_k in the next beamformed data transmission.

20.3.12.3.2 CSI matrices feedback

In CSI matrices feedback, the beamformer receives the quantized MIMO channel matrix, H_{eff}, from the beamformee. The beamformer then may use this matrix to compute a set of transmit steering matrices, Q_k. The CSI matrix, H_{eff}, shall be determined from the transmitter spatial mapper input to the receiver FFT outputs. The beamformee shall remove the CSD in Table 20-10 from the measured channel matrix.

The matrices $H_{eff}(k)$, where k is the subcarrier index, are encoded so that applying the procedure in 20.3.12.3.3 optimally reconstructs the matrix.

20.3.12.3.3 CSI matrices feedback decoding procedure

The received, quantized matrix $H_{eff}^{q}(k)$ (of a specific subcarrier, k) shall be decoded as follows:

a) The real and imaginary parts of each element of the matrix, $H_{eff(m,l)}^{q(R)}(k)$ and $H_{eff(m,l)}^{q(I)}(k)$, are decoded as a pair of twos complement numbers to create the complex element, where $1 \le m \le N_r$ and $1 \le l \le N_c$.

b) Each element in the matrix of subcarrier k is then scaled using the value in the carrier matrix amplitude field (3 bits), $M_H(k)$, interpreted as a positive integer, in decibels, as follows:

   1) Calculate the linear value as defined in Equation (20-71).
   2) Calculate decoded values of the real and imaginary parts of the matrix element as defined in Equation (20-72) and Equation (20-73).

$$r(k) = 10^{M_H(k)/20} \tag{20-71}$$

$$\mathrm{Re}\{\tilde{H}_{eff(m,l)}(k)\} = \frac{H_{eff(m,l)}^{q(R)}(k)}{r(k)} \tag{20-72}$$

$$\mathrm{Im}\{\tilde{H}_{eff(m,l)}(k)\} = \frac{H_{eff(m,l)}^{q(I)}(k)}{r(k)} \tag{20-73}$$

20.3.12.3.4 Example of CSI matrices feedback encoding

The following is an example of an encoding process:

a) The maximums of the real and imaginary parts of each element of the matrix in each subcarrier are found, as defined by Equation (20-74).

$$m_H(k) = \max\left\{\max\left\{\left|\mathrm{Re}(H_{eff(m,l)}(k))\right|_{m=1,\,l=1}^{m=N_r,\,l=N_c}\right\}, \max\left\{\left|\mathrm{Im}(H_{eff(m,l)}(k))\right|_{m=1,\,l=1}^{m=N_r,\,l=N_c}\right\}\right\} \tag{20-74}$$

b) The scaling ratio is calculated and quantized to 3 bits as defined by Equation (20-75). A linear scaler is given by Equation (20-76).

$$M_H(k) = \min\left\{7, \left\lfloor 20\log_{10}\left(\frac{\max\{m_H(z)\}_{z=-N_{SR}}^{z=N_{SR}}}{m_H(k)}\right)\right\rfloor\right\} \tag{20-75}$$

where

$\lfloor x \rfloor$ is the largest integer smaller than or equal to x

$$M_H^{\text{lin}}(k) = \frac{\max\{m_H(z)\}_{z=-N_{SR}}^{z=N_{SR}}}{10^{M_H(k)/20}} \tag{20-76}$$

c) The real and imaginary parts of each element in the matrix $H_{eff(m,l)}(k)$ are quantized to N_b bits in twos complement encoding as defined by Equation (20-77) and Equation (20-78).

$$H_{eff(m,l)}^{q(R)}(k) = \text{round}\left(\frac{\text{Re}\{H_{eff(m,l)}(k)\}}{M_H^{\text{lin}}(k)}(2^{N_b-1}-1)\right) \tag{20-77}$$

$$H_{eff(m,l)}^{q(I)}(k) = \text{round}\left(\frac{\text{Im}\{H_{eff(m,l)}(k)\}}{M_H^{\text{lin}}(k)}(2^{N_b-1}-1)\right) \tag{20-78}$$

Each matrix is encoded using $3 + 2 \times N_b \times N_r \times N_c$ bits, where N_r and N_c are the number of rows and columns, respectively, in the channel matrix estimate computed by the receiving station and where N_b may have the value of 4, 5, 6, or 8 bits.

20.3.12.3.5 Noncompressed beamforming feedback matrix

In noncompressed beamforming feedback matrix, the beamformee shall remove the space-time stream CSD in Table 20-10 from the measured channel before computing a set of matrices for feedback to the beamformer. The beamforming feedback matrices, $V(k)$, found by the beamformee are sent to the beamformer in the order of real and imaginary components per tone as specified in 8.4.1.28. The beamformer might use these matrices to determine the steering matrices, Q_k.

The beamformee shall encode the matrices $V(k)$ so a beamformer applying the procedure below optimally reconstructs the matrix.

The received matrix $V^q(k)$ (of a specific subcarrier k) shall be decoded as follows:

a) The real and imaginary parts of each element of the matrix, $V_{m,l}^{q,R}$ and $V_{m,l}^{q,I}$, shall be decoded as a pair of twos complement numbers to create the complex element, where $1 \le m \le N_r$ and $1 \le l \le N_c$.

b) The dimensions of the beamforming feedback matrices are $N_r \times N_c$, where N_r and N_c are the number of rows and columns, respectively, in the beamforming feedback matrix computed by the receiving station. Each matrix is encoded using $2 \times N_b \times N_r \times N_c$ bits. N_b may have the value of 2, 4, 6, or 8 bits.

c) Columns $1 \ldots N_c$ of the beamforming feedback matrix correspond to spatial streams $1 \ldots N_c$, respectively. The mapping of spatial stream to modulation is defined in the MCS tables in 20.6. A transmitter shall not reorder the columns of the beamforming feedback matrices.

20.3.12.3.6 Compressed beamforming feedback matrix

In compressed beamforming feedback matrix, the beamformee shall remove the space-time stream CSD in Table 20-10 from the measured channel before computing a set of matrices for feedback to the beamformer. The beamforming feedback matrices, $V(k)$, found by the beamformee are compressed in the form of angles, which are sent to the beamformer. The beamformer might use these angles to decompress the matrices and determine the steering matrices Q_k.

The matrix V per tone shall be compressed as follows: The $N_r \times N_c$ beamforming feedback orthonormal column matrix V found by the beamformee shall be represented as shown in Equation (20-79). When the number of rows and columns is equal, the orthonormal column matrix becomes a unitary matrix.

$$V = \left[\prod_{i=1}^{\min(N_c,\, N_r - 1)} \left[D_i\left(1_{i-1}\ e^{j\phi_{i,i}}\ \ldots\ e^{j\phi_{N_r-1,i}}\ 1\right) \prod_{l=i+1}^{N_r} G_{li}^T(\psi_{li})\right]\right]\tilde{I}_{N_r \times N_c} \qquad (20\text{-}79)$$

The matrix $D_i\left(1_{i-1}\ e^{j\phi_{i,i}}\ \ldots\ e^{j\phi_{N_r-1,i}}\ 1\right)$ is an $N_r \times N_r$ diagonal matrix, where 1_{i-1} represents a sequence of ones with length of i–1, as shown in Equation (20-80).

$$D\left(1_{i-1}\ e^{j\phi_{i,i}} \ldots\ e^{j\phi_{N_r-1,i}}\ 1\right) = \begin{bmatrix} I_{i-1} & 0 & \ldots & \ldots & 0 \\ 0 & e^{j\phi_{i,i}} & 0 & \ldots & 0 \\ \vdots & 0 & \ddots & 0 & 0 \\ \vdots & \vdots & 0 & e^{j\phi_{N_r-1,i}} & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad (20\text{-}80)$$

The matrix $G_{li}(\psi)$ is an $N_r \times N_r$ Givens rotation matrix as shown in Equation (20-81).

$$G_{li}(\psi) = \begin{bmatrix} I_{i-1} & 0 & 0 & 0 & 0 \\ 0 & \cos(\psi) & 0 & \sin(\psi) & 0 \\ 0 & 0 & I_{l-i-1} & 0 & 0 \\ 0 & -\sin(\psi) & 0 & \cos(\psi) & 0 \\ 0 & 0 & 0 & 0 & I_{N_r-l} \end{bmatrix} \qquad (20\text{-}81)$$

where each I_m is an $m \times m$ identity matrix, and $\cos(\psi)$ and $\sin(\psi)$ are located at row l and column i. $\tilde{I}_{N_r \times N_c}$ is an identity matrix padded with 0s to fill the additional rows or columns when $N_r \neq N_c$.

For example, a 4×2 V matrix has the representation shown in Equation (20-82).

$$\begin{aligned} V = {} & \begin{bmatrix} e^{j\phi_{11}} & 0 & 0 & 0 \\ 0 & e^{j\phi_{21}} & 0 & 0 \\ 0 & 0 & e^{j\phi_{31}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \cos\psi_{21} & \sin\psi_{21} & 0 & 0 \\ -\sin\psi_{21} & \cos\psi_{21} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^T \times \begin{bmatrix} \cos\psi_{31} & 0 & \sin\psi_{31} & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\psi_{31} & 0 & \cos\psi_{31} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^T \times \begin{bmatrix} \cos\psi_{41} & 0 & 0 & \sin\psi_{41} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\sin\psi_{41} & 0 & 0 & \cos\psi_{41} \end{bmatrix}^T \\ & \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{j\phi_{22}} & 0 & 0 \\ 0 & 0 & e^{j\phi_{32}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\psi_{32} & \sin\psi_{32} & 0 \\ 0 & -\sin\psi_{32} & \cos\psi_{32} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^T \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\psi_{42} & 0 & \sin\psi_{42} \\ 0 & 0 & 1 & 0 \\ 0 & -\sin\psi_{42} & 0 & \cos\psi_{42} \end{bmatrix}^T \times \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \end{aligned} \qquad (20\text{-}82)$$

The procedure for finding a compressed V matrix is described as follows:

A $N_r \times N_c$ beamforming feedback orthonormal column matrix V is column-wise phase invariant because the steering matrix needs a reference in phase per each column. When the number of rows and columns is equal, the orthonormal column matrix becomes a unitary matrix. In other words, V is equivalent to $\tilde{V}\tilde{D}$, where $\tilde{D}$ is a column-wise phase shift matrix such as $\tilde{D} = \mathrm{diag}\left(e^{j\theta_1}, e^{j\theta_2}, \ldots, e^{j\theta_{N_c}}\right)$. When the beamformee estimates the channel, it may find $\tilde{V}$ for the beamforming feedback matrix for the beamformer, but it should send $\tilde{V}\tilde{D}$ back to the beamformer, where $V = \tilde{V}\tilde{D}$. The angle, θ_i, in $\tilde{D}$ is found to make the last row of $\tilde{V}\tilde{D}$ to be non-negative real numbers.

The angles $\phi_{1,1} \ldots \phi_{N_r-1,1}$ in the diagonal matrix $D_1\left(e^{j\phi_{11}} \; \ldots \; e^{j\phi_{N_r-1,1}} \; 1 \right)^*$ shall satisfy the constraint that all elements in the first column of $D_1^* V$ are non-negative real numbers. Now, the first column of $(G_{N_r 1} \ldots G_{31} G_{21} D_1^*) \times V$ can be $\begin{bmatrix} 1 & 0 & \ldots & 0 \end{bmatrix}^T$ by the Givens rotations G_{l1} such as shown in Equation (20-83).

$$\begin{bmatrix} \cos\psi_{N_r 1} & 0 & 0 & \sin\psi_{N_r 1} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\sin\psi_{N_r 1} & 0 & 0 & \cos\psi_{N_r 1} \end{bmatrix} \cdots \begin{bmatrix} \cos\psi_{31} & 0 & \sin\psi_{31} & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\psi_{31} & 0 & \cos\psi_{31} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\psi_{21} & \sin\psi_{21} & 0 & 0 \\ -\sin\psi_{21} & \cos\psi_{21} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} e^{j\phi_{11}} & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 \\ 0 & 0 & e^{j\phi_{N_r-1,1}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^* \times V = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & & & \\ 0 & & V_2 & \\ 0 & & & \end{bmatrix} \tag{20-83}$$

For a new $(N_r - 1) \times (N_c - 1)$ submatrix V_2, this process is applied in the same way. Then, the angles $\phi_{2,2} \ldots \phi_{N_r-1,2}$ in the diagonal matrix $D_2\left(1 \; e^{j\phi_{22}} \; \ldots \; e^{j\phi_{N_r-1,2}} \; 1 \right)^*$ shall satisfy the constraint that all elements in the second column of $D_2^* \times \mathrm{diag}(1, V_2)$ are non-negative real numbers. Now, the first two columns of $(G_{N_r 2} \ldots G_{32} D_2^*)(G_{N_r 1} \ldots G_{31} G_{21} D_1^*) \times V$ can be $\tilde{I}_{N_r \times 2}$ by the Givens rotations G_{l2} such as shown in Equation (20-84).

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\psi_{N_r 2} & 0 & \sin\psi_{N_r 2} \\ 0 & 0 & 1 & 0 \\ 0 & -\sin\psi_{N_r 2} & 0 & \cos\psi_{N_r 2} \end{bmatrix} \cdots \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\psi_{32} & \sin\psi_{32} & 0 \\ 0 & -\sin\psi_{32} & \cos\psi_{32} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{j\phi_{22}} & 0 & 0 \\ 0 & 0 & e^{j\phi_{N_r-1,2}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^* \times G_{N,1} \ldots G_{31} G_{21} D_1^* \times V = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & & \\ 0 & 0 & & V_3 \end{bmatrix} \tag{20-84}$$

This process continues until the first N_c columns of the right side matrix become $\tilde{I}_{N_r \times N_c}$. When $N_c < N_r$, this process does not need to continue because $V_{N_c + 1}$ is nulled out by $\tilde{I}_{N_r \times N_c}$. Then, by multiplying the complex conjugate transpose of the products of the D_i and G_{li} matrices on the left, V can be expressed as shown in Equation (20-85).

$$V = D_1 G_{21}^T G_{31}^T \ldots G_{N_r 1}^T \times D_2 G_{32}^T G_{42}^T \ldots G_{N_r 2}^T \times \ldots \times D_p G_{p+1,p}^T G_{p+2,p}^T \ldots G_{N_r p}^T \times \tilde{I}_{N_r \times N_C} \qquad (20\text{-}85)$$

where $p = \min(N_c, N_r - 1)$, which can be written in short form as in Equation (20-79).

The angles found from the decomposition process above, e.g., the values of $\psi_{i,j}$ and $\phi_{k,l}$, are quantized as described in 8.5.12.8.

Columns $1 \ldots N_c$ of the beamforming feedback matrix correspond to spatial streams $1 \ldots N_c$, respectively. The mapping of spatial stream to modulation is defined in the MCS tables in 20.6. A transmitter shall not reorder the columns of the beamforming feedback matrices in determining steering matrices.

20.3.13 HT Preamble format for sounding PPDUs

20.3.13.1 General

The MIMO channel measurement takes place in every PPDU as a result of transmitting the HT-LTFs as part of the PLCP preamble. The number of HT-LTFs transmitted shall be determined by the number of space-time streams transmitted unless additional dimensions are optionally sounded using HT-ELTFs and these are transmitted using the same spatial transformation that is used for the Data field of the HT PPDU. The use of the same spatial transformation enables the computation of the spatial equalization at the receiver.

When the number of space-time streams, N_{STS}, is less than the number of transmit antennas, or less than $\min(N_{TX}, N_{RX})$, sending only N_{STS} HT-LTFs does not allow the receiver to recover a full characterization of the MIMO channel, even though the resulting MIMO channel measurement is sufficient for receiving the Data field of the HT PPDU.

However, there are several cases where it is desirable to obtain as full a characterization of the channel as possible, thus requiring the transmission of a sufficient number of HT-LTFs to sound the full dimensionality of the channel, which is in some cases N_{TX} and in other cases $\min(N_{TX}, N_{RX})$. These cases of MIMO channel measurement are referred to as *MIMO channel sounding*. A sounding packet may be used to sound available channel dimensions. A sounding PPDU is identified by setting the Not Sounding field in the HT-SIG to 0. A sounding PPDU may have any allowed number of HT-LTFs satisfying $N_{LTF} \geq N_{STS}$. In general, if the Not Sounding field in the HT-SIG is equal to 0 and $N_{LTF} > N_{STS}$, HT-ELTFs are used, except where $N_{SS} = 3$ and $N_{LTF} = 4$ or in an NDP.

20.3.13.2 Sounding with a NDP

A STA may sound the channel using a NDP (indicated by the HT Length field in the HT-SIG equal to 0) with the Not Sounding field equal to 0. The number of LTFs is the number implied by the MCS, which shall indicate two or more spatial streams. The last HT-LTF of an NDP shall not be followed by a Data field (see Figure 20-16).

It is optional for a STA to process an NDP.

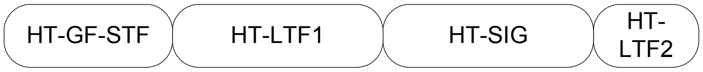


Figure 20-16—Example of an NDP used for sounding

20.3.13.3 Sounding PPDU for calibration

In the case of a bidirectional calibration exchange, two STAs exchange sounding PPDUs, the exchange of which enables the receiving STA to compute an estimate of the MIMO channel matrix H_k for each subcarrier k. In general, in an exchange of calibration messages, the number of spatial streams is less than the number of transmit antennas. In such cases, HT-ELTFs are used. In the case of sounding PPDUs for calibration, the antenna mapping matrix shall be as shown in Equation (20-86).

$$Q_k = C_{CSD}(k)P_{CAL} \tag{20-86}$$

where

$C_{CSD}(k)$ is a diagonal cyclic shift matrix in which the diagonal elements carry frequency-domain representation of the cyclic shifts given in Table 20-9

P_{CAL} is one of the following unitary matrices:

For $N_{TX} = 1, P_{CAL} = 1$

For $N_{TX} = 2, P_{CAL} = \frac{\sqrt{2}}{2}\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$

For $N_{TX} = 3, P_{CAL} = \frac{\sqrt{3}}{3}\begin{bmatrix} 1 & 1 & 1 \\ 1 & e^{-j2\pi/3} & e^{-j4\pi/3} \\ 1 & e^{-j4\pi/3} & e^{-j2\pi/3} \end{bmatrix}$

For $N_{TX} = 4, P_{CAL} = \frac{1}{2}\begin{bmatrix} 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \\ -1 & 1 & 1 & 1 \end{bmatrix}$

20.3.13.4 Sounding PPDU for channel quality assessment

In response to the reception of an MRQ, sent by STA A to STA B, the responding STA B returns to the requesting STA A an MCS selection that STA B determines to be a suitable MCS for STA A to use in subsequent transmissions to STA B. In determining the MCS, STA B performs a channel quality assessment, which entails using whatever information STA B has about the channel, such as an estimate of the MIMO channel derived from the sounding PPDU that carries the MRQ. To enable this calculation, the MRQ is sent in conjunction with a sounding PPDU.

The STA sending the MRQ (STA A) determines how many HT-LTFs to send, and whether to use HT-ELTFs or an NDP, based on the Transmit Beamforming Capabilities field, number of space-time streams used in the PPDU carrying the MRQ, the number of transmit chains it is using (N_{TX}), whether the transmit and receive STAs support STBC, and in some cases, the number of receive chains at the responding STA (N_{RX}).

The maximum number of available space-time streams is set by the number of transmit and receive chains and the STBC capabilities of the transmitter and receiver, as is shown in Table 20-21. While the number of receive chains at a STA is not communicated in a capabilities indicator, the maximum number of space-time streams supported may be inferred from the MCS capabilities and the STBC capabilities of the receiving STA. When the number of receive chains is known at the transmitter, the number of HT-LTFs sent to obtain a full channel quality assessment is determined according to the maximum number of space-time streams indicated in Table 20-21. The number of HT-LTFs to use in conjunction with the indicated number of space-time streams is determined according to 20.3.9.4.6.

Table 20-21—Maximum available space-time streams

N_{TX}	N_{RX}	$N_{STS,\max}$ without STBC	$N_{STS,\max}$ with STBC
1	1	1	N/A
2	1	1	2
3	1	1	2
3	2	2	3
4	1	1	2
4	2	2	4

If the requesting STA A sends an MRQ in a PPDU that uses fewer space-time streams in the data portion than the maximum number of space-time streams possible given the number of antennas at STA A and the responding STA B, the channel quality assessment made by STA B may be based on the HT-DLTFs alone. In this case, the MFB is limited to MCSs using the number of streams used in the Data field of the HT PPDU, or fewer. To determine whether an MCS should be chosen that uses more spatial streams than the PPDU containing the MRQ, it is necessary for the requesting STA A to either use HT-ELTFs (i.e., send the MRQ in a staggered sounding PPDU) or use an NDP (i.e., send the MRQ in conjunction with an NDP).

The sounding PPDU may have nonidentity spatial mapping matrix Q_k. For different receiving STAs, Q_k may vary.

20.3.14 Regulatory requirements

Wireless LANs (WLANs) implemented in accordance with this standard are subject to equipment certification and operating requirements established by regional and national regulatory administrations. The PMD specification establishes minimum technical requirements for interoperability, based upon established regulations at the time this standard was issued. These regulations are subject to revision or may be superseded. Requirements that are subject to local geographic regulations are annotated within the PMD specification. Regulatory requirements that do not affect interoperability are not addressed in this standard. Implementers are referred to the regulatory sources in Annex D for further information. Operation in countries within defined regulatory domains may be subject to additional or alternative national regulations.

20.3.15 Channel numbering and channelization

20.3.15.1 General

The STA may operate in the 5 GHz band and/or 2.4 GHz band. When using 20 MHz channels, it uses channels defined in 18.3.8.4 (5 GHz band) or 17.4.6 (2.4 GHz band). When using 40 MHz channels, it can operate in the channels defined in 20.3.15.2 and 20.3.15.3.

The set of valid operating channel numbers by regulatory domain is defined in Annex E.

20.3.15.2 Channel allocation in the 2.4 GHz Band

Channel center frequencies are defined at every integral multiple of 5 MHz in the 2.4 GHz band. The relationship between center frequency and channel number is given by Equation (20-87).

$$\text{Channel center frequency} = 2407 + 5 \times n_{ch}(\text{MHz}) \tag{20-87}$$

where
n_{ch} = 1, 2, …, 13

20.3.15.3 Channel allocation in the 5 GHz band

Channel center frequencies are defined at every integral multiple of 5 MHz above 5 GHz. The relationship between center frequency and channel number is given in Equation (20-88).

$$\text{Channel center frequency} = \text{Channel starting frequency} + 5 \times n_{ch}(\text{MHz}) \tag{20-88}$$

where
n_{ch} = 1, …, 200

Channel starting frequency is defined as dot11ChannelStartingFactor × 500 kHz or is defined as 5.000 GHz for systems where dot11OperatingClassesRequired is false or not defined. A channel center frequency of 5.000 GHz shall be indicated by dot11ChannelStartingFactor = 8000 and n_{ch} = 200.

20.3.15.4 40 MHz channelization

The set of valid operating channel numbers by regulatory domain is defined in Annex E.

The 40 MHz channels are specified by two fields: (*Nprimary_ch*, *Secondary*). The first field represents the channel number of the primary channel, and the second field indicates whether the secondary channel is above or below the primary channel (1 indicates above, –1 indicates below). The secondary channel number shall be $Nprimary\_ch + Secondary \times 4$.

For example, a 40 MHz channel consisting of 40 MHz channel number 36 and Secondary 1 specifies the primary channel is 36 and the secondary channel is 40.

20.3.16 Transmit and receive in-band and out-of-band spurious transmissions

The OFDM PHY shall conform to in-band and out-of-band spurious emissions as set by regulatory bodies.

20.3.17 Transmitter RF delay

The transmitter RF delay shall follow 18.3.8.6.

20.3.18 Slot time

The slot time shall follow 18.3.8.7 for 5 GHz bands and 19.4.5 for 2.4 GHz bands.

20.3.19 Transmit and receive port impedance

The transmit and receive antenna port impedance for each transmit and receive antenna shall follow 18.3.8.8.

20.3.20 PMD transmit specification

20.3.20.1 Transmit spectrum mask

NOTE 1—In the presence of additional regulatory restrictions, the device has to meet both the regulatory requirements and the mask defined in this subclause, i.e., its emissions can be no higher at any frequency offset than the minimum of the values specified in the regulatory and default masks.

NOTE 2—The transmit spectral mask figures in this subclause are not drawn to scale.

For the 2.4 GHz band, when transmitting in a 20 MHz channel, the transmitted spectrum shall have a 0 dBr (dB relative to the maximum spectral density of the signal) bandwidth not exceeding 18 MHz, –20 dBr at 11 MHz frequency offset, –28 dBr at 20 MHz frequency offset, and the maximum of –45 dBr and –53 dBm/MHz at 30 MHz frequency offset and above. The transmitted spectral density of the transmitted signal shall fall within the spectral mask, as shown in Figure 20-17. The measurements shall be made using a 100 kHz resolution bandwidth and a 30 kHz video bandwidth.

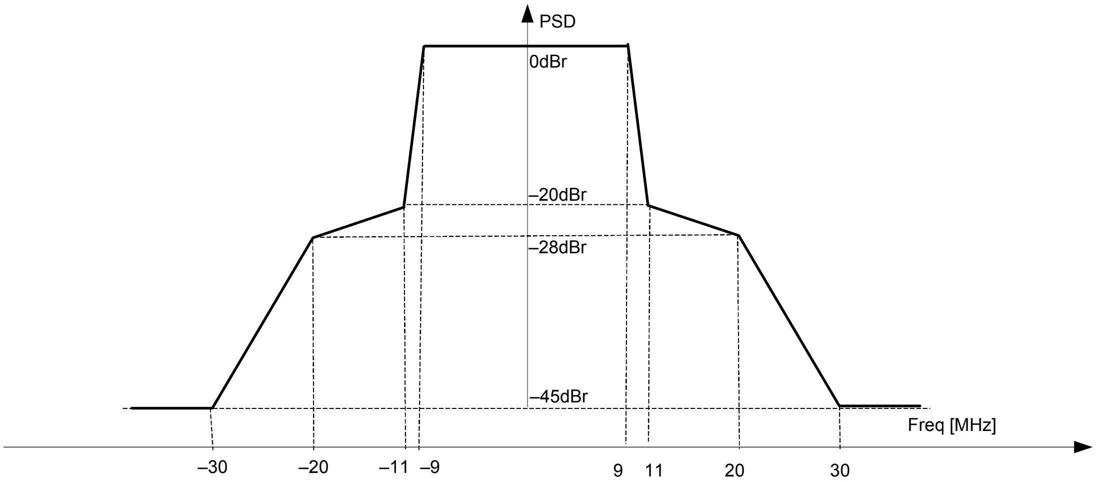


Figure 20-17—Transmit spectral mask for 20 MHz transmission in the 2.4 GHz band

For the 2.4 GHz band, when transmitting in a 40 MHz channel, the transmitted spectrum shall have a 0 dBr bandwidth not exceeding 38 MHz, –20 dBr at 21 MHz frequency offset, –28 dBr at 40 MHz offset, and the maximum of –45 dBr and –56 dBm/MHz at 60 MHz frequency offset and above. The transmitted spectral density of the transmitted signal shall fall within the spectral mask, as shown in Figure 20-18.

For the 5 GHz band, when transmitting in a 20 MHz channel, the transmitted spectrum shall have a 0 dBr (dB relative to the maximum spectral density of the signal) bandwidth not exceeding 18 MHz, –20 dBr at 11 MHz frequency offset, –28 dBr at 20 MHz frequency offset, and the maximum of –40 dBr and –53 dBm/MHz at 30 MHz frequency offset and above. The transmitted spectral density of the transmitted signal shall fall within the spectral mask, as shown in Figure 20-19. The measurements shall be made using a 100 kHz resolution bandwidth and a 30 kHz video bandwidth.

For the 5 GHz band, when transmitting in a 40 MHz channel, the transmitted spectrum shall have a 0 dBr bandwidth not exceeding 38 MHz, –20 dBr at 21 MHz frequency offset, –28 dBr at 40 MHz offset, and the maximum of –40 dBr and –56 dBm/MHz at 60 MHz frequency offset and above. The transmitted spectral density of the transmitted signal shall fall within the spectral mask, as shown in Figure 20-20.

Transmission with CH_OFF_20U, CH_OFF_20L, or CH_OFF_40 shall conform to the same mask that is used for the 40 MHz channel.

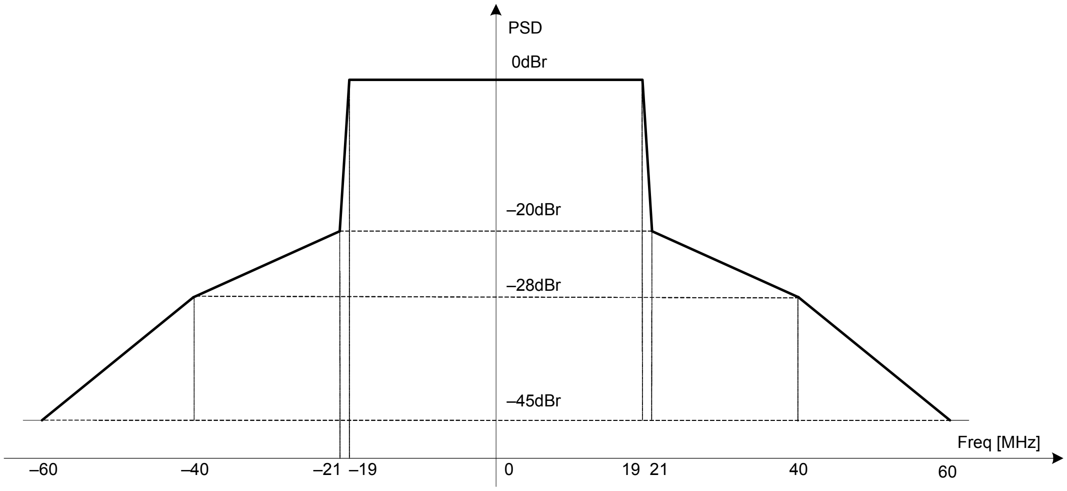


Figure 20-18—Transmit spectral mask for a 40 MHz channel in the 2.4 GHz band

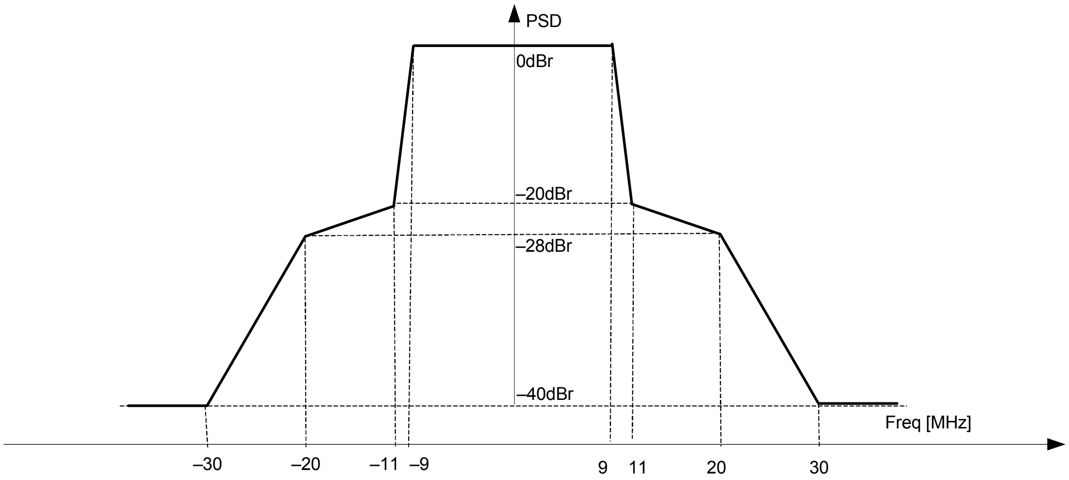


Figure 20-19—Transmit spectral mask for 20 MHz transmission in the 5 GHz band

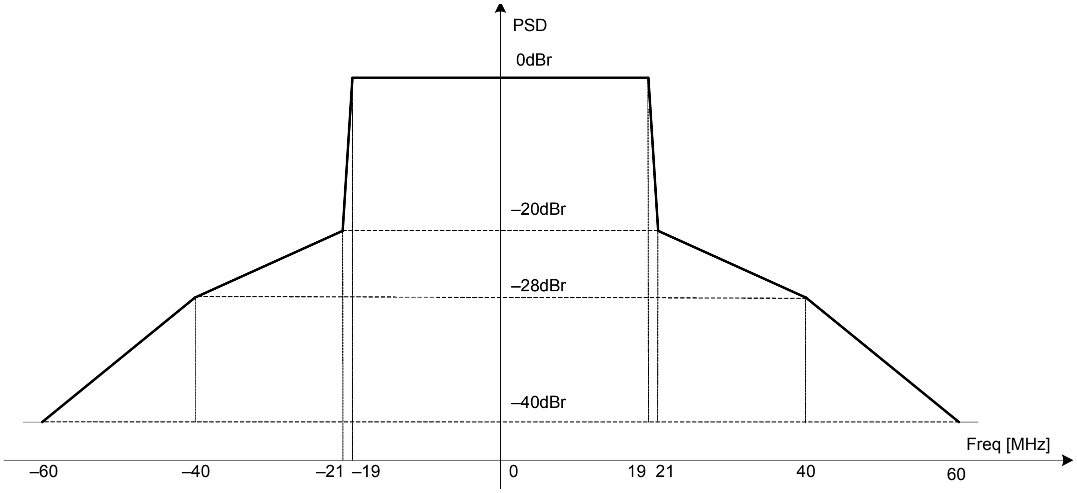


Figure 20-20—Transmit spectral mask for a 40 MHz channel in the 5 GHz band

20.3.20.2 Spectral flatness

In a 20 MHz channel and in corresponding 20 MHz transmission in a 40 MHz channel, the average energy of the constellations in each of the subcarriers with indices –16 to –1 and +1 to +16 shall deviate no more than ± 4 dB from their average energy. The average energy of the constellations in each of the subcarriers with indices –28 to –17 and +17 to +28 shall deviate no more than +4/–6 dB from the average energy of subcarriers with indices –16 to –1 and +1 to +16.

In a 40 MHz transmission (excluding PPDUs in MCS 32 format and non-HT duplicate format), the average energy of the constellations in each of the subcarriers with indices –42 to –2 and +2 to +42 shall deviate no more than ± 4 dB from their average energy. The average energy of the constellations in each of the subcarriers with indices –43 to –58 and +43 to +58 shall deviate no more than +4/–6 dB from the average energy of subcarriers with indices –42 to –2 and +2 to +42.

In MCS 32 format and non-HT duplicate format, the average energy of the constellations in each of the subcarriers with indices –42 to –33, –31 to –6, +6 to +31, and +33 to +42 shall deviate no more than ± 4 dB from their average energy. The average energy of the constellations in each of the subcarriers with indices –43 to –58 and +43 to +58 shall deviate no more than +4/–6 dB from the average energy of subcarriers with indices –42 to –33, –31 to –6, +6 to +31, and +33 to +42.

The tests for the spectral flatness requirements may be performed with spatial mapping $Q_k = \mathbf{I}$ (see 20.3.11.11.2).

20.3.20.3 Transmit power

The maximum allowable output power is measured in accordance with practices specified by the appropriate regulatory bodies.

20.3.20.4 Transmit center frequency tolerance

The transmitter center frequency tolerance shall be ± 20 ppm maximum for the 5 GHz band and ± 25 ppm maximum for the 2.4 GHz band. The different transmit chain center frequencies (LO) and each transmit chain symbol clock frequency shall all be derived from the same reference oscillator.

20.3.20.5 Packet alignment

If no signal extension is required (see 20.3.2), the receiver shall emit a PHY-CCA.indication(idle) primitive (see 7.3.5.11) at the 4 µs boundary following the reception of the last symbol of the packet. If a signal extension is required, the receiver shall emit a PHY-CCA.indication(idle) primitive a duration of aSignalExtension µs after the 4 µs boundary following the reception of the last symbol of the packet. This situation is illustrated for an HT-greenfield format packet using short GI in Figure 20-21.

If no signal extension is required, the transmitter shall emit a PHY-TXEND.confirm primitive (see 7.3.5.8) at the 4 µs boundary following the trailing boundary of the last symbol of the packet on the air. If a signal extension is required, the transmitter shall emit a PHY-TXEND.confirm primitive (see 7.3.5.8) a duration of aSignalExtension µs after the 4 µs boundary following the trailing boundary of the last symbol of the packet on the air. This situation is illustrated in Figure 20-21.

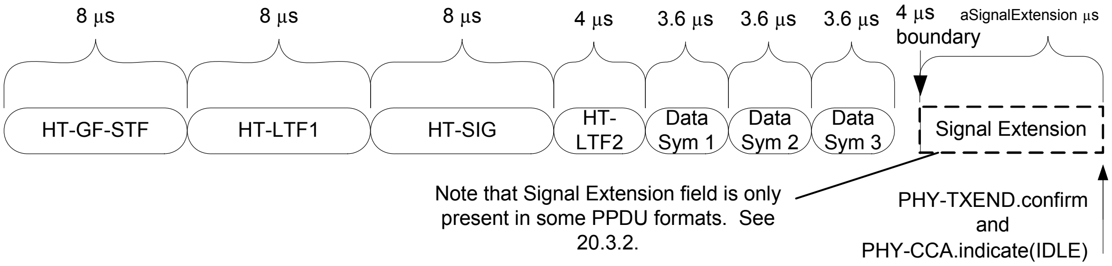


Figure 20-21—Packet alignment example (HT-greenfield format packet with short GI)

20.3.20.6 Symbol clock frequency tolerance

The symbol clock frequency tolerance shall be ± 20 ppm maximum for 5 GHz bands and ± 25 ppm for 2.4 GHz bands. The transmit center frequency and the symbol clock frequency for all transmit antennas shall be derived from the same reference oscillator.

20.3.20.7 Modulation accuracy

20.3.20.7.1 Introduction to modulation accuracy tests

Transmit modulation accuracy specifications are described in 20.3.20.7.2 and 20.3.20.7.3. The test method is described in 20.3.20.7.4.

20.3.20.7.2 Transmit center frequency leakage

The transmitter center frequency leakage shall follow 18.3.9.7.2 for all transmissions in a 20 MHz channel width. For transmissions in a 40 MHz channel width, the center frequency leakage shall not exceed –20 dB relative to overall transmitted power, or, equivalently, 0 dB relative to the average energy of the rest of the subcarriers. For upper or lower 20 MHz transmissions in a 40 MHz channel, the center frequency leakage (center of a 40 MHz channel) shall not exceed –17 dB relative to overall transmitted power, or, equivalently, 0 dB relative to the average energy of the rest of the subcarriers. The transmit center frequency leakage is specified per antenna.

20.3.20.7.3 Transmitter constellation error

The relative constellation frame-averaged RMS error, calculated first by averaging over subcarriers, OFDM frames, and spatial streams, shall not exceed a data-rate-dependent value according to Table 20-22. The number of spatial streams under test shall be equal to the number of utilized transmitting STA antenna (output) ports and also equal to the number of utilized testing instrumentation input ports. In the test, $N_{SS} = N_{STS}$ with EQM MCSs shall be used. Each output port of the transmitting STA shall be connected through a cable to one input port of the testing instrumentation. The same requirement applies both to 20 MHz channels and 40 MHz channels.

Table 20-22—Allowed relative constellation error versus constellation size and coding rate

Modulation	Coding rate	Relative constellation error (dB)
BPSK	1/2	–5
QPSK	1/2	–10

Table 20-22—Allowed relative constellation error versus constellation size and coding rate *(continued)*

Modulation	Coding rate	Relative constellation error (dB)
QPSK	3/4	–13
16-QAM	1/2	–16
16-QAM	3/4	–19
64-QAM	2/3	–22
64-QAM	3/4	–25
64-QAM	5/6	–27

20.3.20.7.4 Transmitter modulation accuracy (EVM) test

The transmit modulation accuracy test shall be performed by instrumentation capable of converting the transmitted signals into a streams of complex samples at 40 Msample/s or more, with sufficient accuracy in terms of I/Q arm amplitude and phase balance, dc offsets, phase noise, and analog-to-digital quantization noise. Each transmit chain is connected directly through a cable to the setup input port. A possible embodiment of such a setup is converting the signals to a low intermediate frequency with a microwave synthesizer, sampling the signal with a digital oscilloscope, and decomposing it digitally into quadrature components. The sampled signal shall be processed in a manner similar to an actual receiver, according to the following steps, or an equivalent procedure:

a) Detect the start of frame.

b) Detect the transition from short sequences to channel estimation sequences, and establish fine timing (with one sample resolution).

c) Estimate the coarse and fine frequency offsets.

d) Derotate the frame according to estimated frequency offset.

e) Estimate the complex channel response coefficients for each of the subcarriers and each of the transmit chains.

f) For each of the data OFDM symbols, transform the symbol into subcarrier received values, estimate the phase from the pilot subcarriers in all spatial streams, derotate the subcarrier values according to estimated phase, group the results from all the receiver chains in each subcarrier to a vector, multiply the vector by a zero-forcing equalization matrix generated from the channel estimated during the channel estimation phase.

g) For each data-carrying subcarrier in each spatial stream, find the closest constellation point and compute the Euclidean distance from it.

h) Compute the average of the RMS of all errors in a frame. It is given by Equation (20-89).

$$Error_{RMS} = \frac{\sum_{i_f=1}^{N_f} \sqrt{\frac{\sum_{i_s=1}^{N_{SYM}} \left[\sum_{i_{ss}=1}^{N_{SS}} \left(\sum_{i_{sc}=1}^{N_{ST}} ((I(i_f, i_s, i_{ss}, i_{sc}) - I_0(i_f, i_s, i_{ss}, i_{sc}))^2 + (Q(i_f, i_s, i_{ss}, i_{sc}) - Q_0(i_f, i_s, i_{ss}, i_{sc}))^2) \right) \right]}{N_{SYM} \times N_{SS} \times N_{ST} \times P_0}}}{N_f} \qquad (20\text{-}89)$$

where
N_f is the number of frames for the measurement

$I_0(i_f, i_s, i_{ss}, i_{sc})$, $Q_0(i_f, i_s, i_{ss}, i_{sc})$ denotes the ideal symbol point in the complex plane in subcarrier i_{sc}, spatial stream i_{ss}, and OFDM symbol i_s of frame i_f

$I(i_f, i_s, i_{ss}, i_{sc})$, $Q(i_f, i_s, i_{ss}, i_{sc})$ denotes the observed symbol point in the complex plane in subcarrier i_{sc}, spatial stream i_{ss}, and OFDM symbol i_s of frame i_f

P_0 is the average power of the constellation

The vector error on a phase plane is shown in Figure 18-16.

The test shall be performed over at least 20 frames (N_f), and the average of the RMS shall be taken. The frames under test shall be at least 16 OFDM symbols long. Random data shall be used for the symbols.

20.3.20.8 Time of Departure accuracy

The Time of Departure accuracy test evaluates TIME_OF_DEPARTURE against aTxPmdTxStartRMS and aTxPmdTxStartRMS against TIME_OF_DEPARTURE_ACCURACY_TEST_THRESH as defined in Annex T with the following test parameters:

— MULTICHANNEL_SAMPLING_RATE is:

$20 \times 10^6\left(1 + \left\lceil \frac{f_H - f_L}{20\text{ MHz}} \right\rceil\right)$ sample/s, for a CH_BANDWIDTH parameter equal to HT_CBW20

$40 \times 10^6\left(1 + \left\lceil \frac{f_H - f_L}{40\text{ MHz}} \right\rceil\right)$ sample/s, for a CH_BANDWIDTH parameter equal to HT_CBW40

where

f_H is the nominal center frequency in Hz of the highest channel in the channel set

f_L is the nominal center frequency in Hz of the lowest channel in the channel set, the channel set is the set of channels upon which frames providing measurements are transmitted, the channel set comprises channels uniformly spaced across $f_H - f_L \geq 50$ MHz

$\lceil x \rceil$ equals the smallest integer equal to or larger than x.

— FIRST_TRANSITION_FIELD is L-STF (for HT-mixed format) or HT-GF-STF (for HT-greenfield format)

— SECOND_TRANSITION_FIELD is L-LTF (for HT-mixed format) or HT-GF-LTF1 (for HT-greenfield format)

— TRAINING_FIELD is L-LTF (for HT-mixed format) or HT-LTF1 (for HT-greenfield format) windowed in a manner which should approximate the windowing described in 18.3.2.5 with T_{TR} = 100 ns.

— TIME_OF_DEPARTURE_ACCURACY_TEST_THRESH is 80 ns (for a CH_BANDWIDTH parameter equal to HT_CBW20) or 80 ns (for a CH_BANDWIDTH parameter equal to HT_CBW40).

NOTE—The indicated windowing applies to the time of departure accuracy test equipment, and not the transmitter or receiver.

20.3.21 HT PMD receiver specification

20.3.21.1 Receiver minimum input sensitivity

The packet error ratio (PER) shall be less than 10% for a PSDU length of 4096 octets with the rate-dependent input levels listed in Table 20-23 or less. The minimum input levels are measured at the antenna connectors and are referenced as the average power per receive antenna. The number of spatial streams under test shall be equal to the number of utilized transmitting STA antenna (output) ports and also equal to the number of utilized device under test input ports. Each output port of the transmitting STA shall be connected through a cable to one input port of the device under test. The test in this subclause and the minimum sensitivity levels specified in Table 20-23 apply only to non-STBC modes, MCSs 0–31, 800 ns GI, and BCC.

Table 20-23—Receiver minimum input level sensitivity

Modulation	Rate (R)	Adjacent channel rejection (dB)	Nonadjacent channel rejection (dB)	Minimum sensitivity (20 MHz channel spacing) (dBm)	Minimum sensitivity (40 MHz channel spacing) (dBm)
BPSK	1/2	16	32	–82	–79
QPSK	1/2	13	29	–79	–76
QPSK	3/4	11	27	–77	–74
16-QAM	1/2	8	24	–74	–71
16-QAM	3/4	4	20	–70	–67
64-QAM	2/3	0	16	–66	–63
64-QAM	3/4	–1	15	–65	–62
64-QAM	5/6	–2	14	–64	–61

20.3.21.2 Adjacent channel rejection

For all transmissions in a 20 MHz channel width, the adjacent channel rejection shall be measured by setting the desired signal's strength 3 dB above the rate-dependent sensitivity specified in Table 20-23 and raising the power of the interfering signal until 10% PER is caused for a PSDU length of 4096 octets. The power difference between the interfering channel and the desired channel is the corresponding adjacent channel rejection. The adjacent channel center frequencies shall be separated by 20 MHz when operating in the 5 GHz band, and the adjacent channel center frequencies shall be separated by 25 MHz when operating in the 2.4 GHz band.

For all transmissions in a 40 MHz channel width, the adjacent channel rejection shall be measured by setting the desired signal's strength 3 dB above the rate-dependent sensitivity specified in Table 20-23 and raising the power of the interfering signal until 10% PER is caused for a PSDU length of 4096 octets. The power difference between the interfering channel and the desired channel is the corresponding adjacent channel rejection. The adjacent channel center frequencies shall be separated by 40 MHz.

The interfering signal in the adjacent channel shall be a conformant OFDM signal, unsynchronized with the signal in the channel under test. For a conforming OFDM PHY, the corresponding rejection shall be no less than specified in Table 20-23. The interference signal shall have a minimum duty cycle of 50%.

The test in this subclause and the adjacent channel rejection levels specified in Table 20-23 apply only to non-STBC modes, MCSs 0–31, 800 ns GI, and BCC.

20.3.21.3 Nonadjacent channel rejection

For all transmissions in a 20 MHz channel width in the 5 GHz band, the nonadjacent channel rejection shall be measured by setting the desired signal's strength 3 dB above the rate-dependent sensitivity specified in Table 20-23 and raising the power of the interfering signal until a 10% PER occurs for a PSDU length of 4096 octets. The power difference between the interfering channel and the desired channel is the corresponding nonadjacent channel rejection. The nonadjacent channel center frequencies shall be separated by 40 MHz or more.

For all transmissions in a 40 MHz channel width in the 5 GHz band, the nonadjacent channel rejection shall be measured by setting the desired signal's strength 3 dB above the rate-dependent sensitivity specified in Table 20-23 and raising the power of the interfering signal until a 10% PER occurs for a PSDU length of 4096 octets. The power difference between the interfering channel and the desired channel is the corresponding nonadjacent channel rejection. The nonadjacent channel center frequencies shall be separated by 80 MHz or more.

The interfering signal in the nonadjacent channel shall be a conformant OFDM signal, unsynchronized with the signal in the channel under test. For a conforming OFDM PHY, the corresponding rejection shall be no less than specified in Table 20-23. The interference signal shall have a minimum duty cycle of 50%. The nonadjacent channel rejection for transmissions in a 20 MHz or 40 MHz channel width is applicable only to 5 GHz band.

The test in this subclause and the nonadjacent channel rejection level specified in Table 20-23 apply only to non-STBC modes, MCSs 0–31, 800 ns GI, and BCC.

20.3.21.4 Receiver maximum input level

The receiver shall provide a maximum PER of 10% at a PSDU length of 4096 octets, for a maximum input level of –30 dBm in the 5 GHz band and –20 dBm in the 2.4 GHz band, measured at each antenna for any baseband modulation.

20.3.21.5 CCA sensitivity

20.3.21.5.1 CCA sensitivity for non-HT PPDUs

CCA sensitivity requirements for non-HT PPDUs in the primary channel are described in 18.3.10.6 and 19.4.7.

20.3.21.5.2 CCA sensitivity in 20 MHz

For an HT STA with the operating channel width equal to 20 MHz, the start of a valid 20 MHz HT signal at a receive level equal to or greater than the minimum modulation and coding rate sensitivity of –82 dBm shall cause the PHY to set PHY-CCA.indicate(BUSY) with a probability > 90% within 4 µs. The receiver shall hold the CCA signal busy for any signal 20 dB or more above the minimum modulation and coding rate sensitivity (–82 + 20 = –62 dBm) in the 20 MHz channel.

A receiver that does not support the reception of HT-GF format PPDUs shall hold the CCA signal busy (PHY_CCA.indicate(BUSY)) for any valid HT-GF signal in the 20 MHz channel at a receive level equal to or greater than –72 dBm.

20.3.21.5.3 CCA sensitivity in 40 MHz

This subclause describes the CCA sensitivity requirements for an HT STA with the operating channel width equal to 40 MHz.

The receiver of a 20/40 MHz STA with the operating channel width equal to 40 MHz shall provide CCA on both the primary and secondary channels.

When the secondary channel is idle, the start of a valid 20 MHz HT signal in the primary channel at a receive level equal to or greater than the minimum modulation and coding rate sensitivity of –82 dBm shall cause the PHY to set PHY-CCA.indicate(BUSY, {primary}) with a probability > 90% within 4 μs. The start of a valid 40 MHz HT signal that occupies both the primary and secondary channels at a receive level equal to or greater than the minimum modulation and coding rate sensitivity of –79 dBm shall cause the PHY to set PHY-CCA.indicate(BUSY, {primary, secondary}) for both the primary and secondary channels with a probability per channel > 90% within 4 μs.

A receiver that does not support the reception of HT-GF format PPDUs shall hold the CCA signal busy (PHY_CCA.indicate(BUSY, {primary})) for any valid HT-GF signal in the primary channel at a receive level equal to or greater than –72 dBm when the secondary channel is idle. A receiver that does not support the reception of HT-GF format PPDUs shall hold both the 20 MHz primary channel CCA and the 20 MHz secondary channel CCA busy (PHY_CCA.indicate(BUSY, {primary, secondary})) for any valid 40 MHz HT-GF signal in both the primary and secondary channels at a receive level equal to or greater than –69 dBm.

The receiver shall hold the 20 MHz primary channel CCA signal busy for any signal at or above –62 dBm in the 20 MHz primary channel. This level is 20 dB above the minimum modulation and coding rate sensitivity for a 20 MHz PPDU. When the primary channel is idle, the receiver shall hold the 20 MHz secondary channel CCA signal busy for any signal at or above –62 dBm in the 20 MHz secondary channel. The receiver shall hold both the 20 MHz primary channel CCA and the 20 MHz secondary channel CCA busy for any signal present in both the primary and secondary channels that is at or above –62 dBm in the primary channel and at or above –62 dBm in the secondary channel.

20.3.21.6 Received channel power indicator (RCPI) measurement

The RCPI is a measure of the received RF power in the selected channel. This parameter shall be a measure by the PHY of the received RF power in the channel measured over the data portion of the received frame. The received power shall be the average of the power in all active receive chains. RCPI shall be a monotonically increasing, logarithmic function of the received power level defined in dBm. The allowed values for the Received Channel Power Indicator (RCPI) parameter shall be an 8 bit value in the range from 0 to 220, with indicated values rounded to the nearest 0.5 dB as follows:

- 0: Power < –110 dBm
- 1: Power = –109.5 dBm
- 2: Power = –109.0 dBm
- And so on up to
- 220: Power > 0 dBm
- 221–254: reserved
- 255: Measurement not available

where

$$\text{RCPI} = \text{Int}\{(\text{Power in dBm} + 110) \times 2\} \text{ for } 0 \text{ dBm} > \text{Power} > -110 \text{ dBm} \tag{20-90}$$

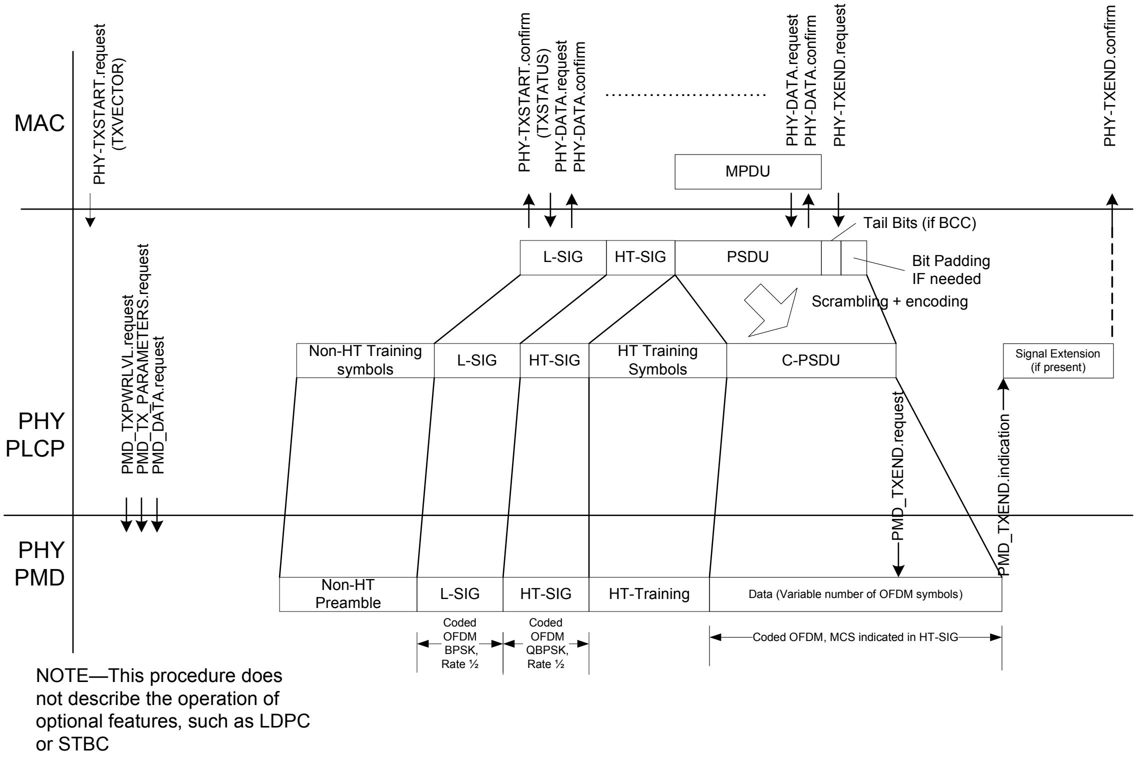
RCPI shall equal the received RF power within an accuracy of ± 5 dB (95% confidence interval) within the specified dynamic range of the receiver. The received RF power shall be determined assuming a receiver noise equivalent bandwidth equal to the channel width multiplied by 1.1.

20.3.21.7 Reduced interframe space (RIFS)

The receiver shall be able to decode a packet that was transmitted by a STA with a RIFS separation from the previous packet.

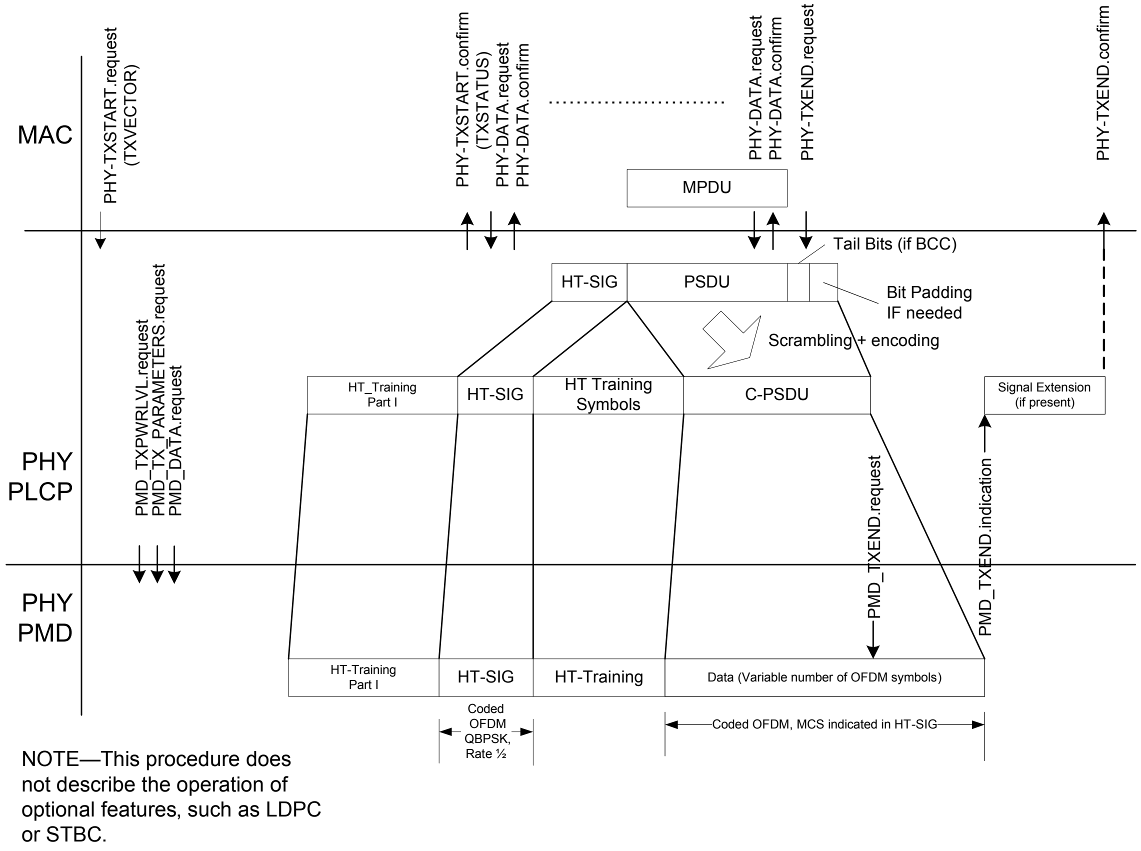
20.3.22 PLCP transmit procedure

There are three options for the transmit PLCP procedure. The first two options, for which typical transmit procedures are shown in Figure 20-22 and Figure 20-23, are selected if the FORMAT field of the PHY-TXSTART.request(TXVECTOR) primitive is equal to HT_MF or HT_GF, respectively. These transmit procedures do not describe the operation of optional features, such as LDPC or STBC. The third option is to follow the transmit procedure in Clause 18 or Clause 19 if the FORMAT field is equal to NON_HT. Additionally, if the FORMAT field is equal to NON_HT, CH_BANDWIDTH indicates NON_HT_CBW20, and NON_HT_MODULATION indicates OFDM, follow the transmit procedure in Clause 18. If the FORMAT field is equal to NON_HT, CH_BANDWIDTH indicates NON_HT_CBW20, and NON_HT_MODULATION indicates other than OFDM, follow the transmit procedure in Clause 19. And furthermore, if the FORMAT field is equal to NON_HT and CH_BANDWIDTH indicates NON_HT_CBW40, follow the transmit procedure in Clause 18, except that the signal in Clause 18 is generated simultaneously on each of the upper and lower 20 MHz channels that constitute the 40 MHz channel as defined in 20.3.8 and 20.3.11.12. In all these options, in order to transmit data, the PHY-TXSTART.request primitive shall be enabled so that the PHY entity shall be in the transmit state. Further, the PHY shall be set to operate at the appropriate frequency through station management via the PLME, as specified in 20.4. Other transmit parameters, such as MCS coding types and transmit power, are set via the PHY-SAP with the PHY-TXSTART.request(TXVECTOR) primitive, as described in 20.2.2.



NOTE—This procedure does not describe the operation of optional features, such as LDPC or STBC

Figure 20-22—PLCP transmit procedure (HT-mixed format PPDU)



NOTE—This procedure does not describe the operation of optional features, such as LDPC or STBC.

Figure 20-23—PLCP transmit procedure (HT-greenfield format PPDU)

A clear channel shall be indicated by a PHY-CCA.indication(IDLE) primitive. Note that under some circumstances, the MAC uses the latest value of the PHY-CCA.indication primitive before issuing the PHY-TXSTART.request primitive. Transmission of the PPDU shall be initiated after receiving the PHY-TXSTART.request(TXVECTOR) primitive. The TXVECTOR elements for the PHY-TXSTART.request primitive are specified in Table 20-1.

The PLCP shall issue the parameters in the following PMD primitives to configure the PHY:

— PMD_TXPWRLVL
— PMD_TX_PARAMETERS

The PLCP shall then issue a PMD_TXSTART.request primitive, and transmission of the PLCP preamble may start if TIME_OF_DEPARTURE_REQUESTED is false, and shall start immediately if TIME_OF_DEPARTURE_REQUESTED is true, based on the parameters passed in the PHY-TXSTART.request primitive. If dot11MgmtOptionTODImplemented and dot11MgmtOptionTODActivated are true or if dot11MgmtOptionTimingMsmtActivated is true and the TXVECTOR parameter TIME_OF_DEPARTURE_REQUESTED is true, then the PLCP shall issue a PHY_TXSTART.confirm(TXSTATUS) primitive to the MAC, forwarding the TIME_OF_DEPARTURE corresponding to the time when the first frame energy is sent by the transmitting port and the TIME_OF_DEPARTURE_ClockRate parameter within the TXSTATUS vector. If dot11MgmtOptionTimingMsmtActivated is true, then the PLCP shall forward the value of TX_START_OF_FRAME_OFFSET in TXSTATUS vector. The data shall then be exchanged between the MAC and the PHY through a series of PHY-DATA.request(DATA) primitives issued by the MAC and PHY-DATA.confirm primitives issued by the PHY. Once PLCP preamble transmission is started, the PHY entity shall immediately initiate data scrambling and data encoding. The encoding method shall be based on the FEC_CODING, CH_BANDWIDTH, and MCS parameter of the TXVECTOR. A modulation rate change, if any, shall be initiated starting with the SERVICE field data, as described in 20.3.2.

The PHY proceeds with PSDU transmission through a series of data octet transfers from the MAC. The SERVICE field and PSDU are encoded by the encoder selected by the FEC_CODING, CH_BANDWIDTH, and MCS parameters of the TXVECTOR as described in 20.3.3. At the PMD layer, the data octets are sent in bit 0–7 order and presented to the PHY through PMD_DATA.request primitives. Transmission can be prematurely terminated by the MAC through the primitive PHY-TXEND.request primitive. PHY-TXSTART shall be disabled by receiving a PHY-TXEND.request primitive. Normal termination occurs after the transmission of the final bit of the last PSDU octet, according to the number supplied in the LENGTH field.

The packet transmission shall be completed, and the PHY entity shall enter the receive state (i.e., PHY-TXSTART shall be disabled). Each PHY-TXEND.request primitive is acknowledged with a PHY-TXEND.confirm primitive from the PHY. If the length of the coded PSDU (C-PSDU) is not an integral multiple of the OFDM symbol length, bits shall be stuffed to make the C-PSDU length an integral multiple of the OFDM symbol length.

In the PMD, the GI or short GI shall be inserted in every OFDM symbol as a countermeasure against delay spread.

In some PPDU formats (as defined in 20.3.2), a signal extension is present. When no signal extension is present, the PHY-TXEND.confirm primitive is generated at the end of last symbol of the PPDU. When a signal extension is present, the PHY-TXEND.confirm primitive is generated at the end of the signal extension.

A typical state machine implementation of the transmit PLCP is provided in Figure 20-24. Requests (.request) and confirmations (.confirm) are issued once per state as shown. This state machine does not describe the operation of optional features, such as LDPC or STBC.

20.3.23 PLCP receive procedure

Typical PLCP receive procedures are shown in Figure 20-25 and Figure 20-26. The receive procedures correspond to HT-mixed format and HT-greenfield format, respectively. A typical state machine implementation of the receive PLCP is given in Figure 20-27. These receive procedures and state machine do not describe the operation of optional features, such as LDPC or STBC. If the detected format indicates a non-HT PPDU format, refer to the receive procedure and state machine in Clause 18 or Clause 19. Further, through station management (via the PLME), the PHY is set to the appropriate frequency, as specified in 20.4. Other receive parameters, such as RSSI and indicated DATARATE, may be accessed via the PHY-SAP.

Upon receiving the transmitted PLCP preamble, the PMD_RSSI.indication primitive shall report a receive signal strength to the PLCP. This PHY indicates activity to the MAC via the PHY-CCA.indication primitive. A PHY-CCA.indication(BUSY, channel-list) primitive shall also be issued as an initial indication of reception of a signal. The channel-list parameter of the PHY-CCA.indication primitive is determined as follows:

- It is absent when the operating channel width is 20 MHz.
- It is set to {primary} when the operating channel width is 40 MHz and the signal is present only in the primary channel.
- It is set to {secondary} when the operating channel width is 40 MHz and the signal is present only in the secondary channel.
- It is set to {primary, secondary} when the operating channel width is 40 MHz and the signal is present in both the primary and secondary channels.

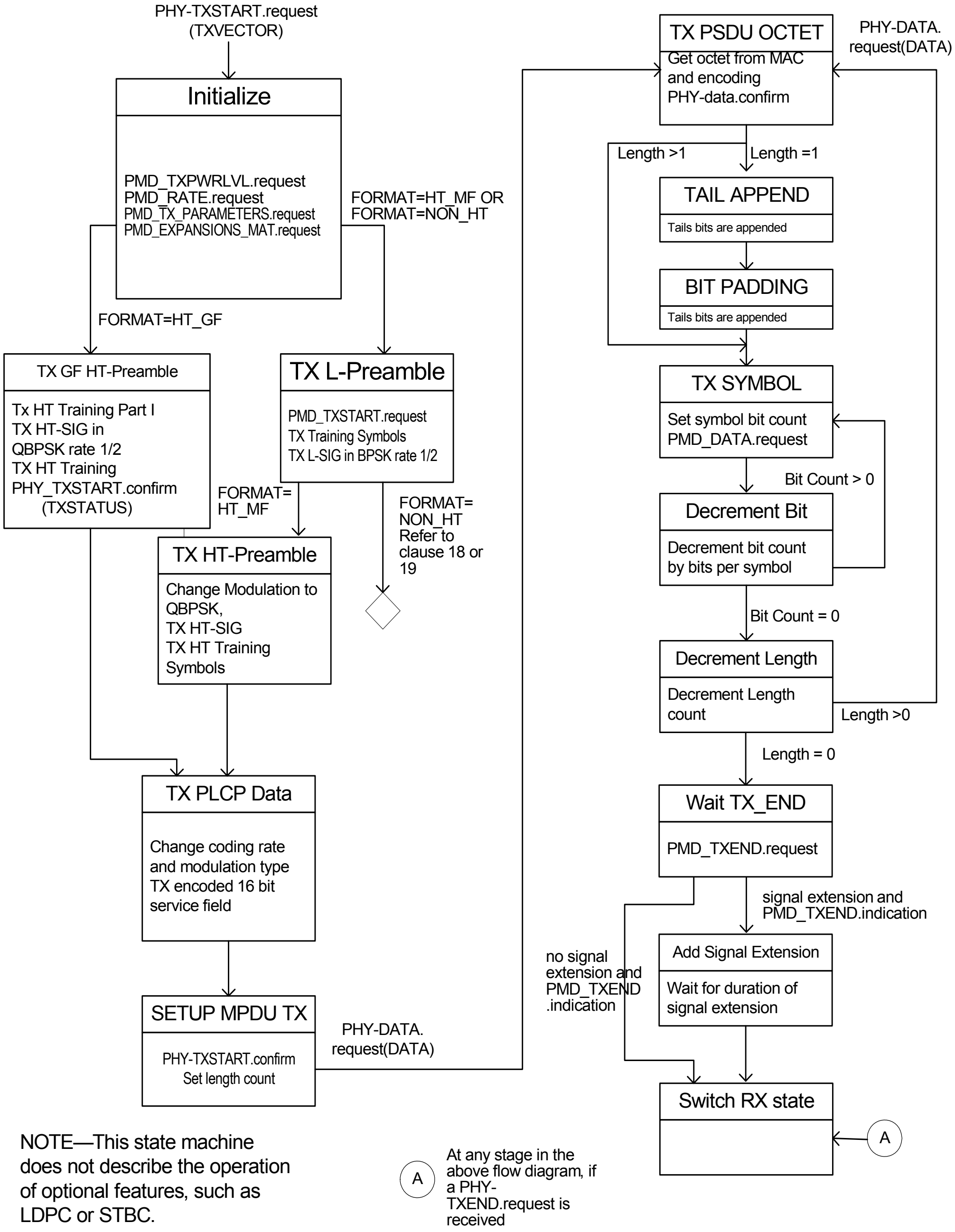
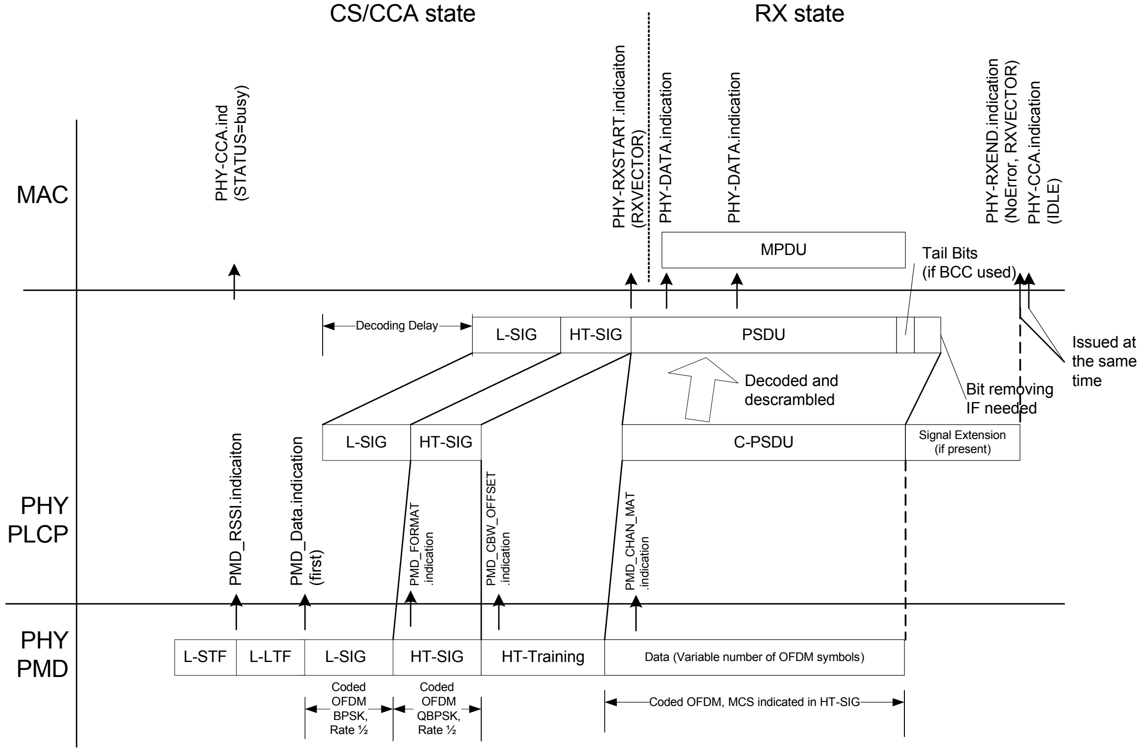

PHY-TXSTART.request (TXVECTOR)
Initialize
PMD_TXPWRLVL.request
PMD_RATE.request
PMD_TX_PARAMETERS.request
PMD_EXPANSIONS_MAT.request
FORMAT=HT_MF OR FORMAT=NON_HT
FORMAT=HT_GF
TX GF HT-Preamble
Tx HT Training Part I
TX HT-SIG in QBPSK rate 1/2
TX HT Training
PHY_TXSTART.confirm (TXSTATUS)
TX L-Preamble
PMD_TXSTART.request
TX Training Symbols
TX L-SIG in BPSK rate 1/2
FORMAT= HT_MF
FORMAT= NON_HT Refer to clause 18 or 19
TX HT-Preamble
Change Modulation to QBPSK,
TX HT-SIG
TX HT Training Symbols
TX PLCP Data
Change coding rate and modulation type
TX encoded 16 bit service field
SETUP MPDU TX
PHY-TXSTART.confirm
Set length count
PHY-DATA. request(DATA)
TX PSDU OCTET
Get octet from MAC and encoding
PHY-data.confirm
PHY-DATA. request(DATA)
Length >1
Length =1
TAIL APPEND
Tails bits are appended
BIT PADDING
Tails bits are appended
TX SYMBOL
Set symbol bit count
PMD_DATA.request
Bit Count > 0
Decrement Bit
Decrement bit count by bits per symbol
Bit Count = 0
Decrement Length
Decrement Length count
Length >0
Length = 0
Wait TX_END
PMD_TXEND.request
signal extension and PMD_TXEND.indication
no signal extension and PMD_TXEND .indication
Add Signal Extension
Wait for duration of signal extension
Switch RX state
A
NOTE—This state machine does not describe the operation of optional features, such as LDPC or STBC.
A
At any stage in the above flow diagram, if a PHY-TXEND.request is received

Figure 20-24—PLCP transmit state machine



NOTE—This procedure does not describe the operation of optional features, such as LDPC or STBC.

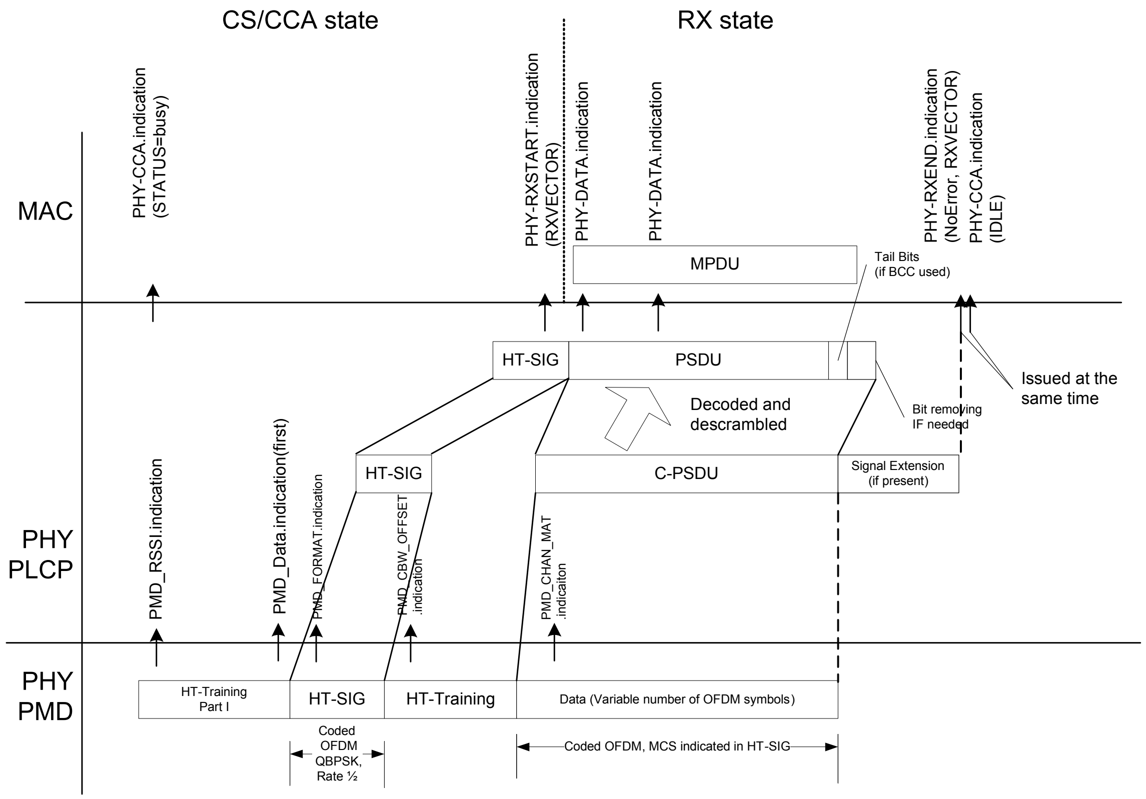
Figure 20-25—PLCP receive procedure for HT-mixed format PLCP format

The PMD primitive PMD_RSSI is issued to update the RSSI and parameter reported to the MAC.

After the PHY-CCA.indication(BUSY, channel-list) primitive is issued, the PHY entity shall begin receiving the training symbols and searching for SIGNAL and HT-SIG in order to set the length of the data stream, the demodulation type, code type, and the decoding rate. If signal loss occurs before validating L-SIG and/or HT-SIG, the HT PHY shall not generate a PHY-CCA.indication(IDLE) primitive until the received level drops below the CCA sensitivity level (for a missed preamble) specified in 20.3.21.5. If the check of the HT-SIG CRC is not valid, a PHY-RXSTART.indication primitive is not issued. The PHY shall indicate the error condition using a PHY-RXEND.indication(FormatViolation) primitive. The HT PHY shall not generate a PHY-CCA.indication(IDLE) primitive until the received level drops below the CCA sensitivity level (for a missed preamble) specified in 20.3.21.5.

If the PLCP preamble reception is successful and a valid HT-SIG CRC is indicated:

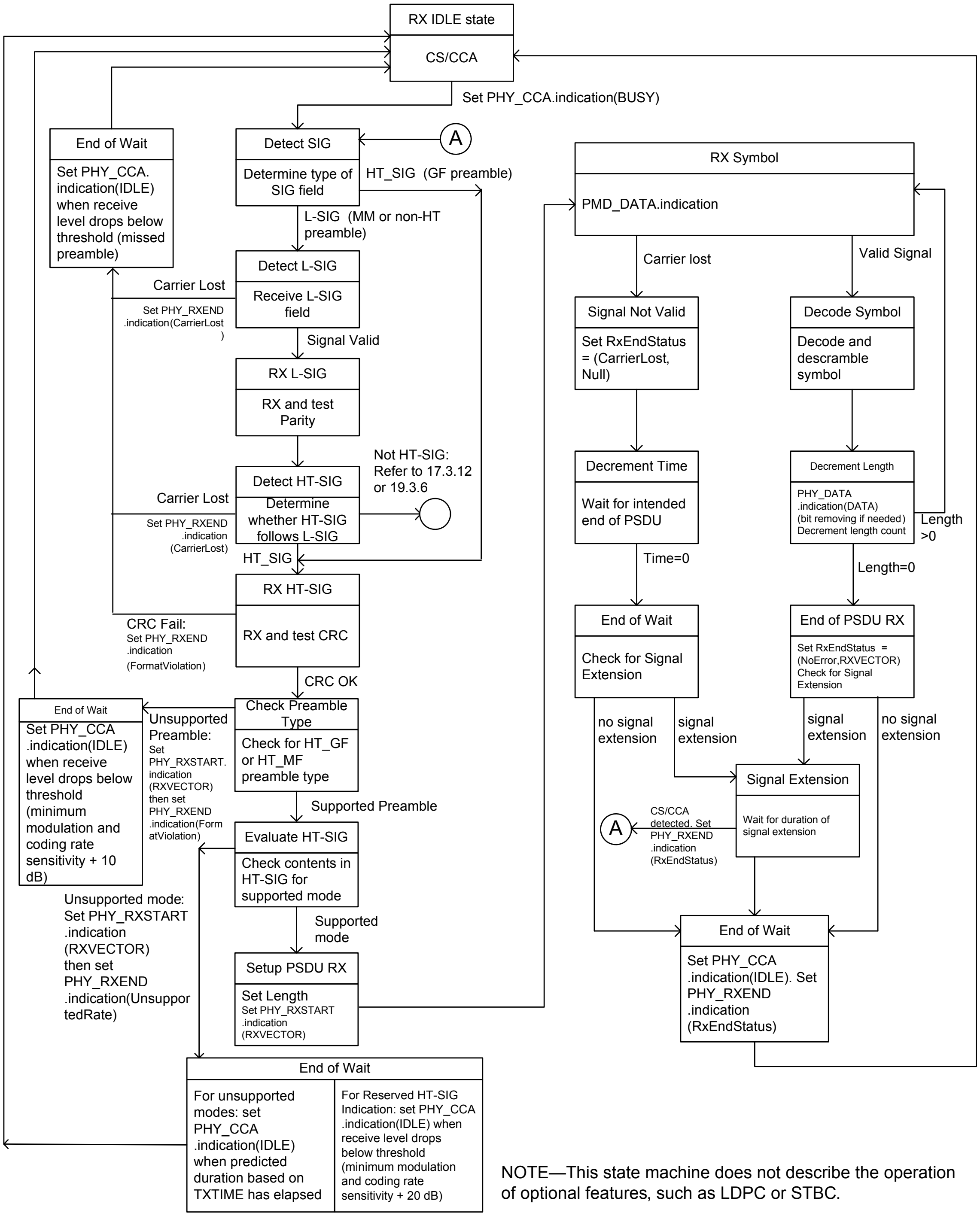
— Upon reception of an HT-mixed format preamble, the HT PHY shall not generate a PHY-CCA.indication(IDLE) primitive for the predicted duration of the transmitted frame, as defined by TXTIME in 20.4.3, for all supported and unsupported modes except Reserved HT-SIG Indication. Reserved HT-SIG Indication is defined in the fourth list item below.



NOTE—This procedure does not describe the operation of optional features, such as LDPC or STBC.

Figure 20-26—PLCP receive procedure for HT-greenfield format PLCP

- Upon reception of a GF preamble by an HT STA that does not support GF, the HT PHY shall not generate a PHY-CCA.indication(IDLE) primitive until either the predicted duration of the packet from the contents of the HT-SIG field, as defined by TXTIME in 20.4.3, except Reserved HT-SIG Indication, elapses or until the received level drops below the receiver minimum sensitivity level of BPSK, R=1/2 in Table 20-23 + 10 dB (–72 dBm for 20 MHz, –69 dBm for 40 MHz). Reserved HT-SIG Indication is defined in the fourth list item below.
- Upon reception of a GF preamble by an HT STA that supports GF, the HT PHY shall not generate a PHY-CCA.indication(IDLE) primitive for the predicted duration of the transmitted frame, as defined by TXTIME in 20.4.3, for all supported and unsupported modes except Reserved HT-SIG Indication. Reserved HT-SIG Indication is defined in the fourth list item below.
- If the HT-SIG indicates a Reserved HT-SIG Indication, the HT PHY shall not generate a PHY-CCA.indication(IDLE) primitive until the received level drops below the CCA sensitivity level (minimum modulation and coding rate sensitivity + 20 dB) specified in 20.3.21.5. Reserved HT-SIG Indication is defined as an HT-SIG with MCS field in the range 77–127 or Reserved field = 0 or STBC field = 3 and any other HT-SIG field bit combinations that do not correspond to modes of PHY operation defined in Clause 20.


RX IDLE state
CS/CCA
Set PHY_CCA.indication(BUSY)
End of Wait
Set PHY_CCA. indication(IDLE) when receive level drops below threshold (missed preamble)
Detect SIG
A
Determine type of SIG field
HT_SIG (GF preamble)
L-SIG (MM or non-HT preamble)
Detect L-SIG
Receive L-SIG field
Carrier Lost
Set PHY_RXEND .indication(CarrierLost)
Signal Valid
RX L-SIG
RX and test Parity
Not HT-SIG: Refer to 17.3.12 or 19.3.6
Detect HT-SIG
Determine whether HT-SIG follows L-SIG
Carrier Lost
Set PHY_RXEND .indication (CarrierLost)
HT_SIG
RX HT-SIG
RX and test CRC
CRC Fail: Set PHY_RXEND .indication (FormatViolation)
CRC OK
End of Wait
Set PHY_CCA .indication(IDLE) when receive level drops below threshold (minimum modulation and coding rate sensitivity + 10 dB)
Unsupported Preamble: Set PHY_RXSTART. indication (RXVECTOR) then set PHY_RXEND .indication(Form atViolation)
Check Preamble Type
Check for HT_GF or HT_MF preamble type
Supported Preamble
Evaluate HT-SIG
Check contents in HT-SIG for supported mode
Unsupported mode: Set PHY_RXSTART .indication (RXVECTOR) then set PHY_RXEND .indication(Unsuppor tedRate)
Supported mode
Setup PSDU RX
Set Length
Set PHY_RXSTART .indication (RXVECTOR)
End of Wait
For unsupported modes: set PHY_CCA .indication(IDLE) when predicted duration based on TXTIME has elapsed
For Reserved HT-SIG Indication: set PHY_CCA .indication(IDLE) when receive level drops below threshold (minimum modulation and coding rate sensitivity + 20 dB)
RX Symbol
PMD_DATA.indication
Carrier lost
Valid Signal
Signal Not Valid
Set RxEndStatus = (CarrierLost, Null)
Decode Symbol
Decode and descramble symbol
Decrement Time
Wait for intended end of PSDU
Decrement Length
PHY_DATA .indication(DATA) (bit removing if needed) Decrement length count
Length >0
Time=0
Length=0
End of Wait
Check for Signal Extension
End of PSDU RX
Set RxEndStatus = (NoError,RXVECTOR) Check for Signal Extension
no signal extension
signal extension
signal extension
no signal extension
Signal Extension
Wait for duration of signal extension
A
CS/CCA detected. Set PHY_RXEND .indication (RxEndStatus)
End of Wait
Set PHY_CCA .indication(IDLE). Set PHY_RXEND .indication (RxEndStatus)

NOTE—This state machine does not describe the operation of optional features, such as LDPC or STBC.

Figure 20-27—PLCP receive state machine

Subsequent to an indication of a valid HT-SIG CRC, a PHY-RXSTART.indication(RXVECTOR) primitive shall be issued. If dot11MgmtOptionTimingMsmtActivated is true, the PLCP shall do the following:

— Complete receiving the PLCP header and verify the validity of the PLCP Header.
— If the PLCP header reception is successful (and the SIGNAL field is completely recognizable and supported), a PHY-RXSTART.indication(RXVECTOR) shall be issued and RX_START_OF_FRAME_OFFSET parameter within the RXVECTOR shall be forwarded (see 20.2.2).

NOTE—The RX_START_OF_FRAME_OFFSET value is used as described in 6.3.57 in order to estimate when the start of the preamble for the incoming frame was detected on the medium at the receive antenna port.

The RXVECTOR associated with this primitive includes the parameters specified in Table 20-1. Upon reception of a GF preamble by an HT STA that does not support GF, the FORMAT field of RXVECTOR is equal to HT_GF, the remaining fields may be empty, and the PHY shall indicate the error condition using a PHY-RXEND.indication(FormatViolation) primitive. If the HT-SIG indicates an unsupported mode or Reserved HT-SIG Indication, the PHY shall indicate the error condition using a PHY-RXEND.indication(UnsupportedRate) primitive.

Following training and SIGNAL fields, the coded PSDU (C-PSDU) (which comprises the coded PLCP SERVICE field and scrambled and coded PSDU) shall be received. If signal loss occurs during reception prior to completion of the PSDU reception, the error condition shall be reported to the MAC using a PHY-RXEND.indication(CarrierLost) primitive. After waiting for the intended end of the PSDU, if no signal extension is present (as defined in 20.3.2), the PHY shall generate a PHY-CCA.indication(IDLE) primitive and return to RX IDLE state. Otherwise, the receiver waits for the duration of the signal extension before returning to the RX IDLE state.

The received PSDU bits are assembled into octets, decoded, and presented to the MAC using a series of PHY-DATA.indication(DATA) primitive exchanges. The number of PSDU octets is indicated in the HT Length field of the HT-SIG. The PHY shall proceed with PSDU reception. After the reception of the final bit of the last PSDU octet and possible tail and padding bits, the receiver shall be returned to the RX IDLE state if no signal extension is present (as defined in 20.3.2), as shown in Figure 20-27. Otherwise, the receiver waits for the duration of the signal extension before returning to the RX IDLE state. A PHY-RXEND.indication(NoError) primitive shall be issued on entry to the RX IDLE state.

While in the Signal Extension state, if the receiver detects a CS/CCA event, it issues an RXEND.indication primitive (with the RXERROR parameter set to NoError or CarrierLost, depending on whether a carrier lost event occurred during the reception of the PPDU), leaves the Signal Extension state, and enters the Detect SIG state. This sequence occurs when signal-extended PPDUs are transmitted while separated by a RIFS.

If the binary convolutional code is used, any data received after the indicated data length are considered pad bits (to fill out an OFDM symbol) and should be discarded.

20.4 HT PLME

20.4.1 PLME_SAP sublayer management primitives

Table 20-24 lists the MIB attributes that may be accessed by the PHY entities and the intralayer of higher level LMEs. These attributes are accessed via the PLME-GET, PLME-SET, PLME-RESET, and PLME-CHARACTERISTICS primitives defined in 10.4.

20.4.2 PHY MIB

HT PHY MIB attributes are defined in Annex C with specific values defined in Table 20-24. The “Operational semantics” column in Table 20-24 contains two types: static and dynamic.

- Static MIB attributes are fixed and cannot be modified for a given PHY implementation.
- Dynamic MIB attributes are interpreted according to the MAX-ACCESS field of the MIB attribute. When MAX-ACCESS is equal to read-only, the MIB attribute value may be updated by the PLME and read from the MIB attribute by management entities. When MAX-ACCESS is equal to read-write, the MIB attribute may be read and written by management entities.

Table 20-24—HT PHY MIB attributes

Managed object	**Default value/range**	**Operational semantics**
dot11PHYOperationTable		
dot11PHYType	HT (X'07')	Static
dot11CurrentRegDomain	Implementation dependent	Dynamic
dot11PHYAntennaTable		
dot11CurrentTxAntenna	Implementation dependent	Dynamic
dot11DiversitySupportImplemented	Implementation dependent	Static
dot11CurrentRxAntenna	Implementation dependent	Dynamic
dot11AntennaSelectionOptionImplemented	False/Boolean	Static
dot11TransmitExplicitCSIFeedbackASOptionImplemented	False/Boolean	Static
dot11TransmitIndicesFeedbackASOptionImplemented	False/Boolean	Static
dot11ExplicitCSIFeedbackASOptionImplemented	False/Boolean	Static
dot11TransmitIndicesComputationASOptionImplemented	False/Boolean	Static
dot11ReceiveAntennaSelectionOptionImplemented	False/Boolean	Static
dot11TransmitSoundingPPDUOptionImplemented	False/Boolean	Static
dot11PHYTxPowerTable		
dot11NumberSupportedPowerLevelsImplemented	Implementation dependent	Static
dot11TxPowerLevel1	Implementation dependent	Static
dot11TxPowerLevel2	Implementation dependent	Static
dot11TxPowerLevel3	Implementation dependent	Static
dot11TxPowerLevel4	Implementation dependent	Static
dot11TxPowerLevel5	Implementation dependent	Static
dot11TxPowerLevel6	Implementation dependent	Static
dot11TxPowerLevel7	Implementation dependent	Static
dot11TxPowerLevel8	Implementation dependent	Static
dot11CurrentTxPowerLevel	Implementation dependent	Dynamic

Table 20-24—HT PHY MIB attributes *(continued)*

Managed object	Default value/range	Operational semantics
dot11PhyDSSSTable		
dot11CurrentChannel	Implementation dependent	Dynamic
dot11RegDomainsSupportedTable		
dot11RegDomainsImplementedValue	Implementation dependent	Static
dot11FrequencyBandsSupported	Implementation dependent	Static
dot11PHYAntennasListTable		
dot11SupportedTxAntenna	Implementation dependent	Dynamic
dot11SupportedRxAntenna	Implementation dependent	Static
dot11DiversitySelectionRx	Implementation dependent	Dynamic
dot11SupportedDataRatesTxTable		
dot11SupportedDataratesTxValue	X'02' = 1 Mb/s (2.4) X'04' = 2 Mb/s (2.4) X'0B' = 5.5 Mb/s (2.4) X'16' = 11 Mb/s (2.4) X'0C' = 6 Mb/s X'12' = 9 Mb/s X'18' = 12 Mb/s X'24' = 18 Mb/s X'2C = 22 Mb/s X'30' = 24 Mb/s X'42 = 33 Mb/s X'48' = 36 Mb/s X'60' = 48 Mb/s X'6C' = 54 Mb/s	Static
dot11SupportedDataRatesRxTable		
dot11SupportedDataratesRxValue	X'02' = 1 Mb/s (2.4) X'04' = 2 Mb/s (2.4) X'0B' = 5.5 Mb/s (2.4) X'16' = 11 Mb/s (2.4) X'0C' = 6 Mb/s X'12' = 9 Mb/s X'18' = 12 Mb/s X'24' = 18 Mb/s X'2C = 22 Mb/s X'30' = 24 Mb/s X'42 = 33 Mb/s X'48' = 36 Mb/s X'60' = 48 Mb/s X'6C' = 54 Mb/s	Static
dot11HRDSSSPHYTable		
dot11ShortPreambleOptionImplemented	True	Static
dot11PBCCOptionImplemented	Implementation dependent	Static
dot11ChannelAgilityPresent	False/Boolean	Static
dot11ChannelAgilityActivated	False/Boolean	Static

Table 20-24—HT PHY MIB attributes *(continued)*

Managed object	Default value/range	Operational semantics
dot11PHYOFDMTable		
dot11CurrentFrequency	Implementation dependent	Dynamic
dot11TIThreshold	Implementation dependent	Dynamic
dot11ChannelStartingFactor	Implementation dependent	Dynamic
dot11PHYERPTable		
dot11ERPPBCCOptionImplemented	Implementation dependent	Static
dot11DSSSOFDMOptionImplemented	Implementation dependent	Static
dot11DSSSOFDMOptionActivated	Implementation dependent	Dynamic
dot11ShortSlotTimeOptionImplemented	Implementation dependent	Static
dot11ShortSlotTimeOptionActivated	Implementation dependent	Dynamic
dot11PHYHTTable		
dot11FortyMHzOperationImplemented	False/Boolean	Static
dot11FortyMHzOperationActivated	False/Boolean	Dynamic
dot11CurrentPrimaryChannel	Implementation dependent	Dynamic
dot11CurrentSecondaryChannel	Implementation dependent	Dynamic
dot11NumberOfSpatialStreamsImplemented	Implementation dependent	Static
dot11NumberOfSpatialStreamsActivated	Implementation dependent	Dynamic
dot11HTGreenfieldOptionImplemented	False/Boolean	Static
dot11HTGreenfieldOptionActivated	False/Boolean	Dynamic
dot11ShortGIOptionInTwentyImplemented	False/Boolean	Static
dot11ShortGIOptionInTwentyActivated	False/Boolean	Dynamic
dot11ShortGIOptionInFortyImplemented	False/Boolean	Static
dot11ShortGIOptionInFortyActivated	False/Boolean	Dynamic
dot11LDPCCodingOptionImplemented	False/Boolean	Static
dot11LDPCCodingOptionActivated	False/Boolean	Dynamic
dot11TxSTBCOptionImplemented	False/Boolean	Static
dot11TxSTBCOptionActivated	False/Boolean	Dynamic
dot11RxSTBCOptionImplemented	False/Boolean	Static
dot11RxSTBCOptionActivated	False/Boolean	Dynamic
dot11BeamFormingOptionImplemented	False/Boolean	Static
dot11BeamFormingOptionActivated	False/Boolean	Dynamic

Table 20-24—HT PHY MIB attributes *(continued)*

Managed object	Default value/range	Operational semantics
dot11HTSupportedMCSTxTable		
dot11SupportedMCSTxValue	MCS 0–76 for 20 MHz; MCS 0–76 for 40 MHz (MCS 0–7 for 20 MHz mandatory at non-AP STA; MCS 0–15 for 20 MHz mandatory at AP)	Static
dot11HTSupportedMCSRxTable		
dot11SupportedMCSRxValue	MCS 0–76 for 20 MHz; MCS 0–76 for 40 MHz (MCS 0–7 for 20 MHz mandatory at non-AP STA; MCS 0–15 for 20 MHz mandatory at AP)	Static
dot11TransmitBeamformingConfigTable		
dot11ReceiveStaggerSoundingOptionImplemented	False/Boolean	Static
dot11TransmitStaggerSoundingOptionImplemented	False/Boolean	Static
dot11ReceiveNDPOptionImplemented	False/Boolean	Static
dot11TransmitNDPOptionImplemented	False/Boolean	Static
dot11ImplicitTransmitBeamformingOptionImplemented	False/Boolean	Static
dot11CalibrationOptionImplemented	Implementation dependent	Static
dot11ExplicitCSITransmitBeamformingOptionImplemented	False/Boolean	Static
dot11ExplicitNonCompressedBeamformingMatrixOption-Implemented	False/Boolean	Static
dot11ExplicitTransmitBeamformingCSIFeedbackOption-Implemented	Implementation dependent	Static
dot11ExplicitNoncompressedBeamformingFeedbackOption-Implemented	Implementation dependent	Static
dot11ExplicitCompressedBeamformingFeedbackOption-Implemented	Implementation dependent	Static
dot11NumberBeamFormingCSISupportAntenna	Implementation dependent	Static
dot11NumberNonCompressedBeamformingMatrixSupport-Antenna	Implementation dependent	Static
dot11NumberCompressedBeamformingMatrixSupportAntenna	Implementation dependent	Static
dot11TxMCSSetDefined	False/Boolean	Static
dot11TxRxMCSSetNotEqual	False/Boolean	Static
dot11TxMaximumNumberSpatialStreamsSupported	False/Boolean	Static
dot11TxUnequalModulationSupported	False/Boolean	Static

20.4.3 TXTIME calculation

The value of the TXTIME parameter returned by the PLME-TXTIME.confirm primitive or calculated for the PLCP receive procedure shall be calculated for HT-mixed format according to the Equation (20-91) and Equation (20-92) for short and regular GI, respectively, and for HT-greenfield format according to Equation (20-93) and Equation (20-94) for short and regular GI, respectively:

$$\begin{aligned}\mathrm{TXTIME} = {} & T_{LEG\_PREAMBLE} + T_{L\_SIG} + T_{HT\_PREAMBLE} + T_{HT\_SIG} \\ & + T_{SYM} \times \mathrm{Ceiling}\left(\frac{T_{SYMS} \times N_{SYM}}{T_{SYM}}\right) + SignalExtension\end{aligned} \qquad (20\text{-}91)$$

$$\begin{aligned}\mathrm{TXTIME} = {} & T_{LEG\_PREAMBLE} + T_{L\_SIG} + T_{HT\_PREAMBLE} + T_{HT\_SIG} \\ & + T_{\mathrm{SYM}} \times N_{SYM} + SignalExtension\end{aligned} \qquad (20\text{-}92)$$

$$\mathrm{TXTIME} = T_{GF\_HT\_PREAMBLE} + T_{HT\_SIG} + T_{SYMS} \times N_{SYM} + SignalExtension \qquad (20\text{-}93)$$

$$\mathrm{TXTIME} = T_{GF\_HT\_PREAMBLE} + T_{HT\_SIG} + T_{SYM} \times N_{SYM} + SignalExtension \qquad (20\text{-}94)$$

where

$T_{LEG\_PREAMBLE} = T_{L\text{-}STF} + T_{L\text{-}LTF}$ is the duration of the non-HT preamble

$T_{HT\_PREAMBLE}$ is the duration of the HT preamble in HT-mixed format, given by $T_{HT-STF} + T_{HT-LTF1} + (N_{LTF} - 1)T_{HT-LTFs}$

$T_{GF\_HT\_PREAMBLE}$ is the duration of the preamble in HT-greenfield format, given by $T_{HT-GF-STF} + T_{HT-LTF1} + (N_{LTF} - 1)T_{HT-LTFs}$

T_{SYM}, T_{SYMS}, $T_{HT\text{-}SIG}$, $T_{L\text{-}STF}$, $T_{HT\text{-}STF}$, $T_{HT\text{-}GF\text{-}STF}$, $T_{L\text{-}LTF}$, $T_{HT\text{-}LTF1}$ and $T_{HT\text{-}LTFs}$ are defined in Table 20-6

SignalExtension is 0 µs when TXVECTOR parameter NO_SIG_EXTN is true and is the duration of signal extension as defined by aSignalExtension in Table 20-5 when TXVECTOR parameter NO_SIG_EXTN is false

N_{LTF} is defined in Equation (20-22)

N_{SYM} is defined in Equation (20-32) for BCC and Equation (20-41) for LDPC

For non-HT modes of operation, refer to Clause 18 and Clause 19 for TXTIME calculations, except that frames transmitted with a value of NON_HT_DUP_OFDM for the TXVECTOR parameter NON_HT_MODULATION shall use Equation (19-6) for TXTIME calculation.

20.4.4 PHY characteristics

The static HT PHY characteristics, provided through the PLME-CHARACTERISTICS service primitive, shall be as shown in Table 20-25. The definitions for these characteristics are given in 10.4.

Table 20-25—MIMO PHY characteristics

Characteristics	Value
aRIFSTime	2 µs
aSlotTime	When operating in the 2.4 GHz band: long = 20 µs, short = 9 µs When operating in the 5 GHz band: 9 µs
aSIFSTime	10 µs when operating in the 2.4 GHz band 16 µs when operating in the 5 GHz bands
aSignalExtension	0 µs when operating in the 5 GHz band 6 µs when operating in the 2.4 GHz band
aCCATime	< 4 µs
aPHY-RX-START-Delay	33 µs for both MF and GF
aRxTxTurnaroundTime	< 2 µs
aTxPLCPDelay	Implementation dependent
aRxPLCPDelay	Implementation dependent
aRxTxSwitchTime	<< 1 µs
aTxRampOnTime	Implementation dependent
aTxRampOffTime	Implementation dependent
aTxRFDelay	Implementation dependent
aRxRFDelay	Implementation dependent
aAirPropagationTime	<< 1 µs
aMACProcessingDelay	< 2 µs
aPreambleLength	16 µs
aSTFOneLength	8 µs
aSTFTwoLength	4 µs
aLTFOneLength	8 µs
aLTFTwoLength	4 µs
aPLCPHeaderLength	4 µs
aPLCPSigTwoLength	8 µs
aPLCPServiceLength	16 bits
aPLCPConvolutionalTailLength	6 bits
aPSDUMaxLength	65 535 octets

Table 20-25—MIMO PHY characteristics *(continued)*

Characteristics	Value
aPPDUMaxTime	10 ms
aIUStime	8 µs
aDTT2UTTTime	32 µs
aCWmin	15
aCWmax	1023
aMaxCSIMatricesReportDelay	250 ms

For non-HT modes of operation, refer to Clause 18 and Clause 19 for PHY characteristics.

20.5 HT PMD sublayer

20.5.1 Scope and field of application

The PMD services provided to the PLCP for the High Throughput (HT) PHY are described in 20.5. Also defined in this subclause are the functional, electrical, and RF characteristics required for interoperability of implementations conforming to this specification. The relationship of this specification to the entire HT PHY is shown in Figure 20-28.

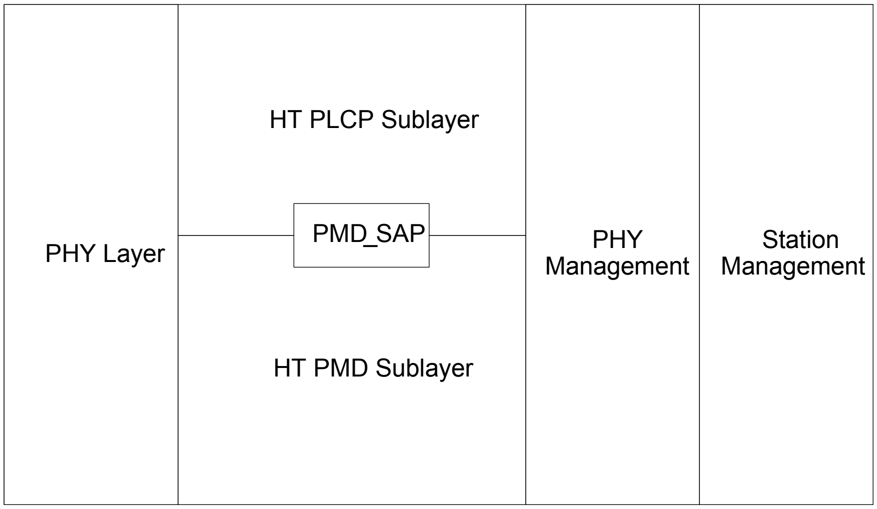


Figure 20-28—PMD layer reference model

20.5.2 Overview of service

The HT PMD sublayer accepts PLCP sublayer service primitives and provides the actual means by which data are transmitted or received from the medium. The combined function of the HT PMD sublayer primitives and parameters for the receive function results in a data stream, timing information, and associated receive signal parameters being delivered to the PLCP sublayer. A similar functionality is provided for data transmission.

20.5.3 Overview of interactions

The primitives provided by the HT PMD fall into two basic categories:

a) Service primitives that support PLCP peer-to-peer interactions

b) Service primitives that have local significance and support sublayer-to-sublayer interactions

20.5.4 Basic service and options

20.5.4.1 Status of service primitives

All of the service primitives described in 20.5.4 are mandatory, unless otherwise specified.

20.5.4.2 PMD_SAP peer-to-peer service primitives

Table 20-26 indicates the primitives for peer-to-peer interactions.

Table 20-26—PMD_SAP peer-to-peer service primitives

Primitive	Request	Indicate	Confirm	Response
PMD_DATA	X	X	—	—

20.5.4.3 PMD_SAP sublayer-to-sublayer service primitives

Table 20-27 indicates the primitives for sublayer-to-sublayer interactions.

Table 20-27—PMD_SAP sublayer-to-sublayer service primitives

Primitive	Request	Indicate	Confirm	Response
PMD_TXSTART	X	—	—	—
PMD_TXEND	X	—	—	—
PMD_TXPWRLVL	X	—	—	—
PMD_TX_PARAMETERS	X	—	—	—
PMD_RSSI	—	X	—	—
PMD_RCPI	—	X	—	—
PMD_CHAN_MAT	—	X	—	—
PMD_FORMAT	—	X	—	—
PMD_CBW_OFFSET	—	X	—	—

20.5.4.4 PMD_SAP service primitive parameters

Table 20-28 shows the parameters used by one or more of the PMD_SAP service primitives.

Table 20-28—List of parameters for PMD primitives

Parameter	Associated primitive	Value
TXD_UNIT	PMD_DATA.request	One OFDM symbol value, N_{DBPS} bits (depending on MCS).
RXD_UNIT	PMD_DATA.indication	Bit, either 0 or 1.
TXPWR_LEVEL	PMD_TXPWRLVL.request	1 to 8 (maximum of 8 levels).
MCS	PMD_TX_PARAMETERS.request	0 to 76, MCS index defined in 20.6.
CH_BANDWIDTH	PMD_TX_PARAMETERS.request PMD_CBW_OFFSET.indication	The CH_BANDWIDTH parameter indicates whether the packet is transmitted using 40 MHz or 20 MHz channel width. Enumerated type: HT_CBW20, for 20 MHz and for 40 MHz upper and lower modes HT_CBW40, for 40 MHz.
CH_OFFSET	PMD_TX_PARAMETERS.request PMD_CBW_OFFSET.indication	Enumerated type: CH_OFF_20 indicates the use of a 20 MHz channel (that is not part of a 40 MHz channel). CH_OFF_40 indicates the entire 40 MHz channel. CH_OFF_20U indicates the upper 20 MHz of the 40 MHz channel CH_OFF_20L indicates the lower 20 MHz of the 40 MHz channel.
STBC	PMD_TX_PARAMETERS.request	Set to a nonzero number indicates the difference between the number of space-time streams N_{STS} and the number of spatial streams N_{SS} indicated by the MCS. Set to 00 indicates no STBC (N_{STS}=N_{SS}).
GI_TYPE	PMD_TX_PARAMETERS.request	Set to 0 indicates short GI is not used in the packet. Set to 1 indicates short GI is used in the packet.
ANTENNA_SET	PMD_TX_PARAMETERS.request	Bit field with 8 bits.
EXPANSION_MAT	PMD_TX_PARAMETERS.request	$(N_{SD} + N_{SP})$ complex matrices of size $(N_{TX} \times N_{STS})$
EXPANSION_MAT_ TYPE	PMD_TX_PARAMETERS.request	COMPRESSED_SV: EXPANSION_MAT contains a set of compressed beamforming feedback matrices. NON_COMPRESSED_SV: EXPANSION_MAT contains a set of noncompressed beamforming feedback matrices. CSI_MATRICES: EXPANSION_MAT contains a set of CSI matrices
RSSI	PMD_RSSI.indication	0 to 255.
RCPI	PMD_RCPI.indication	0 to 255; see 20.3.21.6 for definition of each value.
CHAN_MAT	PMD_CHAN_MAT.indication	$(N_{SD} + N_{SP})$ complex matrices of size $(N_{RX} \times N_{STS})$.

Table 20-28—List of parameters for PMD primitives *(continued)*

Parameter	Associated primitive	Value
FORMAT	PMD_FORMAT.indication	Set to 0 for NON_HT. Set to 1 for HT_MF. Set to 2 for HT_GF.
FEC_CODING	PMD_TX_PARAMETERS.request	Indicates which FEC encoding is used. Enumerated type: BCC_CODING indicates binary convolutional code. LDPC_CODING indicates low-density parity check code.

20.5.5 PMD_SAP detailed service specification

20.5.5.1 Introduction to PMD_SAP service specification

Subclauses 20.5.5.2 through 20.5.5.13 describe the services provided by each PMD primitive.

20.5.5.2 PMD_DATA.request

20.5.5.2.1 Function

This primitive defines the transfer of data from the PLCP sublayer to the PMD entity.

20.5.5.2.2 Semantics of the service primitive

This primitive shall provide the following parameter:

```
  PMD_DATA.request(
                              TXD_UNIT
                              )
```

The TXD_UNIT parameter shall be the *n*-bit combination of 0 and 1 for one symbol of OFDM modulation. If the length of a coded PSDU (C-PSDU) is shorter than *n* bits, 0 bits are added to form an OFDM symbol. This parameter represents a single block of data that, in turn, shall be used by the PHY to be encoded into an OFDM transmitted symbol.

20.5.5.2.3 When generated

This primitive shall be generated by the PLCP sublayer to request transmission of one OFDM symbol.

20.5.5.2.4 Effect of receipt

The PMD performs transmission of the data.

20.5.5.3 PMD_DATA.indication

20.5.5.3.1 Function

This primitive defines the transfer of data from the PMD entity to the PLCP sublayer.

20.5.5.3.2 Semantics of the service primitive

This primitive shall provide the following parameter:

```
PMD_DATA.indication(
                              RXD_UNIT
                              )
```

The RXD_UNIT parameter shall be 0 or 1 and shall represent either a SIGNAL field bit or a data field bit after the decoding of the FEC by the PMD entity.

20.5.5.3.3 When generated

This primitive, generated by the PMD entity, forwards received data to the PLCP sublayer.

20.5.5.3.4 Effect of receipt

The PLCP sublayer decodes the bits that it receives from the PMD and either interprets them as part of its own signaling or passes them to the MAC sublayer as part of the PSDU after any necessary additional processing (e.g., descrambling).

20.5.5.4 PMD_TXSTART.request

20.5.5.4.1 Function

This primitive, generated by the PHY PLCP sublayer, initiates PPDU transmission by the PMD layer.

20.5.5.4.2 Semantics of the service primitive

This primitive has no parameters.

20.5.5.4.3 When generated

This primitive shall be generated by the PLCP sublayer to initiate the PMD layer transmission of the PPDU. The PHY-TXSTART.request primitive shall be provided to the PLCP sublayer prior to issuing the PMD_TXSTART command.

20.5.5.4.4 Effect of receipt

PMD_TXSTART initiates transmission of a PPDU by the PMD sublayer.

20.5.5.5 PMD_TXEND.request

20.5.5.5.1 Function

This primitive, generated by the PHY PLCP sublayer, ends PPDU transmission by the PMD layer.

20.5.5.5.2 Semantics of the service primitive

This primitive has no parameters.

20.5.5.5.3 When generated

This primitive shall be generated by the PLCP sublayer to terminate the PMD layer transmission of the PPDU.

20.5.5.5.4 Effect of receipt

PMD_TXEND terminates transmission of a PPDU by the PMD sublayer.

20.5.5.6 PMD_TXEND.confirm

20.5.5.6.1 Function

This primitive, generated by the PMD entity, indicates the end of PPDU transmission by the PMD layer. It is generated at the 4 µs boundary following the trailing boundary of the last symbol transmitted.

20.5.5.6.2 Semantics of the service primitive

This primitive has no parameters.

20.5.5.6.3 When generated

This primitive shall be generated by the PMD entity at the 4 µs boundary following the trailing boundary of the last symbol transmitted.

20.5.5.6.4 Effect of receipt

The PLCP sublayer determines that transmission of the last symbol of the PPDU is complete. This completion is used as a timing reference in the PLCP state machines. See 20.3.22.

20.5.5.7 PMD_TXPWRLVL.request

20.5.5.7.1 Function

This primitive, generated by the PHY PLCP sublayer, selects the power level used by the PHY for transmission.

20.5.5.7.2 Semantics of the service primitive

This primitive shall provide the following parameter:

```
PMD_TXPWRLVL.request(
                              TXPWR_LEVEL
                              )
```

TXPWR_LEVEL selects which of the transmit power levels should be used for the current packet transmission. The number of available power levels shall be determined by the MIB parameter aNumberSupportedPowerLevels. See 20.3.20.3 for further information on the OFDM PHY power level control capabilities.

20.5.5.7.3 When generated

This primitive shall be generated by the PLCP sublayer to select a specific transmit power. This primitive shall be applied prior to setting PMD_TXSTART into the transmit state.

20.5.5.7.4 Effect of receipt

PMD_TXPWRLVL immediately sets the transmit power level to the level given by TXPWR_LEVEL.

20.5.5.8 PMD_RSSI.indication

20.5.5.8.1 Function

This primitive, generated by the PMD sublayer, provides the receive signal strength to the PLCP and MAC entity.

20.5.5.8.2 Semantics of the service primitive

This primitive shall provide the following parameter:

```
  PMD_RSSI.indication(
                                     RSSI
                                     )
```

The RSSI shall be a measure of the RF energy received by the HT PHY. RSSI indications of up to 8 bits (256 levels) are supported.

20.5.5.8.3 When generated

This primitive shall be generated by the PMD after the reception of the HT training fields.

20.5.5.8.4 Effect of receipt

This parameter shall be provided to the PLCP layer for information only. The RSSI may be used as part of a CCA scheme.

20.5.5.9 PMD_RCPI.indication

20.5.5.9.1 Function

This primitive, generated by the PMD sublayer, provides the received channel power indicator to the PLCP and MAC entity.

20.5.5.9.2 Semantics of the service primitive

The primitive shall provide the following parameter:

```
  PMD_RCPI.indication(
                                     RCPI
                                     )
```

The RCPI is a measure of the channel power received by the OFDM PHY. RCPI measurement and parameter values are defined in 20.3.21.6.

20.5.5.9.3 When generated

This primitive shall be generated by the PMD when the OFDM PHY is in the receive state. It is generated at the end of the last received symbol.

20.5.5.9.4 Effect of receipt

This parameter shall be provided to the PLCP layer for information only. The RCPI may be used in conjunction with RSSI to measure input signal quality.

20.5.5.10 PMD_TX_PARAMETERS.request

20.5.5.10.1 Function

This primitive, generated by the PHY PLCP sublayer, selects the related parameters used by the PHY for transmission.

20.5.5.10.2 Semantics of the service primitive

This primitive shall provide the following parameters:

```
PMD_TX_PARAMETERS.request(
                MCS,
                CH_BANDWIDTH,
                CH_OFFSET,
                STBC,
                GI_TYPE,
                ANTENNA_SET,
                FEC_CODING,
                PMD_EXPANSIONS_MAT,
                PMD_EXPANSIONS_MAT_TYPE
                )
```

20.5.5.10.3 When generated

This primitive shall be generated by the PLCP sublayer to select a specific transmit parameter. This primitive shall be applied prior to setting PMD_TXSTART into the transmit state.

20.5.5.10.4 Effect of receipt

PMD_TX_PARAMETERS immediately sets the transmit parameters. The receipt of these parameters selects the values that shall be used for all subsequent PPDU transmissions.

20.5.5.11 PMD_CBW_OFFSET.indication

20.5.5.11.1 Function

This primitive, generated by the PMD sublayer, provides the bandwidth and channel offset of the received frame to the PLCP and MAC entity.

20.5.5.11.2 Semantics of the service primitive

This primitive shall provide the following parameters:

```
PMD_CBW_OFFSET.indication(
                CH_BANDWIDTH,
                CH_OFFSET
                )
```

CH_BANDWIDTH represents channel width (20 MHz or 40 MHz) in which the data are transmitted and the transmission format (non-HT duplicate or MCS 32). CH_OFFSET indicates in a 20 MHz bandwidth, in a 20 MHz subchannel of the 40 MHz channel (upper or lower), or in the entire 40 MHz channel.

20.5.5.11.3 When generated

This primitive shall be generated by the PMD when the OFDM PHY is in the receive state. It shall be available continuously to the PLCP that, in turn, shall provide the parameter to the MAC entity.

20.5.5.11.4 Effect of receipt

The PLCP sublayer passes the data to the MAC sublayer as part of the RXVECTOR.

20.5.5.12 PMD_CHAN_MAT.indication

20.5.5.12.1 Function

This primitive, generated by the PMD sublayer, provides the channel response matrices to the PLCP and MAC entity.

20.5.5.12.2 Semantics of the service primitive

This primitive shall provide the following parameter:

```
PMD_CHAN_MAT.indication(
                              CHAN_MAT
                              )
```

The CHAN_MAT parameter contains the channel response matrices that were measured during the reception of the current frame.

20.5.5.12.3 When generated

This primitive shall be generated by the PMD when the OFDM PHY is in the receive state. It shall be available continuously to the PLCP that, in turn, shall provide the parameter to the MAC entity.

20.5.5.12.4 Effect of receipt

The PLCP sublayer passes the data to the MAC sublayer as part of the RXVECTOR.

20.5.5.13 PMD_FORMAT.indication

20.5.5.13.1 Function

This primitive, generated by the PMD sublayer, provides the format of the received frame to the PLCP and MAC entity.

20.5.5.13.2 Semantics of the service primitive

This primitive shall provide the following parameter:

```
PMD_FORMAT.indication(
                              FORMAT
                              )
```

The format indicates one of the PPDU formats: non-HT, HT-mixed format, or HT-greenfield format.

20.5.5.13.3 When generated

This primitive shall be generated by the PMD after the reception of the HT training fields.

20.5.5.13.4 Effect of receipt

The PLCP sublayer passes the data to the MAC sublayer as part of the RXVECTOR.

20.6 Parameters for HT MCSs

Table 20-29 defines the symbols used in the rate-dependent parameter tables.

Table 20-29—Symbols used in MCS parameter tables

Symbol	Explanation
N_{SS}	Number of spatial streams
R	Coding rate
N_{BPSC}	Number of coded bits per single carrier (total across spatial streams)
$N_{BPSCS}(i_{SS})$	Number of coded bits per single carrier for each spatial stream, $i_{SS} = 1,...,N_{SS}$
N_{SD}	Number of complex data numbers per spatial stream per OFDM symbol
N_{SP}	Number of pilot values per OFDM symbol
N_{CBPS}	Number of coded bits per OFDM symbol
N_{DBPS}	Number of data bits per OFDM symbol
N_{ES}	Number of BCC encoders for the DATA field
N_{TBPS}	Total bits per subcarrier

The rate-dependent parameters for mandatory 20 MHz, N_{SS} = 1 MCSs with N_{ES} = 1 shall be as shown in Table 20-30.

Table 20-30—MCS parameters for mandatory 20 MHz, N_{SS} = 1, N_{ES} = 1

MCS Index	Modulation	R	$N_{BPSCS}(i_{SS})$	N_{SD}	N_{SP}	N_{CBPS}	N_{DBPS}	Data rate (Mb/s)	
								800 ns GI	400 ns GI (see NOTE)
0	BPSK	1/2	1	52	4	52	26	6.5	7.2
1	QPSK	1/2	2	52	4	104	52	13.0	14.4
2	QPSK	3/4	2	52	4	104	78	19.5	21.7
3	16-QAM	1/2	4	52	4	208	104	26.0	28.9
4	16-QAM	3/4	4	52	4	208	156	39.0	43.3
5	64-QAM	2/3	6	52	4	312	208	52.0	57.8
6	64-QAM	3/4	6	52	4	312	234	58.5	65.0
7	64-QAM	5/6	6	52	4	312	260	65.0	72.2
NOTE—Support of 400 ns GI is optional on transmit and receive.									

The rate-dependent parameters for optional 20 MHz, N_{SS} = 2 MCSs with N_{ES} = 1 and EQM of the spatial streams shall be as shown in Table 20-31.

Table 20-31—MCS parameters for optional 20 MHz, N_{SS} = 2, N_{ES} = 1, EQM

MCS Index	Modulation	*R*	$N_{BPSCS}(i_{SS})$	N_{SD}	N_{SP}	N_{CBPS}	N_{DBPS}	Data rate (Mb/s)	
								800 ns GI	400 ns GI (see NOTE)
8	BPSK	1/2	1	52	4	104	52	13.0	14.4
9	QPSK	1/2	2	52	4	208	104	26.0	28.9
10	QPSK	3/4	2	52	4	208	156	39.0	43.3
11	16-QAM	1/2	4	52	4	416	208	52.0	57.8
12	16-QAM	3/4	4	52	4	416	312	78.0	86.7
13	64-QAM	2/3	6	52	4	624	416	104.0	115.6
14	64-QAM	3/4	6	52	4	624	468	117.0	130.0
15	64-QAM	5/6	6	52	4	624	520	130.0	144.4
NOTE—The 400 ns GI rate values are rounded to 1 decimal place.									

The rate-dependent parameters for optional 20 MHz, N_{SS} = 3 MCSs with N_{ES} = 1 and EQM of the spatial streams shall be as shown in Table 20-32.

Table 20-32—MCS parameters for optional 20 MHz, N_{SS} = 3, N_{ES} = 1, EQM

MCS Index	Modulation	*R*	$N_{BPSCS}(i_{SS})$	N_{SD}	N_{SP}	N_{CBPS}	N_{DBPS}	Data rate (Mb/s)	
								800 ns GI	400 ns GI
16	BPSK	1/2	1	52	4	156	78	19.5	21.7
17	QPSK	1/2	2	52	4	312	156	39.0	43.3
18	QPSK	3/4	2	52	4	312	234	58.5	65.0
19	16-QAM	1/2	4	52	4	624	312	78.0	86.7
20	16-QAM	3/4	4	52	4	624	468	117.0	130.0
21	64-QAM	2/3	6	52	4	936	624	156.0	173.3
22	64-QAM	3/4	6	52	4	936	702	175.5	195.0
23	64-QAM	5/6	6	52	4	936	780	195.0	216.7

The rate-dependent parameters for optional 20 MHz, N_{SS} = 4 MCSs with N_{ES} = 1 and EQM of the spatial streams shall be as shown in Table 20-33.

Table 20-33—MCS parameters for optional 20 MHz, N_{SS} = 4, N_{ES} = 1, EQM

MCS Index	Modulation	R	$N_{BPSCS}(i_{SS})$	N_{SD}	N_{SP}	N_{CBPS}	N_{DBPS}	Data rate (Mb/s)	
								800 ns GI	400 ns GI
24	BPSK	1/2	1	52	4	208	104	26.0	28.9
25	QPSK	1/2	2	52	4	416	208	52.0	57.8
26	QPSK	3/4	2	52	4	416	312	78.0	86.7
27	16-QAM	1/2	4	52	4	832	416	104.0	115.6
28	16-QAM	3/4	4	52	4	832	624	156.0	173.3
29	64-QAM	2/3	6	52	4	1248	832	208.0	231.1
30	64-QAM	3/4	6	52	4	1248	936	234.0	260.0
31	64-QAM	5/6	6	52	4	1248	1040	260.0	288.9

The rate-dependent parameters for optional 40 MHz, N_{SS} = 1 MCSs with N_{ES} = 1 shall be as shown in Table 20-34.

Table 20-34—MCS parameters for optional 40 MHz, N_{SS} = 1, N_{ES} = 1

MCS Index	Modulation	R	$N_{BPSCS}(i_{SS})$	N_{SD}	N_{SP}	N_{CBPS}	N_{DBPS}	Data rate (Mb/s)	
								800 ns GI	400 ns GI
0	BPSK	1/2	1	108	6	108	54	13.5	15.0
1	QPSK	1/2	2	108	6	216	108	27.0	30.0
2	QPSK	3/4	2	108	6	216	162	40.5	45.0
3	16-QAM	1/2	4	108	6	432	216	54.0	60.0
4	16-QAM	3/4	4	108	6	432	324	81.0	90.0
5	64-QAM	2/3	6	108	6	648	432	108.0	120.0
6	64-QAM	3/4	6	108	6	648	486	121.5	135.0
7	64-QAM	5/6	6	108	6	648	540	135.0	150.0

The rate-dependent parameters for optional 40 MHz, N_{SS} = 2 MCSs with N_{ES} = 1 and EQM of the spatial streams shall be as shown in Table 20-35.

Table 20-35—MCS parameters for optional 40 MHz, N_{SS} = 2, N_{ES} = 1, EQM

MCS Index	Modulation	R	$N_{BPSCS}(i_{SS})$	N_{SD}	N_{SP}	N_{CBPS}	N_{DBPS}	Data rate (Mb/s)	
								800 ns GI	400 ns GI
8	BPSK	1/2	1	108	6	216	108	27.0	30.0
9	QPSK	1/2	2	108	6	432	216	54.0	60.0
10	QPSK	3/4	2	108	6	432	324	81.0	90.0
11	16-QAM	1/2	4	108	6	864	432	108.0	120.0
12	16-QAM	3/4	4	108	6	864	648	162.0	180.0
13	64-QAM	2/3	6	108	6	1296	864	216.0	240.0
14	64-QAM	3/4	6	108	6	1296	972	243.0	270.0
15	64-QAM	5/6	6	108	6	1296	1080	270.0	300.0

The rate-dependent parameters for optional 40 MHz, N_{SS} = 3 MCSs, with EQM of the spatial streams shall be as shown in Table 20-36.

Table 20-36—MCS parameters for optional 40 MHz, N_{SS} = 3, EQM

MCS Index	Modulation	R	$N_{BPSCS}(i_{SS})$	N_{SD}	N_{SP}	N_{CBPS}	N_{DBPS}	N_{ES}	Data rate (Mb/s)	
									800 ns GI	400 ns GI
16	BPSK	1/2	1	108	6	324	162	1	40.5	45.0
17	QPSK	1/2	2	108	6	648	324	1	81.0	90.0
18	QPSK	3/4	2	108	6	648	486	1	121.5	135.0
19	16-QAM	1/2	4	108	6	1296	648	1	162.0	180.0
20	16-QAM	3/4	4	108	6	1296	972	1	243.0	270.0
21	64-QAM	2/3	6	108	6	1944	1296	2	324.0	360.0
22	64-QAM	3/4	6	108	6	1944	1458	2	364.5	405.0
23	64-QAM	5/6	6	108	6	1944	1620	2	405.0	450.0

The rate-dependent parameters for optional 40 MHz, N_{SS} = 4 MCSs, with EQM of the spatial streams shall be as shown in Table 20-37.

Table 20-37—MCS parameters for optional 40 MHz, N_{SS} = 4, EQM

MCS Index	Modulation	R	$N_{BPSCS}(i_{SS})$	N_{SD}	N_{SP}	N_{CBPS}	N_{DBPS}	N_{ES}	Data rate (Mb/s)	
									800 ns GI	400 ns GI
24	BPSK	1/2	1	108	6	432	216	1	54.0	60.0
25	QPSK	1/2	2	108	6	864	432	1	108.0	120.0
26	QPSK	3/4	2	108	6	864	648	1	162.0	180.0
27	16-QAM	1/2	4	108	6	1728	864	1	216.0	240.0
28	16-QAM	3/4	4	108	6	1728	1296	2	324.0	360.0
29	64-QAM	2/3	6	108	6	2592	1728	2	432.0	480.0
30	64-QAM	3/4	6	108	6	2592	1944	2	486.0	540.0
31	64-QAM	5/6	6	108	6	2592	2160	2	540.0	600.0

The rate-dependent parameters for optional 40 MHz MCS 32 format with N_{SS} = 1 and N_{ES} = 1 shall be as shown in Table 20-38.

Table 20-38—MCS parameters for optional 40 MHz MCS 32 format, N_{SS} = 1, N_{ES} = 1

MCS Index	Modulation	R	$N_{BPSCS}(i_{SS})$	N_{SD}	N_{SP}	N_{CBPS}	N_{DBPS}	Data rate (Mb/s)	
								800 ns GI	400 ns GI
32	BPSK	1/2	1	48	4	48	24	6.0	6.7

The rate-dependent parameters for optional 20 MHz, N_{SS} = 2 MCSs with N_{ES} = 1 and UEQM of the spatial streams shall be as shown in Table 20-39.

Table 20-39—MCS parameters for optional 20 MHz, N_{SS} = 2, N_{ES} = 1, UEQM

MCS Index	Modulation		R	N_{BPSC}	N_{SD}	N_{SP}	N_{CBPS}	N_{DBPS}	Data rate (Mb/s)	
	Stream 1	Stream 2							800 ns GI	400 ns GI
33	16-QAM	QPSK	1/2	6	52	4	312	156	39	43.3
34	64-QAM	QPSK	1/2	8	52	4	416	208	52	57.8
35	64-QAM	16-QAM	1/2	10	52	4	520	260	65	72.2
36	16-QAM	QPSK	3/4	6	52	4	312	234	58.5	65.0
37	64-QAM	QPSK	3/4	8	52	4	416	312	78	86.7
38	64-QAM	16-QAM	3/4	10	52	4	520	390	97.5	108.3

The rate-dependent parameters for optional 20 MHz, N_{SS} = 3 MCSs with N_{ES} = 1 and UEQM of the spatial streams shall be as shown in Table 20-40.

Table 20-40—MCS parameters for optional 20 MHz, N_{SS} = 3, N_{ES} = 1, UEQM

MCS Index	Modulation			R	N_{BPSC}	N_{SD}	N_{SP}	N_{CBPS}	N_{DBPS}	Data rate (Mb/s)	
	Stream 1	Stream 2	Stream 3							800 ns GI	400 ns GI
39	16-QAM	QPSK	QPSK	1/2	8	52	4	416	208	52	57.8
40	16-QAM	16-QAM	QPSK	1/2	10	52	4	520	260	65	72.2
41	64-QAM	QPSK	QPSK	1/2	10	52	4	520	260	65	72.2
42	64-QAM	16-QAM	QPSK	1/2	12	52	4	624	312	78	86.7
43	64-QAM	16-QAM	16-QAM	1/2	14	52	4	728	364	91	101.1
44	64-QAM	64-QAM	QPSK	1/2	14	52	4	728	364	91	101.1
45	64-QAM	64-QAM	16-QAM	1/2	16	52	4	832	416	104	115.6
46	16-QAM	QPSK	QPSK	3/4	8	52	4	416	312	78	86.7
47	16-QAM	16-QAM	QPSK	3/4	10	52	4	520	390	97.5	108.3
48	64-QAM	QPSK	QPSK	3/4	10	52	4	520	390	97.5	108.3
49	64-QAM	16-QAM	QPSK	3/4	12	52	4	624	468	117	130.0
50	64-QAM	16-QAM	16-QAM	3/4	14	52	4	728	546	136.5	151.7
51	64-QAM	64-QAM	QPSK	3/4	14	52	4	728	546	136.5	151.7
52	64-QAM	64-QAM	16-QAM	3/4	16	52	4	832	624	156	173.3

The rate-dependent parameters for optional 20 MHz, N_{SS} = 4 MCSs with N_{ES} = 1 and UEQM in the spatial streams shall be as shown in Table 20-41.

Table 20-41—MCS parameters for optional 20 MHz, N_{SS} = 4, N_{ES} = 1, UEQM

MCS Index	Modulation				R	N_{BPSC}	N_{SD}	N_{SP}	N_{CBPS}	N_{DBPS}	Data rate (Mb/s)	
	Stream 1	Stream 2	Stream 3	Stream 4							800 ns GI	400 ns GI
53	16-QAM	QPSK	QPSK	QPSK	1/2	10	52	4	520	260	65	72.2
54	16-QAM	16-QAM	QPSK	QPSK	1/2	12	52	4	624	312	78	86.7
55	16-QAM	16-QAM	16-QAM	QPSK	1/2	14	52	4	728	364	91	101.1
56	64-QAM	QPSK	QPSK	QPSK	1/2	12	52	4	624	312	78	86.7
57	64-QAM	16-QAM	QPSK	QPSK	1/2	14	52	4	728	364	91	101.1
58	64-QAM	16-QAM	16-QAM	QPSK	1/2	16	52	4	832	416	104	115.6

Table 20-41—MCS parameters for optional 20 MHz, N_{SS} = 4, N_{ES} = 1, UEQM *(continued)*

MCS Index	Modulation				*R*	N_{BPSC}	N_{SD}	N_{SP}	N_{CBPS}	N_{DBPS}	Data rate (Mb/s)	
	Stream 1	Stream 2	Stream 3	Stream 4							800 ns GI	400 ns GI
59	64-QAM	16-QAM	16-QAM	16-QAM	1/2	18	52	4	936	468	117	130.0
60	64-QAM	64-QAM	QPSK	QPSK	1/2	16	52	4	832	416	104	115.6
61	64-QAM	64-QAM	16-QAM	QPSK	1/2	18	52	4	936	468	117	130.0
62	64-QAM	64-QAM	16-QAM	16-QAM	1/2	20	52	4	1040	520	130	144.4
63	64-QAM	64-QAM	64-QAM	QPSK	1/2	20	52	4	1040	520	130	144.4
64	64-QAM	64-QAM	64-QAM	16-QAM	1/2	22	52	4	1144	572	143	158.9
65	16-QAM	QPSK	QPSK	QPSK	3/4	10	52	4	520	390	97.5	108.3
66	16-QAM	16-QAM	QPSK	QPSK	3/4	12	52	4	624	468	117	130.0
67	16-QAM	16-QAM	16-QAM	QPSK	3/4	14	52	4	728	546	136.5	151.7
68	64-QAM	QPSK	QPSK	QPSK	3/4	12	52	4	624	468	117	130.0
69	64-QAM	16-QAM	QPSK	QPSK	3/4	14	52	4	728	546	136.5	151.7
70	64-QAM	16-QAM	16-QAM	QPSK	3/4	16	52	4	832	624	156	173.3
71	64-QAM	16-QAM	16-QAM	16-QAM	3/4	18	52	4	936	702	175.5	195.0
72	64-QAM	64-QAM	QPSK	QPSK	3/4	16	52	4	832	624	156	173.3
73	64-QAM	64-QAM	16-QAM	QPSK	3/4	18	52	4	936	702	175.5	195.0
74	64-QAM	64-QAM	16-QAM	16-QAM	3/4	20	52	4	1040	780	195	216.7
75	64-QAM	64-QAM	64-QAM	QPSK	3/4	20	52	4	1040	780	195	216.7
76	64-QAM	64-QAM	64-QAM	16-QAM	3/4	22	52	4	1144	858	214.5	238.3

The rate-dependent parameters for optional 40 MHz, N_{SS} = 2 MCSs with N_{ES} = 1 and UEQM of the spatial streams shall be as shown in Table 20-42.

Table 20-42—MCS parameters for optional 40 MHz, N_{SS} = 2, N_{ES} = 1, UEQM

MCS Index	Modulation		R	N_{BPSC}	N_{SD}	N_{SP}	N_{CBPS}	N_{DBPS}	Data rate (Mb/s)	
	Stream 1	Stream 2							800 ns GI	400 ns GI
33	16-QAM	QPSK	1/2	6	108	6	648	324	81	90
34	64-QAM	QPSK	1/2	8	108	6	864	432	108	120
35	64-QAM	16-QAM	1/2	10	108	6	1080	540	135	150
36	16-QAM	QPSK	3/4	6	108	6	648	486	121.5	135
37	64-QAM	QPSK	3/4	8	108	6	864	648	162	180
38	64-QAM	16-QAM	3/4	10	108	6	1080	810	202.5	225

The rate-dependent parameters for optional 40 MHz, N_{SS} = 3 MCSs, with UEQM of the spatial streams shall be as shown in Table 20-43.

Table 20-43—MCS parameters for optional 40 MHz, N_{SS} = 3, UEQM

MCS Index	Modulation			R	N_{BPSC}	N_{SD}	N_{SP}	N_{CBPS}	N_{DBPS}	N_{ES}	Data rate (Mb/s)	
	Stream 1	Stream 2	Stream 3								800 ns GI	400 ns GI
39	16-QAM	QPSK	QPSK	1/2	8	108	6	864	432	1	108	120
40	16-QAM	16-QAM	QPSK	1/2	10	108	6	1080	540	1	135	150
41	64-QAM	QPSK	QPSK	1/2	10	108	6	1080	540	1	135	150
42	64-QAM	16-QAM	QPSK	1/2	12	108	6	1296	648	1	162	180
43	64-QAM	16-QAM	16-QAM	1/2	14	108	6	1512	756	1	189	210
44	64-QAM	64-QAM	QPSK	1/2	14	108	6	1512	756	1	189	210
45	64-QAM	64-QAM	16-QAM	1/2	16	108	6	1728	864	1	216	240
46	16-QAM	QPSK	QPSK	3/4	8	108	6	864	648	1	162	180
47	16-QAM	16-QAM	QPSK	3/4	10	108	6	1080	810	1	202.5	225
48	64-QAM	QPSK	QPSK	3/4	10	108	6	1080	810	1	202.5	225
49	64-QAM	16-QAM	QPSK	3/4	12	108	6	1296	972	1	243	270
50	64-QAM	16-QAM	16-QAM	3/4	14	108	6	1512	1134	1	283.5	315
51	64-QAM	64-QAM	QPSK	3/4	14	108	6	1512	1134	1	283.5	315
52	64-QAM	64-QAM	16-QAM	3/4	16	108	6	1728	1296	2	324	360

The rate-dependent parameters for optional 40 MHz, N_{SS} = 4 MCSs, with UEQM of the spatial streams shall be as shown in Table 20-44.

Table 20-44—MCS parameters for optional 40 MHz, N_{SS} = 4, UEQM

MCS Index	Modulation				R	N_{BPSC}	N_{SD}	N_{SP}	N_{CBPS}	N_{DBPS}	N_{ES}	Data rate (Mb/s)	
	Stream 1	Stream 2	Stream 3	Stream 4								800 ns GI	400 ns GI
53	16-QAM	QPSK	QPSK	QPSK	1/2	10	108	6	1080	540	1	135	150
54	16-QAM	16-QAM	QPSK	QPSK	1/2	12	108	6	1296	648	1	162	180
55	16-QAM	16-QAM	16-QAM	QPSK	1/2	14	108	6	1512	756	1	189	210
56	64-QAM	QPSK	QPSK	QPSK	1/2	12	108	6	1296	648	1	162	180
57	64-QAM	16-QAM	QPSK	QPSK	1/2	14	108	6	1512	756	1	189	210
58	64-QAM	16-QAM	16-QAM	QPSK	1/2	16	108	6	1728	864	1	216	240
59	64-QAM	16-QAM	16-QAM	16-QAM	1/2	18	108	6	1944	972	1	243	270
60	64-QAM	64-QAM	QPSK	QPSK	1/2	16	108	6	1728	864	1	216	240
61	64-QAM	64-QAM	16-QAM	QPSK	1/2	18	108	6	1944	972	1	243	270
62	64-QAM	64-QAM	16-QAM	16-QAM	1/2	20	108	6	2160	1080	1	270	300
63	64-QAM	64-QAM	64-QAM	QPSK	1/2	20	108	6	2160	1080	1	270	300
64	64-QAM	64-QAM	64-QAM	16-QAM	1/2	22	108	6	2376	1188	1	297	330
65	16-QAM	QPSK	QPSK	QPSK	3/4	10	108	6	1080	810	1	202.5	225
66	16-QAM	16-QAM	QPSK	QPSK	3/4	12	108	6	1296	972	1	243	270
67	16-QAM	16-QAM	16-QAM	QPSK	3/4	14	108	6	1512	1134	1	283.5	315
68	64-QAM	QPSK	QPSK	QPSK	3/4	12	108	6	1296	972	1	243	270
69	64-QAM	16-QAM	QPSK	QPSK	3/4	14	108	6	1512	1134	1	283.5	315
70	64-QAM	16-QAM	16-QAM	QPSK	3/4	16	108	6	1728	1296	2	324	360

Table 20-44—MCS parameters for optional 40 MHz, N_{SS} = 4, UEQM *(continued)*

MCS Index	Modulation				R	N_{BPSC}	N_{SD}	N_{SP}	N_{CBPS}	N_{DBPS}	N_{ES}	Data rate (Mb/s)	
	Stream 1	Stream 2	Stream 3	Stream 4								800 ns GI	400 ns GI
71	64-QAM	16-QAM	16-QAM	16-QAM	3/4	18	108	6	1944	1458	2	364.5	405
72	64-QAM	64-QAM	QPSK	QPSK	3/4	16	108	6	1728	1296	2	324	360
73	64-QAM	64-QAM	16-QAM	QPSK	3/4	18	108	6	1944	1458	2	364.5	405
74	64-QAM	64-QAM	16-QAM	16-QAM	3/4	20	108	6	2160	1620	2	405	450
75	64-QAM	64-QAM	64-QAM	QPSK	3/4	20	108	6	2160	1620	2	405	450
76	64-QAM	64-QAM	64-QAM	16-QAM	3/4	22	108	6	2376	1782	2	445.5	495

Annex A

(informative)

Bibliography

Bibliographical references are resources that provide additional or helpful material but do not need to be understood or used to implement this standard. Reference to these resources is made for informational use only.

[B1] 3GPP TS 23.167, IMS emergency sessions architecture: http://www.3gpp.org/ftp/Specs/html-info/23167.htm.

[B2] 3GPP TR 21.905, Vocabulary for 3GPP Specifications.[43]

[B3] 3GPP TS 22.067: Enhanced Multi-Level Precedence and Pre-emption service (EMLPP); Stage 1.

[B4] 3GPP2 X.S0060-0 IMS emergency sessions architecture: http://www.3gpp2.org/Public_html/specs/X.S0060-0_v1.0_080729.pdf.

[B5] ANSI Z136.1-1993, American National Standard for the Safe Use of Lasers.[44]

[B6] Arazi, E. G., *A Commonsense Approach to the Theory of Error Correcting Codes*, MIT Press, 1988.

[B7] ARIB RCR STD-33 (5.0), Low Power Data Communication/Wireless LAN System, ARIB, Feb. 1999.[45]

[B8] ARIB STD-T71 (5.0), Broadband Mobile Access Communication System (CSMA), ARIB, Dec. 2007.

[B9] Code of Federal Regulations (CFR), Title 47, Telecommunication, Oct. 2001.[46]

[B10] Code of Federal Regulations, Title 47, Telecommunication, Part 90, Private Land Mobile Radio Services, Section 90.210(m), Emission masks.

[B11] Engwer, D., and Zweig, J., “Algorithmically Derived Hop Sequences,” submission 99/195 to the IEEE P802.11 Working Group, Sept. 1999.

[B12] “Ethernet, a local area network: Data link layer and physical layer specifications version 1.0.” Digital Equipment Corporation, Intel Corporation, Xerox Corporation, Sept. 1980.

[B13] ETSI EN 300 328, Electromagnetic compatibility and Radio spectrum Matters (ERM); Wideband transmission systems; Data transmission equipment operating in the 2,4 GHz ISM band and using wide band modulation techniques; Harmonized EN covering essential requirements under article 3.2 of the R&TTE Directive.[47]

[43] 3GPP documents are available from the 3rd Generation Partnership Project Web site (http://www.3gpp.org).

[44] ANSI publications are available from the American National Standards Institute (http://www.ansi.org).

[45] ARIB publications are available from the Association of Radio Industries and Businesses (www.arib.or.jp).

[46] The CFR is available from the U.S. Government Printing Office (www.gpo.gov).

[47] ETSI publications are available from the European Telecommunications Standards Institute (www.etsi.org).

[B14] ETSI EN 302 571 V1.1.1 (2008-09), Intelligent Transport Systems (ITS); Radiocommunications equipment operating in the 5 855 MHz to 5 925 MHz frequency band; Harmonized EN covering the essential requirements of article 3.2 of the R&TTE Directive.

[B15] ETSI ES 202 663 V1.1.0 (2010-01), Intelligent Transport Systems (ITS); European profile standard for the physical and medium access control layer of Intelligent Transport Systems operating in the 5 GHz frequency band.

[B16] GSMA, IR.34 v4.6, Inter-Service Provider IP Backbone Guidelines, http://gsmworld.com/documents/IR3446.pdf, Apr. 2009.

[B17] IANA Protocol Assignments for the Internet Key Exchange (IKE) Attributes, http://www.iana.org/assignments/ipsec-registry.

[B18] IEC 60825-1:1993, Safety of laser products—Part 1: Equipment classification, requirements and user's guide.[48]

[B19] IEEE Standards Registration Authority—Frequently Asked Questions. [49]

[B20] IEEE Std 802.1D™-2004, IEEE Standard for Local and metropolitan area networks—Media Access Control (MAC) Bridges.[50,51]

[B21] IEEE Std 802.1H™-1995, IEEE Standards for Local and metropolitan area networks—Recommended Practice for Media Access Control (MAC) Bridging of Ethernet V2.0 in IEEE 802 Local Area Networks [now known as ISO/IEC Technical Report 11802-5:1997(E); see Clause 2].

[B22] IEEE Std 802.1Q™, 2003 Edition, IEEE Standards for Local and metropolitan area networks—Virtual Bridged Local Area Networks.

[B23] IETF ECRIT, Extended ECRIT architecture supporting unauthenticated emergency services, http://tools.ietf.org/html/draft-schulzrinne-ecrit-unauthenticated-access.

[B24] IETF RFC 1157-1990, Simple Network Management Protocol (SNMP), J. Case, M. Fedor, M. Schoffstall, and J. Davin, May 1990.[52]

[B25] IETF RFC 1305-1992, Network Time Protocol (Version 3) Specification, Implementation and Analysis.

[B26] IETF RFC 2202-1997, Test Cases for HMAC-MD5 and HMAC-SHA-1, P. Cheng and R. Glenn, Sept. 1997 (status: informational).

[B27] IETF RFC 2212-1997, Specification of the Guaranteed Quality of Service.

[B28] IETF RFC 2215-1997, General Characterization Parameters for Integrated Service Network Elements.

[48] IEC publications are available from the International Electrotechnical Commission (http://www.iec.ch/). IEC publications are also available in the United States from the American National Standards Institute (http://www.ansi.org).

[49] This document is available from The Institute of Electrical and Electronics Engineers (http://standards.ieee.org/regauth/faqs.html).

[50] The IEEE standards or products referred to in this annex are trademarks owned by The Institute of Electrical and Electronics Engineers, Inc.

[51] IEEE publications are available from The Institute of Electrical and Electronics Engineers (http://standards.ieee.org/).

[52] IETF documents (i.e., RFCs) are available for download at http://www.rfc-archive.org/.

[B29] IETF RFC 2474-1998, Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers.

[B30] IETF RFC 2548-1999, Microsoft Vendor-specific RADIUS Attributes.

[B31] IETF RFC 2865-2000, Remote Authentication Dial in User Service (RADIUS).

[B32] IETF RFC 2903, Generic AAA architecture, C. de Laat, G. Gross, L. Gommans, J. Vollbrecht, and D. Spence, Aug. 2000 (status informational).

[B33] IETF RFC 3290-2002, An Informal Management Model for Diffserv Routers.

[B34] IETF RFC 3561, Ad hoc On-Demand Distance Vector (AODV) Routing, C. Perkins, E. Belding-Royer, and S. Das, July 2003. (status: experimental).

[B35] IETF RFC 3580, IEEE 802.1X Remote Authentication Dial In User Service (RADIUS) Usage Guidelines, P. Congdon, B. Aboba, A. Smith, G. Zorn, and J. Roese, Sept. 2003.

[B36] IETF RFC 3588-2003, Diameter Base Protocol.

[B37] IETF RFC 3693, Geopriv Requirements, Feb. 2004.

[B38] IETF RFC 3748-2004, Extensible Authentication Protocol (EAP).

[B39] IETF RFC 4493-2006, The AES-CMAC Algorithm.

[B40] IETF RFC 4776, Dynamic Host Configuration Protocol (DHCPv4 and DHCPv6) Option for Civic Addresses Configuration Information, H. Schulzrinne, Nov. 2006.

[B41] IETF RFC 5031, A Uniform Resource Name (URN) for Emergency and Other Well-Known Services, H. Schulzrinne, Jan. 2008.

[B42] IETF RFC 5139, Revised Civic Location Format for Presence Information Data Format Location Object (PIDF-LO), M. Thomson, Feb. 2008.

[B43] International Code Council, Inc., International Building Code 2006, Nov. 2006, ISBN-13: 978-1-58001-251-5.

[B44] ISO 639, Codes for the Representation of Names of Languages.[53]

[B45] ISO 14962:1997, Space data and information transfer systems — ASCII encoded English.

[B46] ISO/IEC 14977:1996, Information technology — Syntactic metalanguage — Extended BNF.[54]

[B47] ITU Radio Regulations, volumes 1–4.[55]

[B48] ITU-R Recommendation TF.460-6 (2002), Standard-Frequency and Time-Signal Emissions.

[53] ISO publications are available from the ISO Central Secretariat (http://www.iso.ch/). ISO publications are also available in the United States from the American National Standards Institute (http://www.ansi.org/).

[54] ISO/IEC publications are available from the ISO Central Secretariat (http://www.iso.ch/). ISO/IEC publications are also available in the United States from the American National Standards Institute (http://www.ansi.org/).

[55] ITU, ITU-R, and ITU-T publications are available from the International Telecommunications Union (http://www.itu.int/).

[B49] ITU-T Recommendation V.41 (11/88), Code-independent error-control system.

[B50] ITU-T Recommendation X.210 (11/93), Information technology—Open systems interconnection—Basic Reference Model: Conventions for the definition of OSI services (common text with ISO/IEC 10731).

[B51] Maric, S. V., and Titlebaum, E. L., "A Class of Frequency Hop Codes with Nearly Ideal Characteristics for Use in Multiple-Access Spread-Spectrum Communications and Radar and Sonar Systems," *IEEE Transactions on Communications*, vol. 40, no. 9, pp. 1442–1447, Sept. 1992.

[B52] NENA 08-002, Functional and Interface Standards for Next Generation 9-1-1 (i3), ver. 1.0, http://www.nena.org/standards/technical/voip/functional-interface-NG911-i3.

[B53] PKCS #5 v2.0, "Password-Based Cryptography Standard," http://www.rsasecurity.com/rsalabs/node.asp?id=2127.

[B54] Rumbaugh, J., Jacobson, I., and Booch, G., *The Unified Modeling Language (UML) Reference Manual (Second Edition)*, Reading, MA: Addison-Wesley Longman, 2004.

Annex B

(normative)

Protocol Implementation Conformance Statement (PICS) proforma

B.1 Introduction

The supplier of a protocol implementation that is claimed to conform to IEEE Std 802.11-2012 shall complete the following protocol implementation conformance statement (PICS) proforma.

A completed PICS proforma is the PICS for the implementation in question. The PICS is a statement of which capabilities and options of the protocol have been implemented. This annex may not be compatible with operation in any Regulatory Domain or describe combinations of usable features in any Regulatory Domain. The PICS has a number of uses, including use:

a) By the protocol implementer, as a checklist to reduce the risk of failure to conform to the standard through oversight;

b) By the supplier and acquirer, or potential acquirer, of the implementation, as a detailed indication of the capabilities of the implementation, stated relative to the common basis for understanding provided by the standard PICS proforma;

c) By the user, or potential user, of the implementation, as a basis for initially checking the possibility of interworking with another implementation (note that, while interworking is not guaranteed, failure to interwork can often be predicted from incompatible PICS proformas);

d) By a protocol tester, as the basis for selecting appropriate tests against which to assess the claim for conformance of the implementation.

B.2 Abbreviations and special symbols

B.2.1 Symbols for Status column

M	mandatory
O	optional
O.<n>	optional, but support of at least one of the group of options labeled by the same numeral <n> is required
pred:	conditional symbol, including predicate identification

B.2.2 General abbreviations for Item and Support columns

N/A	not applicable
AD	address function capability
CF	implementation under test (IUT) configuration
DS	direct sequence
DSE	dynamic station enablement
ERP	extended rate physical layer (PHY)
FH	frequency hopping
FR	medium access control (MAC) frame capability
FS	frame sequence capability

FT	frame transmission
HRDS	high rate direct sequence
HTM	High-throughput (HT) medium access control (MAC) features
HTP	High-throughput (HT) physical layer (PHY) features
HWM	HWMP path selection protocol capability
IR	Infrared
IW	interworking with external networks
MD	multidomain
MP	Mesh protocol capability
OF	orthogonal frequency division multiplexing (OFDM)
PC	protocol capability
RC	operating classes (formerly “regulatory classes”)
RM	radio management
QB	quality-of-service (QoS) base functionality
QD	quality-of-service (QoS) enhanced distributed channel access (EDCA)
QP	quality-of-service (QoS) hybrid coordination function (HCF) controlled channel access (HCCA)
SM	spectrum management
TDLS	tunneled direct-link setup
WNM	wireless network management

B.3 Instructions for completing the PICS proforma

B.3.1 General structure of the PICS proforma

The first parts of the PICS proforma, Implementation identification and Protocol summary, are to be completed as indicated with the information necessary to identify fully both the supplier and the implementation.

The main part of the PICS proforma is a fixed questionnaire, divided into subclauses, each containing a number of individual items. Answers to the questionnaire items are to be provided in the rightmost column, either by simply marking an answer to indicate a restricted choice (usually Yes or No) or by entering a value or a set or a range of values. (Note that there are some items where two or more choices from a set of possible answers may apply. All relevant choices are to be marked in these cases.)

Each item is identified by an item reference in the first column. The second column contains the question to be answered. The third column contains the reference or references to the material that specifies the item in the main body of this standard. The remaining columns record the status of each item, i.e., whether support is mandatory, optional, or conditional, and provide the space for the answers (see also B.3.4). Marking an item as supported is to be interpreted as a statement that all relevant requirements of the subclauses and normative annexes, cited in the References column for the item, are met by the implementation.

A supplier may also provide, or be required to provide, further information, categorized as either Additional Information or Exception Information. When present, each kind of further information is to be provided in a further subclause of items labeled A<*I*> or X<*I*>, respectively, for cross-referencing purposes, where <*I*> is any unambiguous identification for the item (e.g., simply a numeral). There are no other restrictions on its format or presentation.

A completed PICS proforma, including any Additional Information and Exception Information, is the PICS for the implementation in question.

NOTE—Where an implementation is capable of being configured in more than one way, a single PICS might be able to describe all such configurations. However, the supplier has the choice of providing more than one PICS, each covering some subset of the implementation's capabilities, if this makes for easier and clearer presentation of the information.

B.3.2 Additional information

Items of Additional Information allow a supplier to provide further information intended to assist in the interpretation of the PICS. It is not intended or expected that a large quantity of information will be supplied, and a PICS can be considered complete without any such information. Examples of such Additional Information might be an outline of the ways in which an (single) implementation can be set up to operate in a variety of environments and configurations, or information about aspects of the implementation that are outside the scope of this standard but have a bearing upon the answers to some items.

References to items of Additional Information may be entered next to any answer in the questionnaire, and may be included in items of Exception Information.

B.3.3 Exception information

It may happen occasionally that a supplier wishes to answer an item with mandatory status (after any conditions have been applied) in a way that conflicts with the indicated requirement. No preprinted answer is found in the Support column for this. Instead, the supplier shall write the missing answer into the Support column, together with an X<*I*> reference to an item of Exception Information, and shall provide the appropriate rationale in the Exception Information item itself.

An implementation for which an Exception Information item is required in this way does not conform to this standard.

NOTE—A possible reason for the situation described above is that a defect in this standard has been reported, a correction for which is expected to change the requirement not met by the implementation.

B.3.4 Conditional status

The PICS proforma contains a number of conditional items. These are items for which both the applicability of the item itself, and its status if it does apply, mandatory or optional, are dependent upon whether certain other items are supported.

Where a group of items is subject to the same condition for applicability, a separate preliminary question about the condition appears at the head of the group, with an instruction to skip to a later point in the questionnaire if the N/A answer is selected. Otherwise, individual conditional items are indicated by a conditional symbol in the Status column.

A conditional symbol is of the form "<pred>:<S>", where "<pred>" is a predicate as described below, and "<S>" is one of the status symbols M or O.

If the value of the predicate is true, the conditional item is applicable, and its status is given by S: the support column is to be completed in the usual way. Otherwise, the conditional item is not relevant and the N/A answer is to be marked.

A predicate is one of the following:

a) An item-reference for an item in the PICS proforma: the value of the predicate is true if the item is marked as supported, and is false otherwise.

b) An expression constructed by combining item-references using the boolean operators “not,” “OR,” and “AND” (synonym “&”), with or without the use of parenthetical groupings: the value of the predicate is true if the expression evaluates to true and is false otherwise.

Each item referenced in a predicate, or in a preliminary question for grouped conditional items, is indicated by an asterisk in the Item column.

B.4 PICS proforma—IEEE Std 802.11-2012[56]

B.4.1 Implementation identification

Supplier	
Contact point for queries about the PICS	
Implementation Name(s) and Version(s)	
Other information necessary for full identification, e.g., name(s) and version(s) of the machines and/or operating systems(s), system names	

NOTE 1—Only the first three items are required for all implementations. Other information may be completed as appropriate in meeting the requirement for full identification.

NOTE 2—The terms Name and Version need to be interpreted appropriately to correspond with a supplier’s terminology (e.g., Type, Series, Model).

[56] *Copyright release for PICS proforma:* Users of this standard may freely reproduce the PICS proforma in this annex so that it can be used for its intended purpose and may further publish the completed PICS.

B.4.2 Protocol summary

Identification of protocol standard	IEEE Std 802.11-2012
Identification of amendments and corrigenda to this PICS proforma that have been completed as part of this PICS	Amd. : Corr. : Amd. : Corr. :
Have any exception items been required? (See B.3.3; the answer Yes means that the implementation does not conform to IEEE Std 802.11-2012.)	Yes □ No □

Date of statement (yyyy-mm-dd)	

B.4.3 IUT configuration

Item	IUT configuration	References	Status	Support
	What is the configuration of the IUT?			
* CF1	Access point (AP)	4.3	O.1	Yes □ No □
* CF2	Independent station (neither an AP nor a mesh STA)	4.3	O.1	Yes □ No □
*CF2.1	Operation in an infrastructure BSS	4.3	CF2:M	Yes □ No □ N/A □
*CF2.2	Operation in an IBSS	4.3	CF2:O	Yes □ No □ N/A □
*CF2.3	Independent station operating outside the context of a BSS (dot11OCBActivated is true)	10.20	(not CF17):O, CF17:M	Yes □ No □
NOTE—See CF21 for mesh STA				
* CF3	Frequency-hopping spread spectrum (FHSS) PHY for the 2.4 GHz band	—	O.2	Yes □ No □
* CF4	Direct sequence spread spectrum (DSSS) PHY for the 2.4 GHz band	—	O.2	Yes □ No □
CF5	Infrared (IR) PHY	—	O.2	Yes □ No □
* CF6	Orthogonal frequency division multiplexing (OFDM) PHY	—	O.2	Yes □ No □
* CF7	High-speed PHY	—	O.2	Yes □ No □
* CF8	Is multidomain operation capability implemented?	8.4.2.11, 8.4.2.12, 9.18, 10.1.4.5	O.3	Yes □ No □
* CF9	Extended Rate PHY (ERP)	Clause 19	O.2	Yes □ No □
* CF10	Is spectrum management operation supported?	8.4.1.4, 10.6	(CF6 OR CF16): O	Yes □ No □
*CF11	Is operating classes capability implemented?	8.4.2.13, 18.3.8.4.2, 18.3.8.7, 18.4.2, Annex D, Annex E	(CF6 OR CF16) &CF8& CF10:O	Yes □ No □ N/A □

B.4.3 IUT configuration *(continued)*

Item	IUT configuration	References	Status	Support
* CF12	Quality of service (QoS) supported	9.19, 9.21, 4.3.10, 4.3.15.3	O (CF16 or CF21): M	Yes □ No □ N/A □
* CF13	Is Radio Measurement supported?	8.4.1.4, 10.11	(CF6 AND CF11):O	Yes □ No □ N/A □
*CF14	Is infrastructure mode implemented?	4.3.3	O	Yes □ No □
*CF15	3.65–3.70 GHz band in United States	8.4.2.54, 10.12, 18.3.6, 18.3.10.6, Annex D, Annex E	CF6&CF8&CF10&CF11:O	Yes □ No □ N/A □
*CF16	High-throughput (HT) features	8.4.2.58	O	Yes □ No □
*CF17	5.9 GHz band	Annex E	CF6:O	Yes □ No □
*CF18	Is tunneled direct-link setup supported?	10.22	O	Yes □ No □ N/A □
*CF19	Is WNM supported?		(CF8 & CF11 & CF13 & CF15 & DSE5 & DSE6 & DSE7 & DSE8 & DSE9):O	Yes □ No □ N/A □
*CF20	Is interworking with external networks service supported?	Extended Capabilities 8.4.2.29	(CF 15, CF8 & CF11):O	Yes □ No □
*CF21	Mesh station	4.3.15	O.1	Yes □ No □
CF21.1	Operation in an MBSS	4.3.15	CF21:M	Yes □ No □ N/A □

B.4.4 MAC protocol

B.4.4.1 MAC protocol capabilities

Item	Protocol capability	References	Status	Support
	Are the following MAC protocol capabilities supported?			
PC1	Authentication service	4.5.4.2, 4.5.4.3, 11.1, 10.20, Annex J	(not CF2.3): M	Yes □ No □ N/A □
PC1.1	Authentication state	10.3	M	Yes □ No □
PC1.2	Open System authentication	11.1.2	M	Yes □ No □
PC1.3	Shared Key authentication	11.1.3, 11.4	PC2:M	Yes □ No □ N/A □
* PC2	Wired equivalent privacy (WEP) algorithm This capability is deprecated (applicable only to systems that are backward compatible).	4.5.4.4, 11.2.2, Annex J	O	Yes □ No □

B.4.4.1 MAC protocol capabilities *(continued)*

Item	Protocol capability	References	Status	Support
PC2.1	WEP encryption procedure	11.2.2	PC2:M	Yes □ No □ N/A □
PC2.2	WEP decryption procedure	11.2.2	PC2:M	Yes □ No □ N/A □
PC3	Distributed coordination function (DCF)	9.2, 9.3, Annex J	M	Yes □ No □
PC3.1	Network allocation vector (NAV) function	9.3.2.1, 9.3.4, 9.4.3.3	M	Yes □ No □
PC3.2	Interframe space usage and timing	9.3.2.3, 9.3.4, 9.3.7	M	Yes □ No □
PC3.3	Random Backoff function	9.3.3	M	Yes □ No □
PC3.4	DCF Access procedure	9.3.4.2, 9.3.4.5	M	Yes □ No □
PC3.5	Random Backoff procedure	9.3.4.3	M	Yes □ No □
PC3.6	Recovery procedures and retransmit limits	9.3.4.4	M	Yes □ No □
PC3.7	Request to send (RTS)/clear to send (CTS) procedure	9.3.2.4, 9.3.2.5, 9.3.2.6	M	Yes □ No □
PC3.8	Individually addressed MAC protocol data unit (MPDU) transfer	9.3.5	M	Yes □ No □
PC3.9	Group addressed MPDU transfer	9.3.6	M	Yes □ No □
PC3.10	MAC-level acknowledgment	9.3.2.2, 9.3.2.8	M	Yes □ No □
PC3.11	Duplicate detection and recovery	9.3.2.10	M	Yes □ No □
* PC4	Point coordinator (PC)	9.2, 9.4, Annex J	CF1:O	Yes □ No □ N/A □
PC4.1	Maintenance of contention-free period (CFP) structure and timing	9.4.2, 9.4.3	PC4:M	Yes □ No □ N/A □
PC4.2	Point coordination function (PCF) MPDU transfer from PC	9.4.4	PC4:M	Yes □ No □ N/A □
* PC4.3	PCF MPDU transfer to PC	9.4.4	PC4:O	Yes □ No □ N/A □
PC4.4	Overlapping PC provisions	9.4.4.3	PC4:M	Yes □ No □ N/A □
PC4.5	Polling list maintenance	9.4.5	PC4.3:M	Yes □ No □ N/A □
* PC5	Contention-free (CF)-Pollable	9.2, 9.4, Annex J	CF2.1:O	Yes □ No □ N/A □
PC5.1	Interpretation of CFP structure and timing	9.4.2, 9.4.3	PC5:M	Yes □ No □ N/A □
PC5.2	PCF MPDU transfer to/from and CF-Pollable station (STA)	9.4.4	PC5:M	Yes □ No □ N/A □
PC5.3	Polling list update	9.4.5	PC5:M	Yes □ No □ N/A □
PC6	Fragmentation	9.3, 9.5, Annex J	M	Yes □ No □
PC7	Defragmentation	9.3, 9.6, Annex J	M	Yes □ No □
PC8	MAC data service	9.2.8, 9.8, Annex J	M	Yes □ No □
PC8.1	ReorderableGroupAddressed service class	9.8	M	Yes □ No □

B.4.4.1 MAC protocol capabilities *(continued)*

Item	Protocol capability	References	Status	Support
PC8.2	StrictlyOrdered service class	9.8	O	Yes □ No □
PC9	Multirate support	9.7, Annex J	M	Yes □ No □
* PC10	Multiple outstanding MAC service data unit (MSDU) support	9.8, Annex J	O	Yes □ No □
PC10.1	Multiple outstanding MSDU transmission restrictions	9.8	PC10:M	Yes □ No □ N/A □
PC11	Timing synchronization function (TSF)	10.1, Annex J	(not CF2.3): M, CF2.3:O	Yes □ No □
PC11.1	Timing in an infrastructure network	10.1.2.1, 10.1.5	CF1:M	Yes □ No □ N/A □
PC11.2	Timing in an independent basic service set (IBSS)	10.1.2.2, 10.1.5	CF2.2:M	Yes □ No □ N/A □
PC11.3	Beacon generation function	10.1.3	M	Yes □ No □
PC11.4	TSF synchronization and accuracy	10.1.2, 10.1.3	(not CF2.3): M	Yes □ No □ N/A □
PC11.5	Infrastructure basic service set (BSS) initialization	10.1.4	CF1:M	Yes □ No □ N/A □
PC11.6	IBSS initialization	10.1.4	CF2.2:M	Yes □ No □ N/A □
PC11.7	Passive scanning	10.1.4	(CF2.1 or CF2.2):M	Yes □ No □ N/A □
PC11.8	Active scanning	10.1.4	(CF2.1 or CF2.2):M	Yes □ No □ N/A □
PC11.9	Probe response	10.1.4	(not CF2.3): M	Yes □ No □ N/A □
PC11.10	Hop Synchronization function	10.1.6	CF3:M	Yes □ No □ N/A □
PC12	Infrastructure power management	10.2.1, Annex J	M	Yes □ No □
PC12.1	STA power management modes	10.2.1.2, 10.2.1.9	(CF2.1 or CF2.2):M	Yes □ No □ N/A □
PC12.2	Traffic indication map (TIM) transmission	10.2.1.3, 10.2.1.4	CF1:M	Yes □ No □ N/A □
PC12.3	AP function during contention period (CP)	10.2.1.5	CF1:M	Yes □ No □ N/A □
PC12.4	AP function during CFP	10.2.1.6	PC4:M	Yes □ No □ N/A □
PC12.5	Receive function during CP	10.2.1.7	(CF2.1 or CF2.2):M	Yes □ No □ N/A □
PC12.6	Receive function during CFP	10.2.1.8	PC5:M	Yes □ No □ N/A □
PC12.7	Aging function	10.2.1.10	CF1:M	Yes □ No □ N/A □
PC13	IBSS power management	10.2.2, Annex J	CF2.2:M	Yes □ No □ N/A □
PC13.1	Initialization of power management	10.2.2.3	CF2.2:M	Yes □ No □ N/A □
PC13.2	STA power state transitions	10.2.2.4	CF2.2:M	Yes □ No □ N/A □

B.4.4.1 MAC protocol capabilities *(continued)*

Item	Protocol capability	References	Status	Support
PC13.3	Announcement traffic indication message (ATIM) and frame transmission	10.2.2.5	CF2.2:M	Yes □ No □ N/A □
PC14	Association and reassociation	4.5, 10.3, 10.3.5, 10.20, Annex J	(not CF2.3): M	Yes □ No □ N/A □
PC14.1	Association state	10.3.5	M	Yes □ No □
PC14.2	STA association procedure	10.3.5.2	CF2.1:M	Yes □ No □ N/A □
PC14.3	AP association procedure	10.3.5.3	CF1:M	Yes □ No □ N/A □
PC14.4	STA reassociation procedure	10.3.5.4	CF2.1:M	Yes □ No □ N/A □
PC14.5	AP reassociation procedure	10.3.5.5	CF1:M	Yes □ No □ N/A □
PC15	Management information base (MIB)	Annex C	M	Yes □ No □
PC15.1	dot11SMTbase, dot11SmtAuthenticationAlgorithms	Annex C	M	Yes □ No □
* PC15.2	dot11SMTprivacy	Annex C	PC2:M	Yes □ No □ N/A □
PC15.3	dot11MACbase, dot11CountersGroup, dot11MacGroupAddresses	Annex C	M	Yes □ No □
* PC15.4	dot11MACStatistics	Annex C	O	Yes □ No □
PC15.5	dot11ResourceTypeID	Annex C	M	Yes □ No □
PC16	Set dot11ShortPreambleOptionImplemented to 1	8.4.1.4	CF9:M	Yes □ No □ N/A □
PC17	Set packet binary convolutional code (PBCC) subfield as described in reference The PBCC option is obsolete. Consequently this option may be removed in a later revision of the standard.	8.4.1.4	CF9:M	Yes □ No □ N/A □
PC18	Set DSSS-OFDM subfield as described in reference The use of the DSSS-OFDM option is deprecated, and this option may be removed in a later revision of the standard.	8.4.1.4	CF9:M	Yes □ No □ N/A □
PC19	Set channel agility subfield as described in reference	8.4.1.4	CF9:M	Yes □ No □ N/A □
PC20	Set Short Slot Time subfield as described in reference	8.4.1.4	CF9:M	Yes □ No □ N/A □
PC21	Monitor each received short time slot subfield and take action as described in reference.	8.4.1.4	CF9:M	Yes □ No □ N/A □
PC22	Transmit the ERP element in each transmitted Beacon or Probe Responses in the format and with content as described in reference	8.4.1.4	CF9:M	Yes □ No □ N/A □

B.4.4.1 MAC protocol capabilities *(continued)*

Item	Protocol capability	References	Status	Support
PC23	Receive the ERP element and employ a protection mechanism when required prior to transmitting information using ERP-OFDM modulation	8.4.1.4	CF9:M	Yes □ No □ N/A □
PC24	Determine the value of aCWmin based on the characteristic rate set as described in the reference	9.3.9	CF9:M	Yes □ No □ N/A □
PC25	Transmit control response frames at the largest basic rates less than equal to the rate received and with the same PHY options or use the highest mandatory rate if no basic rate meets the above criterion	9.7	CF9:M	Yes □ No □ N/A □
PC26	Transmit group addressed frames at a rate contained in the BSSBasicRateSet parameter	9.7	CF9:M	Yes □ No □ N/A □
PC27	Transmit individually addressed frames at any supported rate selected by a rate switching mechanism as long as it is supported by the destination STA	9.7	CF9:M	Yes □ No □ N/A □
PC28	Do not transmit at a data rate higher than the greatest rate in the OperationalRateSet	9.7	CF9:M	Yes □ No □ N/A □
PC29	Use ERP element to control use of protection mechanism as described in the reference	9.23	CF9:M	Yes □ No □ N/A □
PC30	Updated NAV is long enough to cover frame and any response	9.23	CF9:M	Yes □ No □ N/A □
PC31	Support transmission of CTS-to-self sequence as described in the references	9.3.2.11	CF9:O	Yes □ No □ N/A □
PC32	Support reception of CTS-to-self sequence as described in the references	9.3.2.11	CF9:M	Yes □ No □ N/A □
PC33	Update NAV	9.23	CF9:M	Yes □ No □ N/A □
* PC34	Robust security network association (RSNA)	8.3.2.1, 8.4.1.4, 4.5.4.4, 11.8.2, 11.8.2.2, 11.8.2.4, 11.8.2.6, 11.8.2.8, 10.3.4, 10.3.5, 11.4.3	O	Yes □ No □
PC34.1	RSN element	8.4.2.27	PC34:M	Yes □ No □ N/A □
PC34.1.1	Group cipher suite	8.4.2.27	PC34.1:M	Yes □ No □ N/A □
PC34.1.2	Pairwise cipher suite list	8.4.2.27	PC34.1:M	Yes □ No □ N/A □
PC34.1.2.1	Counter mode with Cipher-block chaining Message authentication code Protocol (CCMP) data confidentiality protocol	11.4.3	PC34:M	Yes □ No □ N/A □
PC34.1.2.1.1	CCMP cryptographic encapsulation procedure	11.4.3.3	PC34.1.2.1:M	Yes □ No □ N/A □

B.4.4.1 MAC protocol capabilities *(continued)*

Item	Protocol capability	References	Status	Support
PC34.1.2.1.2	CCMP decapsulation procedure	11.4.3.4	PC34.1.2.1:M	Yes □ No □ N/A □
PC34.1.2.2	Temporal Key Integrity Protocol (TKIP) data confidentiality protocol	11.4.2	PC34:O	Yes □ No □ N/A □
PC34.1.2.2.1	TKIP cryptographic encapsulation procedure	11.4.2.1.2	PC34.1.2.2:M	Yes □ No □ N/A □
PC34.1.2.2.2	TKIP decapsulation procedure	11.4.2.1.3	PC34.1.2.2:M	Yes □ No □ N/A □
PC34.1.2.2.3	TKIP countermeasures	11.4.2.4	PC34.1.2.2:M	Yes □ No □ N/A □
PC34.1.2.2.4	TKIP security services management	11.4.2.3	PC34.1.2.2:M	Yes □ No □ N/A □
*PC34.1.3	Authentication key management (AKM) suite list	8.4.2.27, 11.4.1	PC34.1:M	Yes □ No □ N/A □
PC34.1.3.1	IEEE 802.1X-defined/RSNA key management	8.4.2.27	PC34.1:M	Yes □ No □ N/A □
PC34.1.3.2	Preshared key (PSK)/ RSNA key management	8.4.2.27	PC34.1:M	Yes □ No □ N/A □
PC34.1.3.3	RSNA key management	11.6	PC34.1:M	Yes □ No □ N/A □
PC34.1.3.3.1	Key hierarchy	11.6, 11.7	PC34.1:M	Yes □ No □ N/A □
PC34.1.3.3.1.1	Pairwise key hierarchy	11.6.1.3	PC34.1:M	Yes □ No □ N/A □
PC34.1.3.3.1.2	Group key hierarchy	11.6.1.4	PC34.1:M	Yes □ No □ N/A □
PC34.1.3.3.2	4-Way Handshake	11.6.6	PC34.1:M	Yes □ No □ N/A □
PC34.1.3.3.3	Group Key Handshake	11.6.7	PC34.1:M	Yes □ No □ N/A □
PC34.1.4	RSN capabilities	8.4.2.27, 11.1.3	PC34.1:M	Yes □ No □ N/A □
PC34.1.5	RSNA preauthentication	11.5.9.2	PC34.1:O	Yes □ No □ N/A □
PC34.1.6	RSNA security association management	11.5	PC34.1:M	Yes □ No □ N/A □
PC34.1.7	RSNA pairwise master key security association (PMKSA) caching	11.5.1, 11.5.9.3	PC34.1:M	Yes □ No □ N/A □
PC34.1.8	RSNA extended service set (ESS)	11.5.9, 11.5.12	(PC34.1 and CF1):M	Yes □ No □ N/A □
PC34.1.8.1	RSNA PeerKey Handshake	11.6.8	PC34.1.8:O	Yes □ No □ N/A □
PC34.1.9	RSNA IBSS	11.5.5, 11.5.10, 11.5.13	(PC34.1 and CF2.2):O	Yes □ No □ N/A □

B.4.4.1 MAC protocol capabilities *(continued)*

Item	Protocol capability	References	Status	Support
*PC 34.1.10	Management frame protection	8.4.1.11, 8.5.3, 8.2.4.1.10, 8.4.2.27.4, 11.4.2.1.2, 11.4.2.1.3, 11.4.2.2, 11.2.3.3.5, 11.4.3.3.3, 11.4.3.3.6, 11.4.3.4.2, 11.4.3.4.4, 11.5.3, 11.8.2.3, 11.8.2.5, 11.8.2.7, 11.8.2.9	PC34:O	Yes □ No □ N/A □
*PC 34.1.10.1	BIP	11.4.4, Clause 10	PC34.1.10:M	Yes □ No □ N/A □
PC 34.1.10.1.1	Management MIC element	8.4.2.57	PC34.1.10.1: M	Yes □ No □ N/A □
PC 34.1.11	AKM: IEEE 802.1X authentication with SHA-256 PRF	8.4.2.27, 11.6	PC34:O	Yes □ No □ N/A □
PC 34.1.12	AKM: PSK with SHA-256 PRF	8.4.2.27, 11.6	PC34:O	Yes □ No □ N/A □
*PC35	Fast basic service set (BSS) transition (FT)	Clause 12	CF14:O	Yes □ No □ N/A □
PC35.1	Mobility Domain element (MDE)	8.4.2.49	PC35:M	Yes □ No □ N/A □
PC35.2	Fast basic service set (BSS) Transition element (FTE)	8.4.2.50	PC35&PC34: M	Yes □ No □ N/A □
PC35.3	Timeout Interval element (TIE)	8.4.2.51	PC35:M	Yes □ No □ N/A □
PC35.4	Fast basic service set (BSS) Transition (FT) authentication algorithm	8.4.1.1	PC35:M	Yes □ No □ N/A □
PC35.5	Fast basic service set (BSS) Transition (FT) Action frames	8.5.9	PC35:M	Yes □ No □ N/A □
PC35.6	Fast basic service set (BSS) Transition (FT) key management based on IEEE 802.1X	11.6.1.7, 8.4.2.27	PC35&PC34: M	Yes □ No □ N/A □
PC35.7	Fast basic service set (BSS) Transition (FT) key management based on preshared keys (PSKs)	11.6.1.7, 8.4.2.27	PC35&PC34: M	Yes □ No □ N/A □
PC35.8	Fast basic service set (BSS) Transition (FT) key hierarchy	11.6.1.7	PC35&PC34: M	Yes □ No □ N/A □
PC35.9	FT initial mobility domain association	12.4	PC35&PC34: M	Yes □ No □ N/A □
PC35.10	Fast Basic Service Set (BSS) Transition (FT) Protocol	12.5	PC35:M	Yes □ No □ N/A □
PC35.10.1	Fast Basic Service Set (BSS) Transition (FT) Protocol in robust security network (RSN)	12.5.2, 12.5.3, 12.7.1	PC35&PC34: M	Yes □ No □ N/A □
PC35.10.2	Fast Basic Service Set (BSS) Transition (FT) Protocol in nonrobust security network (non-RSN)	12.5.4, 12.5.5, 12.7.2	PC35:M	Yes □ No □ N/A □

B.4.4.1 MAC protocol capabilities *(continued)*

Item	Protocol capability	References	Status	Support
*PC35.11	Fast Basic Service Set (BSS) Transition (FT) Resource Request Protocol	12.6	PC35:O	Yes □ No □ N/A □
PC35.11.1	Resource Request protocol over the air	12.6.2	PC35.11:M	Yes □ No □ N/A □
PC35.11.2	Resource Request protocol over the distribution system (DS)	12.6.3, 12.10	PC35.11:M	Yes □ No □ N/A □
PC35.12	QoS procedures for fast basic service set (BSS) transition	12.11	CF12& PC35:M	Yes □ No □ N/A □
*PC35.13	Resource Information Container (RIC) Data element (RDE)	12.11, 8.4.2.52	PC35:M	Yes □ No □ N/A □
PC35.13.1	Resource Request Procedures at the fast basic service set (BSS) transition originator (FTO)	12.11.3.1	PC35.13:M	Yes □ No □ N/A □
PC35.13.2	Resource Request Procedures at the target access point (AP)	12.11.3.2	PC35.13:M	Yes □ No □ N/A □
*PC35.14	Remote Request Procedures at the current access point (AP)	12.10	PC35:M	Yes □ No □ N/A □
PC35.14.1	Remote Request/Response frame support	12.10.3	PC35.14:O	Yes □ No □ N/A □
PC35.14.2	Vendor-specific remote request broker (RRB) mechanism	12.10.3	PC35.14:O	Yes □ No □ N/A □
PC36	SA Query Procedure	8.5.10, 10.3	PC34.1.10:M	Yes □ No □ N/A □
*PC37	Power save multi-poll (PSMP)	8.5.12.4, 9.26	O	Yes □ No □
*PC37.1	Scheduled PSMP	8.4.2.32, 10.4.6	PC37:M	Yes □ No □ N/A □
PC37.1.1	PSMP additions to TSPEC	8.4.2.32	PC37.1:M	Yes □ No □ N/A □
PC37.1.2	AP role in scheduled PSMP sequence	9.26.1.2, 9.26.1.3	(PC37.1 and CF1):M	Yes □ No □ N/A □
PC37.1.3	STA role in scheduled PSMP sequence	9.26.1.2, 9.26.1.3	(PC37.1 and CF2):M	Yes □ No □ N/A □
*PC37.2	Unscheduled PSMP	9.26.3	PC37:M	Yes □ No □ N/A □
PC37.2.1	PSMP additions to TSPEC	8.4.2.32	(CF1 and PC37.2):M (CF2 and PC37.2):O	Yes □ No □ N/A □
PC37.3	Creation, scheduling, and transmission of PSMP frame	8.5.12.4, 9.26.1.1	(PC37 and CF1):M	Yes □ No □ N/A □
PC37.4	Reception and interpretation of PSMP frame	8.5.12.4	(PC37 and CF2):M	Yes □ No □ N/A □
PC37.5	Multi-TID Block Ack rules in PSMP sequence	8.3.1.8.4, 8.3.1.9.4, 9.26.1.7, 10.16.2	PC37: M	Yes □ No □ N/A □
PC37.6	Multi-phase PSMP	9.26.1.5	PC37:M	Yes □ No □ N/A □
PC38	dot11OCBActivated is false when STA is a BSS member	10.20	(CF2.1 or CF2.2): M	Yes □ No □ N/A □
PC39	Simultaneous authentication of equals (SAE)	11.3	CF21:M	Yes □ No □

B.4.4.2 MAC frames

Item	MAC frame	References	Status	Support
	Is transmission of the following MAC frames supported?	Clause 8, Annex J		
FT1	Association request	Clause 8	CF2.1:M	Yes □ No □ N/A □
FT2	Association response	Clause 8	CF1:M	Yes □ No □ N/A □
FT3	Reassociation request	Clause 8	CF2.1:M	Yes □ No □ N/A □
FT4	Reassociation response	Clause 8	CF1:M	Yes □ No □ N/A □
FT5	Probe request	Clause 8	(CF2.1 or CF2.2):M	Yes □ No □ N/A □
FT6	Probe response	Clause 8	(not CF2.3):M	Yes □ No □ N/A □
FT7	Beacon	Clause 8	(not CF2.3):M	Yes □ No □ N/A □
FT8	ATIM	Clause 8	(CF2.1 or CF2.2):M	Yes □ No □ N/A □
FT9	Disassociation	Clause 8	(not CF2.3): M	Yes □ No □ N/A □
FT10	Authentication	Clause 8	(not CF2.3): M	Yes □ No □ N/A □
FT11	Deauthentication	Clause 8	(not CF2.3): M	Yes □ No □ N/A □
FT12	Power save (PS)-Poll	Clause 8	CF2.1:M	Yes □ No □ N/A □
FT13	RTS	Clause 8	M	Yes □ No □
FT14	CTS	Clause 8	M	Yes □ No □
FT15	Acknowledgment (ACK)	Clause 8	M	Yes □ No □
FT16	CF-End	Clause 8	PC4:M	Yes □ No □ N/A □
FT17	CF End+CF-Ack	Clause 8	PC4:M	Yes □ No □ N/A □
FT18	Data	Clause 8	M	Yes □ No □
FT19	Data + CF-Ack	Clause 8	(PC4 OR PC5):M	Yes □ No □ N/A □
FT20	Data + CF-Poll	Clause 8	PC4.3:M	Yes □ No □ N/A □
FT21	Data + CF-Ack+CF-Poll	Clause 8	PC4.3:M	Yes □ No □ N/A □
FT22	Null	Clause 8	M	Yes □ No □
FT23	CF-Ack (no data)	Clause 8	(PC4 OR PC5):M	Yes □ No □ N/A □
FT24	CF-Poll (no data)	Clause 8	PC4.3:M	Yes □ No □ N/A □
FT25	CF-Ack+CF-Poll (no data)	Clause 8	PC4.3:M	Yes □ No □ N/A □
FT26	Timing Advertisement frame	Clause 8	O	Yes □ No □ N/A □
	Is reception of the following MAC frames supported?	Clause 8, Annex J		
FR1	Association request	Clause 8	CF1:M	Yes □ No □ N/A □
FR2	Association response	Clause 8	CF2.1:M	Yes □ No □ N/A □
FR3	Reassociation request	Clause 8	CF1:M	Yes □ No □ N/A □
FR4	Reassociation response	Clause 8	CF2.1:M	Yes □ No □ N/A □
FR5	Probe request	Clause 8	(not CF2.3): M	Yes □ No □ N/A □

B.4.4.2 MAC frames *(continued)*

Item	MAC frame	References	Status	Support
FR6	Probe response	Clause 8	(not CF2.3): M	Yes □ No □ N/A □
FR7	Beacon	Clause 8	(not CF2.3): M	Yes □ No □ N/A □
FR8	ATIM	Clause 8	CF2.2:M	Yes □ No □ N/A □
FR9	Disassociation	Clause 8	(not CF2.3): M	Yes □ No □ N/A □
FR10	Authentication	Clause 8	(not CF2.3): M	Yes □ No □ N/A □
FR11	Deauthentication	Clause 8	(not CF2.3): M	Yes □ No □ N/A □
FR12	PS-Poll	Clause 8	CF1:M	Yes □ No □ N/A □
FR13	RTS	Clause 8	M	Yes □ No □
FR14	CTS	Clause 8	M	Yes □ No □
FR15	ACK	Clause 8	M	Yes □ No □
FR16	CF-End	Clause 8	(not CF2.3): M	Yes □ No □ N/A □
FR17	CF End+CF-Ack	Clause 8	(not CF2.3): M	Yes □ No □ N/A □
FR18	Data	Clause 8	M	Yes □ No □
FR19	Data + CF-Ack	Clause 8	(not CF2.3): M	Yes □ No □ N/A □
FR20	Data + CF-Poll	Clause 8	PC5:M	Yes □ No □ N/A □
FR21	Data + CF-Ack+CF-Poll	Clause 8	PC5:M	Yes □ No □ N/A □
FR22	Null	Clause 8	M	Yes □ No □
FR23	CF-Ack (no data)	Clause 8	(PC4 OR PC5):M	Yes □ No □ N/A □
FR24	CF-Poll (no data)	Clause 8	PC5:M	Yes □ No □ N/A □
FR25	CF-Ack+CF-Poll (no data)	Clause 8	PC5:M	Yes □ No □ N/A □
FR26	Timing Advertisement frame	Clause 8	O	Yes □ No □ N/A □

B.4.4.3 Frame exchange sequences

Item	Frame exchange sequence	References	Status	Support
	Are the following frame sequences supported?			
FS1	Basic frame sequences	9.3.2.5, 9.3.2.6, 9.3.5, 9.3.6, 9.3.2.8, 9.4.3	M	Yes □ No □
FS2	CF-Frame sequences	9.4.3, 9.4.4	(PC4 OR PC5):M	Yes □ No □ N/A □

B.4.4.4 MAC addressing functions

Item	MAC Address function	References	Status	Support
	Are the following MAC Addressing functions supported?			
AD1	STA universal individual IEEE 802 address	8.2.4.3	M	Yes □ No □
AD2	BSS identification (BSSID) generation	8.2.4.3, 10.1.4, Annex J	M	Yes □ No □
AD3	Receive address matching	8.2.4.3, 8.3.2.1, Annex J	M	Yes □ No □
AD4	Wildcard BSSID	8.2.4.3.4, 8.3.2	CF2.3:M	Yes □ No □ N/A □
AD5	MAC and PHY operation resumes with appropriate MIB attributes in less than 2 TU	10.20	CF2.3:M	Yes □ No □ N/A □
AD6	Group addressed Mesh Data frame addressing (3 address frame)	8.2.3, 8.2.4.1, 8.2.4.3, 9.32.3	CF21:M	Yes □ No □ N/A □
AD7	Individually addressed Mesh Data frame addressing (4 address frame)	8.2.3, 8.2.4.1, 8.2.4.3, 9.32.3	CF21:M	Yes □ No □ N/A □
AD8	Proxied group addressed Mesh Data frame addressing (4 address frame)	8.2.3, 8.2.4.1, 8.2.4.3, 8.2.4.7.3, 9.32.3	CF21:M	Yes □ No □ N/A □
AD9	Proxied individually addressed Mesh Data frame addressing (6 address frame)	8.2.3, 8.2.4.1, 8.2.4.3, 8.2.4.7.3, 9.32.3	CF21:M	Yes □ No □ N/A □
AD10	Multihop Action frame addressing (4 address frame)	8.2.3, 8.2.4.1, 8.2.4.3, 8.2.4.7.3, 8.5.18, 9.32.3	CF21:M	Yes □ No □ N/A □
AD11	TA filtering for mesh STA	8.2.4.3, 8.3.2.1, 9.32.3	CF21:M	Yes □ No □ N/A □

B.4.5 Frequency hopping (FH) PHY functions

Item	Protocol feature	References	Status	Support
	Which requirements and options does the PHY support?			
FH1	PHY service primitive parameters			
FH1.1	TXVECTOR parameter: LENGTH	14.3.2.2	M	Yes □ No □
FH1.2	TXVECTOR parameter: PLCPBITRATE	14.3.2.3	M	Yes □ No □
FH1.2.1	PLCPBITRATE = X'00' (1.0 Mb/s)	14.3.2.3	M	Yes □ No □
* FH1.2.2	PLCPBITRATE = X'02' (2.0 Mb/s)	14.3.2.3	O	Yes □ No □
FH1.3	RXVECTOR parameter: LENGTH	14.3.3.2	M	Yes □ No □
FH1.4	RXVECTOR parameter: Receive signal strength indicator (RSSI)	14.3.3.3	O	Yes □ No □

B.4.5 Frequency hopping (FH) PHY functions *(continued)*

Item	Protocol feature	References	Status	Support
FH2	Physical layer convergence procedure (PLCP) frame format			
FH2.1	PLCP preamble: Sync	14.4.3.2.2	M	Yes □ No □
FH2.2	PLCP preamble: Start frame delimiter (SFD)	14.4.3.2.3	M	Yes □ No □
FH2.3	PLCP header: PSDU length word (PLW)	14.4.3.3.2	M	Yes □ No □
FH2.4	PLCP header: PLCP Signaling field (PSF)	14.4.3.3.3	M	Yes □ No □
FH2.5	PLCP header: Header error check (HEC)	14.4.3.3.4	M	Yes □ No □
FH2.6	PLCP data whitener: Scrambling and bias suppression encoding	14.4.3.4, 14.4.4.2.2	M	Yes □ No □
FH3	Transmit PLCP			
FH3.1	Transmit: transmit on MAC request	14.4.4.2.2	M	Yes □ No □
FH3.2	Transmit: format and whiten frame	14.4.4.2.2	M	Yes □ No □
FH3.3	Transmit: Timing	14.4.4.2.2	M	Yes □ No □
FH4	Carrier sense (CS)/clear channel assessment (CCA) procedure			
FH4.1	CS/CCA: perform on a minimum of one antenna	14.4.4.3.2	M	Yes □ No □
FH4.2.	CS/CCA: Detect preamble starting up to 20 µs after start of slot time	14.4.4.3.2	M	Yes □ No □
FH4.3	CS/CCA: Detect preamble starting at least 16 µs prior to end of slot time	14.4.4.3.2	M	Yes □ No □
FH4.4	CS/CCA: Detect random data	14.4.4.3.2	M	Yes □ No □
FH4.5	CS/CCA: Perform on antenna with essentially same gain and pattern as transmit antenna	14.4.4.3.2	M	Yes □ No □
FH4.6	CS/CCA: Detect valid SFD and PLCP header	14.4.4.3.2	M	Yes □ No □
FH4.7	CS/CCA: Maintain BUSY indication until end of length contained in valid PLCP header	14.4.4.3.2	M	Yes □ No □
FH5	Receive PLCP			
FH5.1	Receive: Receive and dewhiten frame	14.4.4.4.2	M	Yes □ No □
FH6	Physical layer management entity (PLME)			
FH6.1	PLME: Support FH sync	14.5.2.2	M	Yes □ No □
FH6.2	PLME: Support PLME primitives	14.5.3.2	O	Yes □ No □
FH7	Geographic area specific requirements			
* FH7.1	Geographic areas			
FH7.1.1	North America	14.7.2	O.1	Yes □ No □
FH7.1.2	Most of Europe	14.7.2	O.1	Yes □ No □
FH7.1.3	Japan	14.7.2	O.1	Yes □ No □
FH7.1.4	Spain	14.7.2	O.1	Yes □ No □

B.4.5 Frequency hopping (FH) PHY functions *(continued)*

Item	Protocol feature	References	Status	Support
FH7.1.5	France	14.7.2	O.1	Yes □ No □
FH7.1.6	China	14.7.2	O.1	Yes □ No □
FH7.2	Operating frequency range	14.7.3	FH7.1:M	Yes □ No □
FH7.3	Number of operating channels	14.7.4	FH7.1:M	Yes □ No □
FH7.4	Operating channel frequencies	14.7.5	FH7.1:M	Yes □ No □
FH7.5	Occupied channel bandwidth	14.7.6	FH7.1:M	Yes □ No □
FH7.6	Minimum hop rate	14.7.7	FH7.1:M	Yes □ No □
FH7.7	Hop sequences	14.7.8	FH7.1:M	Yes □ No □
FH7.8	Unwanted emissions	14.7.9	FH7.1:M	Yes □ No □
FH8	1 Mb/s physical medium dependent (PMD)			
FH8.1	Modulation 2GFSK, bit time (BT) = 0.5, 1 = positive frequency deviation, 0 = negative frequency deviation	14.7.10	M	Yes □ No □
FH8.2	Peak frequency deviation	14.7.10	M	Yes □ No □
FH8.3	Zero-Crossing error	14.7.10	M	Yes □ No □
FH8.4	Nominal channel data rate	14.7.11	M	Yes □ No □
FH8.5	Channel switching/settling time	14.7.12	M	Yes □ No □
FH8.6	Receive to transmit switch time	14.7.13	M	Yes □ No □
FH8.7	Nominal transmit power	14.7.14.2	M	Yes □ No □
FH8.8	Transmit power levels	14.7.14.3	M	Yes □ No □
FH8.9	Transmit power level control to < 100 mW	14.7.14.4	M	Yes □ No □
FH8.10	Transmit spectrum shape	14.7.14.5	M	Yes □ No □
FH8.11	Transmit center frequency tolerance	14.7.14.6	M	Yes □ No □
FH8.12	Transmitter ramp periods	14.7.14.7	M	Yes □ No □
FH8.13	Receiver input dynamic range	14.7.15.2	M	Yes □ No □
FH8.14	Receiver center frequency acceptance range	14.7.15.3	M	Yes □ No □
FH8.15	CCA power threshold for a probability of detection of 90% (preamble)/70% (random data) for 100 mW units	14.7.15.4	M	Yes □ No □
FH8.16	CCA power threshold for units > 100 mW; sensitivity threshold is 1/2 dB lower for every dB above 20 dBm	14.7.15.4	M	Yes □ No □
FH8.17	Minimum receiver sensitivity at frame error ratio (FER) = 3% with 400 octet frames	14.7.15.5	M	Yes □ No □
FH8.18	Intermodulation protection (IMp)	14.7.15.6	M	Yes □ No □
FH8.19	Desensitization (Dp)	14.7.15.7	M	Yes □ No □
FH9	2 Mb/s PMD			
FH9.1	All 1M PMD requirements	14.8.1	FH1.2.2:M	Yes □ No □ N/A □
FH9.2	Modulation 4GFSK, BT = 0.5	14.8.2	FH1.2.2:M	Yes □ No □ N/A □

B.4.5 Frequency hopping (FH) PHY functions *(continued)*

Item	Protocol feature	References	Status	Support
FH9.3	Frame structure for 2M PHY	14.8.3	FH1.2.2:M	Yes □ No □ N/A □
FH9.4	Nominal channel data rate	14.8.4	FH1.2.2:M	Yes □ No □ N/A □
FH9.5	Input dynamic range	14.8.5	FH1.2.2:M	Yes □ No □ N/A □
FH9.6	Minimum receiver sensitivity at FER = 3% with 400 octet frames	14.8.6	FH1.2.2:M	Yes □ No □ N/A □
FH9.7	IMp	14.8.7	FH1.2.2:M	Yes □ No □ N/A □
FH9.8	Dp	14.8.8	FH1.2.2:M	Yes □ No □ N/A □
FH10	MIB	14.9, Annex C	M	Yes □ No □
FH10.1	dot11PhyFHSSComplianceGroup, dot11PhyRegDomainsSupportGroup, and dot11PhyOperationComplianceGroup	14.9	M	Yes □ No □

B.4.6 Direct sequence PHY functions

Item	PHY feature	References	Status	Support
	PLCP sublayer procedures	16.2		
DS1	Preamble prepend on transmit (TX)	16.2.1	M	Yes □ No □
DS1.1	PLCP frame format	16.2.2, 16.2.3	M	Yes □ No □
DS1.2	PLCP integrity check generation	16.2.3, 16.2.3.7	M	Yes □ No □
DS1.3	TX rate change capability	16.2.3.4, 16.2.5	M	Yes □ No □
DS1.4	Supported data rates	16.1, 16.2.3.4	M	Yes □ No □
DS1.5	Data whitener scrambler	16.2.4	M	Yes □ No □
DS1.6	Scrambler initialization	16.2.4	M	Yes □ No □
DS2	Preamble process on receive (RX)	16.2.1		
DS2.1	PLCP frame format	16.2.2, 16.2.3	M	Yes □ No □
DS2.2	PLCP integrity check verify	16.2.3, 16.2.3.7	M	Yes □ No □
DS2.3	RX Rate change capability	16.2.3.4, 16.2.5	M	Yes □ No □
DS2.4	Data whitener descrambler	16.2.4	M	Yes □ No □
DS3	Pseudonoise (PN) code sequence	16.4.6.4	M	Yes □ No □
DS4	Chipping continue on power-down	16.2.6	O	Yes □ No □
*DS5	Operating channel capability	16.2.6, 16.4.6.3		
* DS5.1	North America (FCC)	16.2.6, 16.4.6.3	DS5:O.1	Yes □ No □ N/A □
DS5.1.1	Channel 1	16.2.6, 16.4.6.3	DS5.1:M	Yes □ No □ N/A □
DS5.1.2	Channel 2	16.2.6, 16.4.6.3	DS5.1:M	Yes □ No □ N/A □
DS5.1.3	Channel 3	16.2.6, 16.4.6.3	DS5.1:M	Yes □ No □ N/A □
DS5.1.4	Channel 4	16.2.6, 16.4.6.3	DS5.1:M	Yes □ No □ N/A □
DS5.1.5	Channel 5	16.2.6, 16.4.6.3	DS5.1:M	Yes □ No □ N/A □
DS5.1.6	Channel 6	16.2.6, 16.4.6.3	DS5.1:M	Yes □ No □ N/A □

B.4.6 Direct sequence PHY functions *(continued)*

Item	PHY feature	References	Status	Support
DS5.1.7	Channel 7	16.2.6, 16.4.6.3	DS5.1:M	Yes □ No □ N/A □
DS5.1.8	Channel 8	16.2.6, 16.4.6.3	DS5.1:M	Yes □ No □ N/A □
DS5.1.9	Channel 9	16.2.6, 16.4.6.3	DS5.1:M	Yes □ No □ N/A □
DS5.1.10	Channel 10	16.2.6, 16.4.6.3	DS5.1:M	Yes □ No □ N/A □
DS5.1.11	Channel 11	16.2.6, 16.4.6.3	DS5.1:M	Yes □ No □ N/A □
* DS5.2	Canada (IC)	16.2.6, 16.4.6.3	DS5:O.1	Yes □ No □ N/A □
DS5.2.1	Channel 1	16.2.6, 16.4.6.3	DS5.2:M	Yes □ No □ N/A □
DS5.2.2	Channel 2	16.2.6, 16.4.6.3	DS5.2:M	Yes □ No □ N/A □
DS5.2.3	Channel 3	16.2.6, 16.4.6.3	DS5.2:M	Yes □ No □ N/A □
DS5.2.4	Channel 4	16.2.6, 16.4.6.3	DS5.2:M	Yes □ No □ N/A □
DS5.2.5	Channel 5	16.2.6, 16.4.6.3	DS5.2:M	Yes □ No □ N/A □
DS5.2.6	Channel 6	16.2.6, 16.4.6.3	DS5.2:M	Yes □ No □ N/A □
DS5.2.7	Channel 7	16.2.6, 16.4.6.3	DS5.2:M	Yes □ No □ N/A □
DS5.2.8	Channel 8	16.2.6, 16.4.6.3	DS5.2:M	Yes □ No □ N/A □
DS5.2.9	Channel 9	16.2.6, 16.4.6.3	DS5.2:M	Yes □ No □ N/A □
DS5.2.10	Channel 10	16.2.6, 16.4.6.3	DS5.2:M	Yes □ No □ N/A □
DS5.2.11	Channel 11	16.2.6, 16.4.6.3	DS5.2:M	Yes □ No □ N/A □
* DS5.3	Europe (ETSI)	16.2.6, 16.4.6.3	DS5:O.1	Yes □ No □ N/A □
DS5.3.1	Channel 1	16.2.6, 16.4.6.3	DS5.3:M	Yes □ No □ N/A □
DS5.3.2	Channel 2	16.2.6, 16.4.6.3	DS5.3:M	Yes □ No □ N/A □
DS5.3.3	Channel 3	16.2.6, 16.4.6.3	DS5.3:M	Yes □ No □ N/A □
DS5.3.4	Channel 4	16.2.6, 16.4.6.3	DS5.3:M	Yes □ No □ N/A □
DS5.3.5	Channel 5	16.2.6, 16.4.6.3	DS5.3:M	Yes □ No □ N/A □
DS5.3.6	Channel 6	16.2.6, 16.4.6.3	DS5.3:M	Yes □ No □ N/A □
DS5.3.7	Channel 7	16.2.6, 16.4.6.3	DS5.3:M	Yes □ No □ N/A □
DS5.3.8	Channel 8	16.2.6, 16.4.6.3	DS5.3:M	Yes □ No □ N/A □
DS5.3.9	Channel 9	16.2.6, 16.4.6.3	DS5.3:M	Yes □ No □ N/A □
DS5.3.10	Channel 10	16.2.6, 16.4.6.3	DS5.3:M	Yes □ No □ N/A □
DS5.3.11	Channel 11	16.2.6, 16.4.6.3	DS5.3:M	Yes □ No □ N/A □
DS5.3.12	Channel 12	16.2.6, 16.4.6.3	DS5.3:M	Yes □ No □ N/A □
DS5.3.13	Channel 13	16.2.6, 16.4.6.3	DS5.3:M	Yes □ No □ N/A □
* DS5.4	France	16.2.6, 16.4.6.3	DS5:O.1	Yes □ No □ N/A □
DS5.4.1	Channel 10	16.2.6, 16.4.6.3	DS5.4:M	Yes □ No □ N/A □
DS5.4.2	Channel 11	16.2.6, 16.4.6.3	DS5.4:M	Yes □ No □ N/A □
DS5.4.3	Channel 12	16.2.6, 16.4.6.3	DS5.4:M	Yes □ No □ N/A □
DS5.4.4	Channel 13	16.2.6, 16.4.6.3	DS5.4:M	Yes □ No □ N/A □
* DS5.5	Spain	16.2.6, 16.4.6.3	DS5:O.1	Yes □ No □ N/A □
DS5.5.1	Channel 10	16.2.6, 16.4.6.3	DS5.5:M	Yes □ No □ N/A □
DS5.5.2	Channel 11	16.2.6, 16.4.6.3	DS5.5:M	Yes □ No □ N/A □
* DS5.6	Japan	16.2.6, 16.4.6.3	DS5:O.1	Yes □ No □ N/A □

B.4.6 Direct sequence PHY functions *(continued)*

Item	PHY feature	References	Status	Support
* DS5.7	China	16.2.6, 16.4.6.3	DS5:O.1	Yes □ No □ N/A □
DS5.7.1	Channel 1	16.2.6, 16.4.6.3	DS5.7:M	Yes □ No □ N/A □
DS5.7.2	Channel 2	16.2.6, 16.4.6.3	DS5.7:M	Yes □ No □ N/A □
DS5.7.3	Channel 3	16.2.6, 16.4.6.3	DS5.7:M	Yes □ No □ N/A □
DS5.7.4	Channel 4	16.2.6, 16.4.6.3	DS5.7:M	Yes □ No □ N/A □
DS5.7.5	Channel 5	16.2.6, 16.4.6.3	DS5.7:M	Yes □ No □ N/A □
DS5.7.6	Channel 6	16.2.6, 16.4.6.3	DS5.7:M	Yes □ No □ N/A □
DS5.7.7	Channel 7	16.2.6, 16.4.6.3	DS5.7:M	Yes □ No □ N/A □
DS5.7.8	Channel 8	16.2.6, 16.4.6.3	DS5.7:M	Yes □ No □ N/A □
DS5.7.9	Channel 9	16.2.6, 16.4.6.3	DS5.7:M	Yes □ No □ N/A □
DS5.7.10	Channel 10	16.2.6, 16.4.6.3	DS5.7:M	Yes □ No □ N/A □
DS5.7.11	Channel 11	16.2.6, 16.4.6.3	DS5.7:M	Yes □ No □ N/A □
DS5.7.12	Channel 12	16.2.6, 16.4.6.3	DS5.7:M	Yes □ No □ N/A □
DS5.7.13	Channel 13	16.2.6, 16.4.6.3	DS5.7:M	Yes □ No □ N/A □
DS6	Bits to symbol mapping	16.4.6.5		
DS6.1	1 Mb/s	16.4.6.5	M	Yes □ No □
DS6.2	2 Mb/s	16.4.6.5	M	Yes □ No □
*DS7	CCA functionality	16.4.8.5		
DS7.1	Energy Only (RSSI above threshold)	16.4.8.5	DS7:O.2	Yes □ No □ N/A □
DS7.2	IEEE 802.11 DSSS correlation	16.4.8.5	DS7:O.2	Yes □ No □ N/A □
DS7.3	Both methods	16.4.8.5	DS7:O.2	Yes □ No □ N/A □
DS7.4	Hold CCA busy for packet duration of a correctly received PLCP but carrier lost during reception of MPDU	16.2.7	M	Yes □ No □
DS7.5	Hold CCA busy for packet duration of a correctly received but out of specification PLCP	16.2.7	M	Yes □ No □
DS8	Transmit antenna selection	16.4.5.6, 16.4.5.7	O	Yes □ No □
DS9	Receive antenna diversity	16.4.5.6, 16.4.5.7, 16.4.5.8	O	Yes □ No □
*DS10	Antenna port(s) availability	16.4.6.10	O	Yes □ No □
DS10.1	50 ¾ impedance	16.4.6.10	DS10:M	Yes □ No □ N/A □
*DS11	Transmit power level support	16.4.5.9, 16.4.7.4	O	Yes □ No □
DS11.1	If greater than 100 mW capability	16.4.7.4	DS11:M	Yes □ No □ N/A □
DS12	Spurious emissions conformance	16.4.6.6	M	Yes □ No □
DS13	TX-to-RX turnaround time	16.4.6.7	M	Yes □ No □
DS14	RX-to-TX turnaround time	16.4.6.8	M	Yes □ No □
DS15	Slot time	16.4.6.9	M	Yes □ No □
DS16	Energy detection (ED) reporting time	16.4.6.9, 16.4.8.5	M	Yes □ No □

B.4.6 Direct sequence PHY functions *(continued)*

Item	PHY feature	References	Status	Support
DS17	Minimum transmit power level	16.4.7.3	M	Yes □ No □
DS18	Transmit spectral mask conformance	16.4.7.5	M	Yes □ No □
DS19	Transmitted center frequency tolerance	16.4.7.6	M	Yes □ No □
DS20	Chip clock frequency tolerance	16.4.7.7	M	Yes □ No □
DS21	Transmit power-on ramp	16.4.7.8	M	Yes □ No □
DS22	Transmit power-down ramp	16.4.7.8	M	Yes □ No □
DS23	Radio frequency (RF) carrier suppression	16.4.7.9	M	Yes □ No □
DS24	Transmit modulation accuracy	16.4.7.10	M	Yes □ No □
DS25	Receiver minimum input level sensitivity	16.4.8.2	M	Yes □ No □
DS26	Receiver maximum input level	16.4.8.3	M	Yes □ No □
DS27	Receiver adjacent channel rejection	16.4.8.4	M	Yes □ No □
DS28	MIB	16.3.2, Annex C	M	Yes □ No □
DS28.1	dot11PhyDSSSComplianceGroup, dot11PhyRegDomainsSupportGroup, and dot11PhyOperationComplianceGroup	16.3.2	M	Yes □ No □

B.4.7 IR baseband PHY functions

Item	Feature	References	Status	Support
IR1	Is the transmitted synchronization (SYNC) field length in the range of required number of pulse position modulation (PPM) slots, with the absence of a pulse in the last slot of the field?	15.3.5.1	M	Yes □ No □
IR2	Is the transmitted SYNC field entirely populated by alternating presence and absence of pulses in consecutive PPM slots, with the absence of a pulse in the last slot of the field?	15.3.5.1	M	Yes □ No □
IR3	Is the transmitted SFD field the binary sequence 1001, where 1 indicates a pulse in the PPM slot and 0 indicates no pulse in the PPM slot?	15.3.5.2	M	Yes □ No □
IR4	Is the transmitted data rate (DR) field pulse sequence equal to the correct value for the data rate provided by the TXVECTOR parameter PLCP BITRATE, where 1 indicates a pulse in the PPM slot and 0 indicates no pulse in the PPM slot?	15.3.5.3	M	Yes □ No □

B.4.7 IR baseband PHY functions *(continued)*

Item	Feature	References	Status	Support
IR5	Is the transmitted dc level adjustment (DCLA) field 32 PPM slots long with the specified sequence for 1 Mb/s, where 1 indicates a pulse in the PPM slot and 0 indicates no pulse in the PPM slot? 1 Mb/s: 00000000100000000000000010000000	15.3.5.4	M	Yes □ No □
* IR5a	Does the unit support 2 Mb/s transmission?	15.3.5.4	O	Yes □ No □
IR5b	If the unit supports 2 Mb/s transmission, is the transmitted DCLA field 32 PPM slots long with the specified sequence for 2 Mb/s, where 1 indicates a pulse in the PPM slot and 0 indicates no pulse in the PPM slot? 2 Mb/s: 00100010001000100010001000100010	15.3.5.4	IR5a:M	Yes □ No □ N/A □
IR6	Is the transmitted LENGTH field the correct PPM representation of the unsigned 16-bit binary integer, least significant bit (LSB) transmitted first, equal to the correct value provided by the TXVECTOR parameter LENGTH?	15.3.5.5	M	Yes □ No □
IR7	Is the transmitted cyclic redundancy code (CRC) field the correct PPM representation of the CRC value calculated according to the reference subclause, transmitted LSB first?	15.3.5.6	M	Yes □ No □
IR8	Is the transmitted PLCP service data unit (PSDU) field the correct PPM representation of the PSDU, transmitted LSB first?	15.3.5.7	M	Yes □ No □
IR9	When the CCA is false does transmission begin based on a PHY-TXSTART.request primitive?	15.3.6.1	M	Yes □ No □
IR10	Does the PHY issue a PHY-TXSTART.confirm primitive after the transmission of the PLCP header?	15.3.6.1	M	Yes □ No □
IR11	Does the PHY accept each octet of the PSDU in a PHY-DATA.request primitive and answer with a PHYDATA.confirm primitive?	15.3.6.1	M	Yes □ No □
IR12	Does the PHY cease transmission in response to a PHY-TXEND.request primitive and answer with a PHY-TXEND.confirm primitive?	15.3.6.1	M	Yes □ No □
IR13	Does the PHY of a receiving STA send a PHY-CCA.indication primitive during reception of the SYNC field?	15.3.6.2	M	Yes □ No □
IR14	Does the PHY of a receiving STA properly receive a transmission that changes data rate according to the DR field?	15.3.6.2	M	Yes □ No □
IR15	Does the PHY of a receiving STA properly reject an incorrect CRC?	15.3.6.2	M	Yes □ No □
IR16	Does the PHY of a receiving STA properly reject a DR field other than those specified in reference subclause?	15.3.6.2, 15.3.5.3	M	Yes □ No □

B.4.7 IR baseband PHY functions *(continued)*

Item	Feature	References	Status	Support
IR17	Does the PHY of a receiving STA send PHY-RXSTART.indication primitive with correct RATE and LENGTH parameters after proper reception of PLCP preamble and PLCP header?	15.3.6.2	M	Yes □ No □
IR18	Does the PHY of a receiving STA forward receive octets in PHY-DATA.indication primitives?	15.3.6.2	M	Yes □ No □
IR19	Does the PHY of a receiving STA send a PHY-RXEND.indication primitive after the final octet indicated by the LENGTH field?	15.3.6.2	M	Yes □ No □
IR20	Does the PHY of a receiving STA send a PHY-CCA.indication primitive with a state value of IDLE after the PHY-RXEND.indication primitive?	15.3.6.2	M	Yes □ No □
IR21	Does the PHY reset its CCA detection mechanism upon receiving a PHY-CCARST.request primitive, and respond with a PHY-CCARST.indication primitive?	15.3.6.3	M	Yes □ No □
IR22	When transmitting at 1 Mb/s does the PHY transmit PPM symbols according to the 16-PPM Basic Rate Mapping table, transmitting from left to right?	15.4.3.2, 15.4.3.3	M	Yes □ No □
IR23	When transmitting at 2 Mb/s does the PHY transmit PPM symbols according to the 4-PPM Enhanced Rate Mapping table, transmitting from left to right?	15.4.3.2, 15.4.3.3	IR5a:M	Yes □ No □
* IR24	If the unit is conformant to emitter radiation mask 1, is the peak optical power of an emitted pulse within the specification range averaged over the pulse width?	15.4.4.2	O.1	Yes □ No □
* IR25	If the unit is conformant to emitter radiation mask 2, is the peak optical power of an emitted pulse within the specification range averaged over the pulse width?	15.4.4.2	O.1	Yes □ No □
IR26	Does the transmitted pulse shape conform to the description of the reference subclause?	15.4.4.3	M	Yes □ No □
IR27	Does the emitter radiation pattern as a function of angle conform to the requirements of the reference subclause as applicable based on conformance to emitter radiation Mask 1?	15.4.4.4	IR24:M	Yes □ No □ N/A □
IR27a	Does the emitter radiation pattern as a function of angle conform to the requirements of the reference subclause as applicable based on conformance to emitter radiation Mask 2?	15.4.4.4	IR25:M	Yes □ No □ N/A □
IR28	Is the peak emitter optical output as a function of wavelength in the range specified?	15.4.4.5	M	Yes □ No □
IR29	Does the spectrum of the transmit signal amplitude as a voltage or current meet the requirements of the reference subclause?	15.4.4.6	M	Yes □ No □

B.4.7 IR baseband PHY functions *(continued)*

Item	Feature	References	Status	Support
IR30	Does the receiver sensitivity meet the requirements of the reference subclause for receive signals of both 1 Mb/s and 2 Mb/s?	15.4.5.2	M	Yes □ No □
IR31	Does the receiver exhibit a dynamic range as specified in reference subclause?	15.4.5.3	M	Yes □ No □
IR32	Does the receiver field of view (FOV) conform to the requirements of the reference subclause?	15.4.5.4	M	Yes □ No □
IR33	When it is known that the conditions are such that the carrier detect signal and the ED signal are false, is the CCA asserted IDLE?	15.4.6.1	M	Yes □ No □
IR34	When the conditions are such that ED is true for greater than the time defined in reference subclause, does CCA become IDLE?	15.4.6.1	M	Yes □ No □
IR35	When conditions are such that either carrier detect or ED go true, does CCA go BUSY?	15.4.6.1	M	Yes □ No □
IR36	Are these compliance groups implemented? dot11PhyIRComplianceGroup, dot11PhyRegDomainsSupportGroup, and dot11PhyOperationComplianceGroup	15.5	M	Yes □ No □

B.4.8 OFDM PHY functions

Item	Feature	References	Status	Support
OF1: OFDM PHY Specific Service Parameters				
OF1.1	TXVECTOR parameter: LENGTH	18.2.2.2	M	Yes □ No □
OF1.2	TXVECTOR parameter: DATARATE	18.2.2.3	M	Yes □ No □
OF1.2.1	DATARATE = 6.0 Mb/s	18.2.2.3	M	Yes □ No □
*OF1.2.2	DATARATE = 9.0 Mb/s	18.2.2.3	O	Yes □ No □
OF1.2.3	DATARATE = 12.0 Mb/s	18.2.2.3	M	Yes □ No □
*OF1.2.4	DATARATE = 18.0 Mb/s	18.2.2.3	O	Yes □ No □
*OF1.2.5	DATARATE = 24.0 Mb/s, optional in U.S. 3.65-3.70 GHz band, mandatory elsewhere	18.2.2.3, Annex E	(NOT CF15):M, CF15:O	Yes □ No □ N/A □
*OF1.2.6	DATARATE = 36.0 Mb/s	18.2.2.3	O	Yes □ No □
*OF1.2.7	DATARATE = 48.0 Mb/s	18.2.2.3	O	Yes □ No □
*OF1.2.8	DATARATE = 54.0 Mb/s	18.2.2.3	O	Yes □ No □
OF1.3	TXVECTOR parameter: SERVICE	18.2.2.4	M	Yes □ No □
OF1.4	TXVECTOR parameter: TXPWR_LEVEL	18.2.2.5	M	Yes □ No □
OF1.5	RXVECTOR parameter: LENGTH	18.2.3.2	M	Yes □ No □
OF1.6	RXVECTOR parameter: RSSI	18.2.3.3	M	Yes □ No □

B.4.8 OFDM PHY functions *(continued)*

Item	Feature	References	Status	Support
*OF1.7	10 MHz Channel spacing	18.2.2, 18.2.3, 18.2.3.4, Annex E	CF11:O, CF15&DSE2:M	Yes □ No □ N/A □
*OF1.7.1	DATARATE = 3 Mb/s (10 MHz channel spacing)	18.2.2, 18.2.3, 18.2.3.4	CF11& OF1.7:M	Yes □ No □ N/A □
*OF1.7.2	DATARATE = 4.5 Mb/s (10 MHz channel spacing)	18.2.2, 18.2.3, 18.2.3.4	CF11& OF1.7:O	Yes □ No □ N/A □
*OF1.7.3	DATARATE = 6 Mb/s (10 MHz channel spacing)	18.2.2, 18.2.3, 18.2.3.4	CF11& OF1.7:M	Yes □ No □ N/A □
*OF1.7.4	DATARATE = 9 Mb/s (10 MHz channel spacing)	18.2.2, 18.2.3, 18.2.3.4	CF11& OF1.7:O	Yes □ No □ N/A □
*OF1.7.5	DATARATE = 12 Mb/s (10 MHz channel spacing)	18.2.2, 18.2.3, 18.2.3.4	CF11& OF1.7:M	Yes □ No □ N/A □
*OF1.7.6	DATARATE = 18 Mb/s (10 MHz channel spacing)	18.2.2, 18.2.3, 18.2.3.4	CF11& OF1.7:O	Yes □ No □ N/A □
*OF1.7.7	DATARATE = 24 Mb/s (10 MHz channel spacing)	18.2.2, 18.2.3, 18.2.3.4	CF11& OF1.7:O	Yes □ No □ N/A □
*OF1.7.8	DATARATE = 27 Mb/s (10 MHz channel spacing)	18.2.2, 18.2.3, 18.2.3.4	CF11& OF1.7:O	Yes □ No □ N/A □
*OF1.8	5 MHz Channel spacing	18.2.2, 18.2.3, 18.2.3.4, Annex E	CF11:O, CF15&DSE2:M, CF15&DSE3:M	Yes □ No □ N/A □
*OF1.8.1	DATARATE = 1.5Mb/s (5 MHz channel spacing)	18.2.2, 18.2.3, 18.2.3.4	CF11& OF1.8:M	Yes □ No □ N/A □
*OF1.8.2	DATARATE = 2.25 Mb/s (5 MHz channel spacing)	18.2.2, 18.2.3, 18.2.3.4	CF11& OF1.8:O	Yes □ No □ N/A □
*OF1.8.3	DATARATE = 3Mb/s (5 MHz channel spacing)	18.2.2, 18.2.3, 18.2.3.4	CF11& OF1.8:M	Yes □ No □ N/A □
*OF1.8.4	DATARATE = 4.5 Mb/s (5 MHz channel spacing)	18.2.2, 18.2.3, 18.2.3.4	CF11& OF1.8:O	Yes □ No □ N/A □
*OF1.8.5	DATARATE = 6 Mb/s (5 MHz channel spacing)	18.2.2, 18.2.3, 18.2.3.4	CF11& OF1.8:M	Yes □ No □ N/A □
*OF1.8.6	DATARATE = 9 Mb/s (5 MHz channel spacing)	18.2.2, 18.2.3, 18.2.3.4	CF11& OF1.8:O	Yes □ No □ N/A □

B.4.8 OFDM PHY functions *(continued)*

Item	Feature	References	Status	Support
*OF1.8.7	DATARATE = 12 Mb/s (5 MHz channel spacing)	18.2.2, 18.2.3, 18.2.3.4	CF11& OF1.8:O	Yes □ No □ N/A □
*OF1.8.8	DATARATE = 13.5 Mb/s (5 MHz channel spacing)	18.2.2, 18.2.3, 18.2.3.4	CF11& OF1.8:O	Yes □ No □ N/A □
OF2: OFDM PLCP Sublayer				
OF2.1	RATE-dependent parameters	18.3.2.3	M	Yes □ No □
OF2.2	Timing related parameters	18.3.2.4	M	Yes □ No □
OF2.3	PLCP preamble: SYNC	18.3.3	M	Yes □ No □
OF2.4	PLCP header: SIGNAL	18.3.4	M	Yes □ No □
OF2.5	PLCP header: LENGTH	18.3.4.2	M	Yes □ No □
OF2.6	PLCP header: RATE	18.3.4.3	M	Yes □ No □
OF2.7	PLCP header: parity, reserve	18.3.4.4	M	Yes □ No □
OF2.8	PLCP header: SIGNAL TAIL	18.3.4.4	M	Yes □ No □
OF2.9	PLCP header: SERVICE	18.3.5.2	M	Yes □ No □
OF2.10	PLCP protocol data unit (PPDU): TAIL	18.3.5.3	M	Yes □ No □
OF2.11	PPDU: PAD	18.3.5.4	M	Yes □ No □
OF2.12	PLCP/OFDM PHY data scrambler and descrambler	18.3.5.5	M	Yes □ No □
OF2.13	Convolutional encoder	18.3.5.6	M	Yes □ No □
OF2.13.1	Rate R = 1/2	18.3.5.6	M	Yes □ No □
OF2.13.2	Punctured coding R = 2/3	18.3.5.6	OF1.2.7:M	Yes □ No □ N/A □
OF2.13.3	Punctured coding R = 3/4	18.3.5.6	OF1.2.2 OR OF1.2.4 OR OF1.2.6 OR OF1.2.8:M	Yes □ No □ N/A □
OF2.14	Data interleaving	18.3.5.7	M	Yes □ No □
OF2.15	Subcarrier modulation mapping	18.3.5.8	M	Yes □ No □
OF2.15.1	Binary phase shift keying (BPSK)	18.3.5.8	M	Yes □ No □
OF2.15.2	Quadrature phase shift keying (QPSK)	18.3.5.8	M	Yes □ No □
OF2.15.3	16-quadrature amplitude modulation (QAM)	18.3.5.8	M	Yes □ No □
OF2.15.4	64-QAM	18.3.5.8	OF1.2.7 OR OF1.2.8:M	Yes □ No □ N/A □
OF2.16	Pilot subcarriers	18.3.5.9	M	Yes □ No □
OF2.17	OFDM modulation	18.3.5.10	M	Yes □ No □
OF2.18	Packet duration calculation		M	Yes □ No □
OF2.19	CCA			
OF2.19.1	CCA: RSSI	18.3.6	M	Yes □ No □

B.4.8 OFDM PHY functions *(continued)*

Item	Feature	References	Status	Support
OF2.19.2	CCA: indication to MAC sublayer	18.3.6	M	Yes □ No □
*OF2.19.3	CCA-ED functionality	18.3.10.6	CF15:M	Yes □ No □ N/A □
OF2.19.3.1	CCA-ED energy only (RPI above threshold)	18.3.10.6	OF2.19.3: M	Yes □ No □ N/A □
OF2.19.3.2	Hold CCA busy for packet duration of a correctly received PLCP, but carrier lost during reception of MPDU	18.3.10.6	OF2.19.3: M	Yes □ No □ N/A □
OF2.20	PLCP data modulation and modulation rate change	18.3.7	M	Yes □ No □
OF2.21	Modulation-dependent parameters (10 MHz channel spacing)	18.3.2.3	CF11& OF1.7:M	Yes □ No □ N/A □
OF2.22	Timing-related parameters (10 MHz channel spacing)	18.3.2.4	CF11& OF1.7:M	Yes □ No □ N/A □
OF2.23	PLCP header: RATE (10 MHz channel spacing)	18.3.4.2	CF11& OF1.7:M	Yes □ No □ N/A □
OF2.24	Modulation-dependent parameters (5 MHz channel spacing)	18.3.2.3	CF11& OF1.8:M	Yes □ No □ N/A □
OF2.25	Timing-related parameters (5 MHz channel spacing)	18.3.2.4	CF11& OF1.8:M	Yes □ No □ N/A □
OF2.26	PLCP header: RATE (5 MHz channel spacing)	18.3.4.2	CF11& OF1.8:M	Yes □ No □ N/A □
OF3: PDM Operating Specification General				
OF3.1	Occupied channel bandwidth			
OF3.1.1	20 MHz channel spacing	18.3.8.2	M	Yes □ No □
OF3.1.2	10 MHz channel spacing	18.3.8.2	CF11& OF1.7:M	Yes □ No □ N/A □
OF3.1.3	5 MHz channel spacing	18.3.8.2	CF11& OF1.8:M	Yes □ No □ N/A □
OF3.2	Operating frequency range	18.3.8.4.1		
*OF3.2.1	4.9 GHz band	Annex E	CF11:O	Yes □ No □ N/A □
*OF3.2.2	5.0 GHz band	Annex E	CF11:M	Yes □ No □ N/A □
OF3.2.3	5.15–5.25 GHz band	18.3.8.4	O.1	Yes □ No □
OF3.2.4	5.25–5.35 GHz band	18.3.8.4	O.1	Yes □ No □
*OF3.2.5	5.47–5.725 GHz band	Annex E	CF10:M	Yes □ No □ N/A □
OF3.2.6	5.725–5.85 GHz band	18.3.8.4	O.1	Yes □ No □
*OF3.2.7	3.65–3.70 GHz band	Annex D, Annex E	CF15:M	Yes □ No □ N/A □
OF3.2.8	5.9 GHz band	Annex E	CF17:M	Yes □ No □ N/A □
OF3.3	Channelization			
OF3.3.1	5.15–5.25 GHz (20 MHz channel spacing)	Annex E	O.1	Yes □ No □
OF3.3.2	5.25–5.35 GHz (20 MHz channel spacing)	Annex E	O.1	Yes □ No □
OF3.3.3	5.725–5.825 GHz (20 MHz channel spacing)	Annex E	O.1	Yes □ No □

B.4.8 OFDM PHY functions *(continued)*

Item	Feature	References	Status	Support
OF3.3.4	5.15–5.25 GHz band in Japan (20 MHz channel spacing)	Annex E	CF11:M	Yes □ No □ N/A □
OF3.3.5	5.47–5.725 GHz (20 MHz channel spacing)	Annex E	CF10& OF3.2.5:M	Yes □ No □ N/A □
OF3.3.6	5.725–5.85 GHz (20 MHz channel spacing)	Annex E	O.1	Yes □ No □
OF3.3.7	4.9 GHz band (20 MHz channel spacing)	Annex E	CF11& OF3.2.1:M	Yes □ No □ N/A □
OF3.3.8	5.0 GHz band (20 MHz channel spacing)	Annex E	CF11& OF3.2.2:M	Yes □ No □ N/A □
OF3.3.9	4.9 GHz band (10 MHz channel spacing)	Annex E	CF11& OF3.2.1& OF1.7:M	Yes □ No □ N/A □
OF3.3.10	5.0 GHz band (10 MHz channel spacing)	Annex E	CF11& OF3.2.2& OF1.7:M	Yes □ No □ N/A □
OF3.3.11	4.9 GHz band (5 MHz channel spacing)	Annex E	CF11& OF3.2.1& OF1.8:M	Yes □ No □ N/A □
OF3.3.12	5.0 GHz band (5 MHz channel spacing)	Annex E	CF11& OF3.2.2& OF1.8:M	Yes □ No □ N/A □
OF3.3.13	3.65–3.70 GHz (20 MHz channel spacing)	Annex E	CF15&OF3.2.7:M	Yes □ No □ N/A □
OF3.3.14	3.65–3.70 GHz (10 MHz channel spacing)	Annex E	CF15&OF3.2.7&OF1.7:M	Yes □ No □ N/A □
OF3.3.15	3.65–3.70 GHz (5 MHz channel spacing)	Annex E	CF15&OF3.2.7&OF1.8:M	Yes □ No □ N/A □
OF3.3.16	5.9 GHz band (10 MHz channel spacing)	Annex E	CF17:O	Yes □ No □ N/A □
OF3.3.17	5.9 GHz band (20 MHz channel spacing)	Annex E	CF17:O	Yes □ No □ N/A □
OF3.3.18	5.9 GHz band (5 MHz channel spacing)	Annex E	CF17:O	Yes □ No □ N/A □
OF3.4	Number of operating channels	Annex E	M	Yes □ No □
OF3.5	Operating channel frequencies	Annex E	M	Yes □ No □
OF3.6	Transmit and receive in band and out of band spurious emission	Annex E	M	Yes □ No □
OF3.6.1	Interference-limited areas, 4.9 GHz band (20 MHz channel spacing)	Annex E	CF11& OF3.2.1:M	Yes □ No □ N/A □
OF3.6.2	Interference-limited areas, 5.0 GHz band (20 MHz channel spacing)	Annex E	CF11& OF3.2.2:M	Yes □ No □ N/A □
OF3.6.3	Interference-limited areas, 4.9 GHz band (10 MHz channel spacing)	Annex E	CF11& OF3.2.1& OF1.7:O	Yes □ No □ N/A □
OF3.6.4	Interference-limited areas, 5.0 GHz band (10 MHz channel spacing)	Annex E	CF11& OF3.2.2& OF1.7:O	Yes □ No □ N/A □

B.4.8 OFDM PHY functions *(continued)*

Item	Feature	References	Status	Support
OF3.6.5	Interference-limited areas, 4.9 GHz band (5 MHz channel spacing)	Annex E	CF11& OF3.2.1& OF1.8:O	Yes □ No □ N/A □
OF3.6.6	Interference-limited areas, 5.0 GHz band (5 MHz channel spacing)	Annex E	CF11& OF3.2.2& OF1.8:O	Yes □ No □ N/A □
OF3.7	TX RF delay	18.3.8.6	M	Yes □ No □
OF3.8	Slot time	18.3.8.7	M	Yes □ No □
OF3.8.1	Slot time (20 MHz channel spacing)	18.3.8.7	CF11& RC2:M	Yes □ No □ N/A □
OF3.8.2	Slot time (10 MHz channel spacing)	18.3.8.7	CF11& RC3& OF1.7:M	Yes □ No □ N/A □
OF3.8.3	Slot time (5 MHz channel spacing)	18.3.8.7	CF11& RC4& OF1.8:M	Yes □ No □ N/A □
OF3.9	Transmit and receive antenna port impedance	18.3.8.8	M	Yes □ No □
OF4: PMD Transmit Specification				
OF4.1	Transmit power levels		M	Yes □ No □
OF4.1.1	Power level (5.15–5.25 GHz)	18.3.9.2	OF3.3.1:M	Yes □ No □ N/A □
OF4.1.2	Power level (5.25–5.35 GHz)	18.3.9.2	OF3.3.2:M	Yes □ No □ N/A □
OF4.1.3	Power level (5.725–5.825 GHz)	18.3.9.2	OF3.3.3:M	Yes □ No □ N/A □
*OF4.1.4	Power Level (5.850–5.925 GHz), Class A	D.2.2	CF17:M	Yes □ No □ N/A □
*OF4.1.5	Power Level (5.850–5.925 GHz), Class B	D.2.2	CF17:O	Yes □ No □ N/A □
*OF4.1.6	Power Level (5.850–5.925 GHz), Class C	D.2.2	CF17:O	Yes □ No □ N/A □
*OF4.1.7	Power Level (5.850–5.925 GHz), Class D	D.2.2	CF17:O	Yes □ No □ N/A □
OF4.2	Spectrum mask	18.3.9.3	M	Yes □ No □
OF4.3	Spurious	18.3.9.4	M	Yes □ No □
OF4.4	Center frequency tolerance	18.3.9.5	M	Yes □ No □
OF4.5	Clock frequency tolerance	18.3.9.6	M	Yes □ No □
OF4.6	Modulation accuracy			Yes □ No □
OF4.6.1	Center frequency leakage	18.3.9.7.2	M	Yes □ No □
OF4.6.2	Spectral flatness	18.3.9.7.3	M	Yes □ No □
OF4.6.3	Transmitter constellation error < –5 dB	18.3.9.7.4	M	Yes □ No □
OF4.6.4	Transmitter constellation error < –8 dB	18.3.9.7.4	OF1.2.2:M	Yes □ No □ N/A □
OF4.6.5	Transmitter constellation error < –10 dB	18.3.9.7.4	M	Yes □ No □
OF4.6.6	Transmitter constellation error < –13 dB	18.3.9.7.4	OF1.2.4:M	Yes □ No □ N/A □
OF4.6.7	Transmitter constellation error < –16 dB	18.3.9.7.4	M	Yes □ No □
OF4.6.8	Transmitter constellation error < –19 dB	18.3.9.7.4	OF1.2.6:M	Yes □ No □ N/A □
OF4.6.9	Transmitter constellation error < –22 db	18.3.9.7.4	OF1.2.7:M	Yes □ No □ N/A □
OF4.6.10	Transmitter constellation error < –25 dB	18.3.9.7.4	OF1.2.8:M	Yes □ No □ N/A □

B.4.8 OFDM PHY functions *(continued)*

Item	Feature	References	Status	Support
OF4.7	Power level, 4.9 GHz band (20 MHz channel spacing)	18.3.9.2	CF11& OF3.12.1: M	Yes □ No □ N/A □
OF4.8	Power level, 5.0 GHz band (20 MHz channel spacing)	18.3.9.2	CF11& OF3.12.2: M	Yes □ No □ N/A □
OF4.9	Power level, 5.47–5.725 GHz band	18.3.9.2	CF11& OF3.12.3: M	Yes □ No □ N/A □
OF4.10	Power level, 4.9 GHz band (10 MHz channel spacing)	18.3.9.2	CF11& OF3.12.1& OF1.7:M	Yes □ No □ N/A □
OF4.11	Power level, 5.0 GHz band (10 MHz channel spacing)	18.3.9.2	CF11& OF3.12.2& OF1.7:M	Yes □ No □ N/A □
OF4.12	Power level, 4.9 GHz band (5 MHz channel spacing)	18.3.9.2	CF11& OF3.12.1& OF1.8:M	Yes □ No □ N/A □
OF4.13	Power level, 5.0 GHz band (5 MHz channel spacing)	18.3.9.2	CF11& OF3.12.2& OF1.8:M	Yes □ No □ N/A □
OF4.13a	Power level, 3.65–3.70 GHz (20 MHz channel spacing)	Annex E	CF15&OF3 .2.7:M	Yes □ No □ N/A □
OF4.13b	Power level, 3.65–3.70 GHz (10 MHz channel spacing)	Annex E	CF15&OF3 .2.7&OF1. 7:M	Yes □ No □ N/A □
OF4.13c	Power level, 3.65–3.70 GHz (5 MHz channel spacing)	Annex E	CF15&OF3 .2.7&OF1. 8:M	Yes □ No □ N/A □
OF4.14	Spectrum mask (20 MHz channel spacing)	18.3.9.3	CF11:M	Yes □ No □ N/A □
OF4.15	Spectrum mask (10 MHz channel spacing)	18.3.9.3	CF11& OF1.7:M	Yes □ No □ N/A □
OF4.15.1	Spectrum mask, Class A (10 MHz channel spacing)	D.2.3	OF4.1.4:M	Yes □ No □ N/A □
OF4.15.2	Spectrum mask, Class B (10 MHz channel spacing)	D.2.3	OF4.1.5:M	Yes □ No □ N/A □
OF4.15.3	Spectrum mask, Class C (10 MHz channel spacing)	D.2.3	OF4.1.6:M	Yes □ No □ N/A □
OF4.15.4	Spectrum mask, Class D (10 MHz channel spacing)	D.2.3	OF4.1.7:M	Yes □ No □ N/A □
OF4.16	Spectrum mask (5 MHz channel spacing)	18.3.9.3	CF11& OF1.8:M	Yes □ No □ N/A □
OF4.17	Transmitter constellation error (10 MHz channel spacing)			
OF4.17.1	Transmitter constellation error < –5 dB (10 MHz channel spacing)	18.3.9.7.4	CF11& OF1.7.1:M	Yes □ No □ N/A □
OF4.17.2	Transmitter constellation error < –8 dB (10 MHz channel spacing)	18.3.9.7.4	CF11& OF1.7.2:M	Yes □ No □ N/A □

B.4.8 OFDM PHY functions *(continued)*

Item	Feature	References	Status	Support
OF4.17.3	Transmitter constellation error < –10 dB (10 MHz channel spacing)	18.3.9.7.4	CF11& OF1.7.3:M	Yes □ No □ N/A □
OF4.17.4	Transmitter constellation error < –13 dB (10 MHz channel spacing)	18.3.9.7.4	CF11& OF1.7.4:M	Yes □ No □ N/A □
OF4.17.5	Transmitter constellation error < –16 dB (10 MHz channel spacing)	18.3.9.7.4	CF11& OF1.7.5:M	Yes □ No □ N/A □
OF4.17.6	Transmitter constellation error < –19 dB (10 MHz channel spacing)	18.3.9.7.4	CF11& OF1.7.6:M	Yes □ No □ N/A □
OF4.17.7	Transmitter constellation error < –22 dB (10 MHz channel spacing)	18.3.9.7.4	CF11& OF1.7.7:M	Yes □ No □ N/A □
OF4.17.8	Transmitter constellation error < –25 dB (10 MHz channel spacing)	18.3.9.7.4	CF11& OF1.7.8:M	Yes □ No □ N/A □
OF4.18	Transmitter constellation error (5 MHz channel spacing)			
OF4.18.1	Transmitter constellation error < –5 dB (5 MHz channel spacing)	18.3.9.7.4	CF11& OF1.8.1:M	Yes □ No □ N/A □
OF4.18.2	Transmitter constellation error < –8 dB (5 MHz channel spacing)	18.3.9.7.4	CF11& OF1.8.2:M	Yes □ No □ N/A □
OF4.18.3	Transmitter constellation error < –10 dB (5 MHz channel spacing)	18.3.9.7.4	CF11& OF1.8.3:M	Yes □ No □ N/A □
OF4.18.4	Transmitter constellation error < –13 dB (5 MHz channel spacing)	18.3.9.7.4	CF11& OF1.8.4:M	Yes □ No □ N/A □
OF4.18.5	Transmitter constellation error < –16 dB (5 MHz channel spacing)	18.3.9.7.4	CF11& OF1.8.5:M	Yes □ No □ N/A □
OF4.18.6	Transmitter constellation error < –19 dB (5 MHz channel spacing)	18.3.9.7.4	CF11& OF1.8.6:M	Yes □ No □ N/A □
OF4.18.7	Transmitter constellation error < –22 dB (5 MHz channel spacing)	18.3.9.7.4	CF11& OF1.8.7:M	Yes □ No □ N/A □
OF4.18.8	Transmitter constellation error < –25 dB (5 MHz channel spacing)	18.3.9.7.4	CF11& OF1.8.8:M	Yes □ No □ N/A □
OF5: PMD Receiver Specifications				
OF5.1	Minimum input level sensitivity at packet error ratio (PER) = 10% with 1000 octet frames			
OF5.1.1	–82 dBm for 6 Mb/s	18.3.10.2	M	Yes □ No □
OF5.1.2	–81 dBm for 9 Mb/s	18.3.10.2	OF1.2.2:M	Yes □ No □ N/A □
OF5.1.3	–79 dBm for 12 Mb/s	18.3.10.2	M	Yes □ No □
OF5.1.4	–77 dBm for 18 Mb/s	18.3.10.2	OF1.2.4:M	Yes □ No □ N/A □
OF5.1.5	–74 dBm for 24 Mb/s	18.3.10.2	M	Yes □ No □
OF5.1.6	–70 dBm for 36 Mb/s	18.3.10.2	OF1.2.6:M	Yes □ No □ N/A □
OF5.1.7	–66 dBm for 48 Mb/s	18.3.10.2	OF1.2.7:M	Yes □ No □ N/A □
OF5.1.8	–65 dBm for 54 Mb/s	18.3.10.2	OF1.2.8:M	Yes □ No □ N/A □
OF5.2	Adjacent channel rejection	18.3.10.3	M	Yes □ No □
OF5.2.1	Optional adjacent channel rejection	18.3.10.3	O	Yes □ No □
OF5.3	Nonadjacent channel rejection	18.3.10.4	M	Yes □ No □

B.4.8 OFDM PHY functions *(continued)*

Item	Feature	References	Status	Support
OF5.3.1	Optional nonadjacent channel rejection	18.3.10.4	O	Yes □ No □
OF5.4	Maximum input level	18.3.10.5	M	Yes □ No □
OF5.5	CCA sensitivity	18.3.10.6	M	Yes □ No □
OF5.6	Maximum input level sensitivity at packet error ratio (PER) = 10% with 1000 octet frames (10 MHz channel spacing)			
OF5.6.1	–85 dBm for 3 Mb/s (10 MHz channel spacing)	18.3.10.2	CF11& OF1.7.1:M	Yes □ No □ N/A □
OF5.6.2	–84 dBm for 4.5 Mb/s (10 MHz channel spacing)	18.3.10.2	CF11& OF1.7.2:M	Yes □ No □ N/A □
OF5.6.3	–82 dBm for 6 Mb/s (10 MHz channel spacing)	18.3.10.2	CF11& OF1.7.3:M	Yes □ No □ N/A □
OF5.6.4	–80 dBm for 9 Mb/s (10 MHz channel spacing)	18.3.10.2	CF11& OF1.7.4:M	Yes □ No □ N/A □
OF5.6.5	–77 dBm for 12 Mb/s (10 MHz channel spacing)	18.3.10.2	CF11& OF1.7.5:M	Yes □ No □ N/A □
OF5.6.6	–73 dBm for 18 Mb/s (10 MHz channel spacing)	18.3.10.2	CF11& OF1.7.6:M	Yes □ No □ N/A □
OF5.6.7	–69 dBm for 24 Mb/s (10 MHz channel spacing)	18.3.10.2	CF11& OF1.7.7:M	Yes □ No □ N/A □
OF5.6.8	–68 dBm for 27 Mb/s (10 MHz channel spacing)	18.3.10.2	CF11& OF1.7.8:M	Yes □ No □ N/A □
OF5.7	Adjacent channel rejection (10 MHz channel spacing)	18.3.10.3	CF11& OF1.7:M	Yes □ No □ N/A □
OF5.8	Nonadjacent channel rejection (10 MHz channel spacing)	18.3.10.4	CF11& OF1.7:M	Yes □ No □ N/A □
OF5.9	Maximum input level (10 MHz channel spacing)	18.3.10.5	CF11& OF1.7:M	Yes □ No □ N/A □
OF5.10	CCA sensitivity (10 MHz channel spacing)	18.3.10.6	CF11& OF1.7:M	Yes □ No □ N/A □
OF5.11	Maximum input level sensitivity at packet error ratio (PER) = 10% with 1000 octet frames (5 MHz channel spacing)			
OF5.11.1	–85 dBm for 3 Mb/s (5 MHz channel spacing)	18.3.10.2	CF11& OF1.8.1:M	Yes □ No □ N/A □
OF5.11.2	–84 dBm for 4.5 Mb/s (5 MHz channel spacing)	18.3.10.2	CF11& OF1.8.2:M	Yes □ No □ N/A □
OF5.11.3	–82 dBm for 6 Mb/s (5 MHz channel spacing)	18.3.10.2	CF11& OF1.8.3:M	Yes □ No □ N/A □
OF5.11.4	–80 dBm for 9 Mb/s (5 MHz channel spacing)	18.3.10.2	CF11& OF1.8.4:M	Yes □ No □ N/A □
OF5.11.5	–77 dBm for 12 Mb/s (5 MHz channel spacing)	18.3.10.2	CF11& OF1.8.5:M	Yes □ No □ N/A □
OF5.11.6	–73 dBm for 18 Mb/s (5 MHz channel spacing)	18.3.10.2	CF11& OF1.8.6:M	Yes □ No □ N/A □
OF5.11.7	–69 dBm for 24 Mb/s (5 MHz channel spacing)	18.3.10.2	CF11& OF.1.8.7:M	Yes □ No □ N/A □

B.4.8 OFDM PHY functions *(continued)*

Item	Feature	References	Status	Support
OF5.11.8	–68 dBm for 27 Mb/s (5 MHz channel spacing)	18.3.10.2	CF11& OF1.8.8:M	Yes □ No □ N/A □
OF5.12	Adjacent channel rejection (5 MHz channel spacing)	18.3.10.3	CF11& OF1.8:M	Yes □ No □ N/A □
OF5.13	Nonadjacent channel rejection (5 MHz channel spacing)	18.3.10.4	CF11& OF1.8:M	Yes □ No □ N/A □
OF5.14	Maximum input level (5 MHz channel spacing)	18.3.10.5	CF11& OF1.8:M	Yes □ No □ N/A □
OF5.15	CCA sensitivity (5 MHz channel spacing)	18.3.10.6	CF11& OF1.8:M	Yes □ No □ N/A □
OF6: Transmit PLCP				
OF6.1	Transmit: transmit on MAC request	18.3.11	M	Yes □ No □
OF6.2	Transmit: format and data encoding	18.3.11	M	Yes □ No □
OF6.3	Transmit: timing	18.3.11	M	Yes □ No □
OF7: Receive PLCP				
OF7.1	Receive: receive and data decoding	18.3.12	M	Yes □ No □
OF8: PLME				
OF8.1	PLME: support PLME_SAP management primitives	18.4.1	M	Yes □ No □
OF8.2	PLME: support PHY MIB	18.4.2	M	Yes □ No □
OF8.3	PLME: support PHY characteristics	18.4.3	M	Yes □ No □
OF8.4	PLME:support PHY characteristics (dot11ChannelStartingFactor)	18.4.2	CF11:M	Yes □ No □ N/A □
OF9: OFDM PMD Sublayer				
OF9.1	PMD: support PMD_SAP peer-to-peer service primitives	18.5.4.2, 18.5.5.2, 18.5.5.3	M	Yes □ No □
OF9.2	PMD: support PMD_SAP sublayer-to-sublayer service primitives	18.5.4.3, 18.5.5.4, 18.5.5.5, 18.5.5.6, 18.5.5.7, 18.5.5.8	M	Yes □ No □
OF9.3	PMD_SAP service primitive parameters			
OF9.3.1	Parameter: TXD_UNIT	18.5.4.4	M	Yes □ No □
OF9.3.2	Parameter: RXD_UNIT	18.5.4.4	M	Yes □ No □
OF9.3.3	Parameter: TXPWR_LEVEL	18.5.4.4	M	Yes □ No □
OF9.3.4	Parameter: RATE (12 Mb/s)	18.5.4.4	M	Yes □ No □
OF9.3.5	Parameter: RATE (24 Mb/s)	18.5.4.4	M	Yes □ No □
OF9.3.6	Parameter: RATE (48 Mb/s)	18.5.4.4	M	Yes □ No □
OF9.3.7	Parameter: RATE (72 Mb/s)	18.5.4.4	O	Yes □ No □
OF9.3.8	Parameter: RSSI	18.5.4.4	M	Yes □ No □
OF9.4				

B.4.8 OFDM PHY functions *(continued)*

Item	Feature	References	Status	Support
OF9.4.1	Parameter: RATE (6 Mb/s for 10 MHz channel spacing)	18.5.4.4	CF11& OF1.7:M	Yes □ No □ N/A □
OF9.4.2	Parameter: RATE (12 Mb/s for 10 MHz channel spacing)	18.5.4.4	CF11& OF1.7:M	Yes □ No □ N/A □
OF9.4.3	Parameter: RATE (24 Mb/s for 10 MHz channel spacing)	18.5.4.4	CF11& OF1.7:M	Yes □ No □ N/A □
OF9.4.4	Parameter: RATE (36 Mb/s for 10 MHz channel spacing)	18.5.4.4	CF11& OF1.7:O	Yes □ No □ N/A □
OF9.4.5	Parameter: RATE (3 Mb/s for 5MHz channel spacing)	18.5.4.4	CF11& OF1.8:M	Yes □ No □ N/A □
OF9.4.6	Parameter: RATE (6 Mb/s for 5 MHz channel spacing)	18.5.4.4	CF11& OF1.8:M	Yes □ No □ N/A □
OF9.4.7	Parameter: RATE (12 Mb/s for 5 MHz channel spacing)	18.5.4.4	CF11& OF1.8:M	Yes □ No □ N/A □
OF9.4.8	Parameter: RATE (18 Mb/s for 5 MHz channel spacing)	18.5.4.4	CF11& OF1.8:O	Yes □ No □ N/A □
OF10: Geographic Area Specific Requirements				
*OF10.1	Geographic areas	18.3.8.3, 18.3.8.4, 18.3.8.5, 18.3.9.4	M	Yes □ No □
OF10.2	Regulatory domain extensions	18.3.8.4.3, 18.3.8.5, 18.3.9.2, 18.3.9.3, Annex E	CF11:M	Yes □ No □ N/A □

B.4.9 High Rate, direct sequence PHY functions

Item	PHY feature	References	Status	Support
	Are the following PHY features supported?			
HRDS1	Long preamble and header procedures	17.2	M	Yes □ No □
HRDS1.1	Long direct sequence preamble prepended on TX	17.2.1	M	Yes □ No □
HRDS1.2	Long PLCP integrity check generation	17.2.3, 17.2.3.7	M	Yes □ No □
HRDS1.3	TX rate change capability	17.2.3.4	M	Yes □ No □
HRDS1.4	Supported data rates	17.1, 17.2.3.4	M	Yes □ No □
HRDS1.5	Data scrambler	17.2.4	M	Yes □ No □
HRDS1.6	Scrambler initialization	17.2.4	M	Yes □ No □
*HRDS2	Channel Agility option	17.3.2	O	Yes □ No □
*HRDS3	Short preamble and header procedures	17.2	O	Yes □ No □
HRDS3.1	Short preamble prepended on TX	17.2.2	HRDS3:M	Yes □ No □ N/A □

B.4.9 High Rate, direct sequence PHY functions *(continued)*

Item	PHY feature	References	Status	Support
HRDS3.2	Short header transmission	17.2.3.9, 17.2.3.10, 17.2.3.11, 17.2.3.12, 17.2.3.13, 17.2.3.14, 17.2.3.15	HRDS3:M	Yes □ No □ N/A □
HRDS4	Long preamble process on RX	17.2.6	M	Yes □ No □
HRDS4.1	PLCP format	17.2.6	M	Yes □ No □
HRDS4.2	PLCP integrity check verify	17.2.6	M	Yes □ No □
HRDS4.3	RX Rate change capability	17.2.6	M	Yes □ No □
HRDS4.4	Data whitener descrambler	17.2.6	M	Yes □ No □
*HRDS5	Short preamble process on RX	17.2.6	HRDS3:M	Yes □ No □ N/A □
HRDS5.1	PLCP format	17.2.6	HRDS5:M	Yes □ No □ N/A □
HRDS5.2	PLCP integrity check verify	17.2.6	HRDS5:M	Yes □ No □ N/A □
HRDS5.3	RX rate change capability	17.2.6	HRDS5:M	Yes □ No □ N/A □
HRDS5.4	Data whitener descrambler	17.2.6	HRDS5:M	Yes □ No □ N/A □
*HRDS6	Operating channel capability	—	—	—
*HRDS6.1	North America (FCC)	17.4.6.3	HRDS6:O.3	Yes □ No □ N/A □
HRDS6.1.1	Channel 1	17.4.6.3	HRDS6.1:M	Yes □ No □ N/A □
HRDS6.1.2	Channel 2	17.4.6.3	HRDS6.1:M	Yes □ No □ N/A □
HRDS6.1.3	Channel 3	17.4.6.3	HRDS6.1:M	Yes □ No □ N/A □
HRDS6.1.4	Channel 4	17.4.6.3	HRDS6.1:M	Yes □ No □ N/A □
HRDS6.1.5	Channel 5	17.4.6.3	HRDS6.1:M	Yes □ No □ N/A □
HRDS6.1.6	Channel 6	17.4.6.3	HRDS6.1:M	Yes □ No □ N/A □
HRDS6.1.7	Channel 7	17.4.6.3	HRDS6.1:M	Yes □ No □ N/A □
HRDS6.1.8	Channel 8	17.4.6.3	HRDS6.1:M	Yes □ No □ N/A □
HRDS6.1.9	Channel 9	17.4.6.3	HRDS6.1:M	Yes □ No □ N/A □
HRDS6.1.10	Channel 10	17.4.6.3	HRDS6.1:M	Yes □ No □ N/A □
HRDS6.1.11	Channel 11	17.4.6.3	HRDS6.1:M	Yes □ No □ N/A □
*HRDS6.2	Canada (IC)	17.4.6.3	HRDS6:O.3	Yes □ No □ N/A □
HRDS6.2.1	Channel 1	17.4.6.3	HRDS6.2:M	Yes □ No □ N/A □
HRDS6.2.2	Channel 2	17.4.6.3	HRDS6.2:M	Yes □ No □ N/A □
HRDS6.2.3	Channel 3	17.4.6.3	HRDS6.2:M	Yes □ No □ N/A □
HRDS6.2.4	Channel 4	17.4.6.3	HRDS6.2:M	Yes □ No □ N/A □
HRDS6.2.5	Channel 5	17.4.6.3	HRDS6.2:M	Yes □ No □ N/A □
HRDS6.2.6	Channel 6	17.4.6.3	HRDS6.2:M	Yes □ No □ N/A □
HRDS6.2.7	Channel 7	17.4.6.3	HRDS6.2:M	Yes □ No □ N/A □
HRDS6.2.8	Channel 8	17.4.6.3	HRDS6.2:M	Yes □ No □ N/A □
HRDS6.2.9	Channel 9	17.4.6.3	HRDS6.2:M	Yes □ No □ N/A □
HRDS6.2.10	Channel 10	17.4.6.3	HRDS6.2:M	Yes □ No □ N/A □
HRDS6.2.11	Channel 11	17.4.6.3	HRDS6.2:M	Yes □ No □ N/A □

B.4.9 High Rate, direct sequence PHY functions *(continued)*

Item	PHY feature	References	Status	Support
*HRDS6.3	Europe (ETSI)	17.4.6.3	HRDS6:O.3	Yes □ No □ N/A □
HRDS6.3.1	Channel 1	17.4.6.3	HRDS6.3:M	Yes □ No □ N/A □
HRDS6.3.2	Channel 2	17.4.6.3	HRDS6.3:M	Yes □ No □ N/A □
HRDS6.3.3	Channel 3	17.4.6.3	HRDS6.3:M	Yes □ No □ N/A □
HRDS6.3.4	Channel 4	17.4.6.3	HRDS6.3:M	Yes □ No □ N/A □
HRDS6.3.5	Channel 5	17.4.6.3	HRDS6.3:M	Yes □ No □ N/A □
HRDS6.3.6	Channel 6	17.4.6.3	HRDS6.3:M	Yes □ No □ N/A □
HRDS6.3.7	Channel 7	17.4.6.3	HRDS6.3:M	Yes □ No □ N/A □
HRDS6.3.8	Channel 8	17.4.6.3	HRDS6.3:M	Yes □ No □ N/A □
HRDS6.3.9	Channel 9	17.4.6.3	HRDS6.3:M	Yes □ No □ N/A □
HRDS6.3.10	Channel 10	17.4.6.3	HRDS6.3:M	Yes □ No □ N/A □
HRDS6.3.11	Channel 11	17.4.6.3	HRDS6.3:M	Yes □ No □ N/A □
HRDS6.3.12	Channel 12	17.4.6.3	HRDS6.3:M	Yes □ No □ N/A □
HRDS6.3.13	Channel 13	17.4.6.3	HRDS6.3:M	Yes □ No □ N/A □
*HRDS6.4	France	17.4.6.3	HRDS6:O.3	Yes □ No □ N/A □
HRDS6.4.1	Channel 10	17.4.6.3	HRDS6.4:M	Yes □ No □ N/A □
HRDS6.4.2	Channel 11	17.4.6.3	HRDS6.4:M	Yes □ No □ N/A □
HRDS6.4.3	Channel 12	17.4.6.3	HRDS6.4:M	Yes □ No □ N/A □
HRDS6.4.4	Channel 13	17.4.6.3	HRDS6.4:M	Yes □ No □ N/A □
*HRDS6.5	Spain	17.4.6.3	HRDS6:O.3	Yes □ No □ N/A □
HRDS6.5.1	Channel 10	17.4.6.3	HRDS6.5:M	Yes □ No □ N/A □
HRDS6.5.2	Channel 11	17.4.6.3	HRDS6.5:M	Yes □ No □ N/A □
*HRDS6.6	Japan	17.4.6.3	HRDS6:O.3	Yes □ No □ N/A □
HRDS6.6.1	Channel 1	17.4.6.3	HRDS6.6:M	Yes □ No □ N/A □
HRDS6.6.2	Channel 2	17.4.6.3	HRDS6.6:M	Yes □ No □ N/A □
HRDS6.6.3	Channel 3	17.4.6.3	HRDS6.6:M	Yes □ No □ N/A □
HRDS6.6.4	Channel 4	17.4.6.3	HRDS6.6:M	Yes □ No □ N/A □
HRDS6.6.5	Channel 5	17.4.6.3	HRDS6.6:M	Yes □ No □ N/A □
HRDS6.6.6	Channel 6	17.4.6.3	HRDS6.6:M	Yes □ No □ N/A □
HRDS6.6.7	Channel 7	17.4.6.3	HRDS6.6:M	Yes □ No □ N/A □
HRDS6.6.8	Channel 8	17.4.6.3	HRDS6.6:M	Yes □ No □ N/A □
HRDS6.6.9	Channel 9	17.4.6.3	HRDS6.6:M	Yes □ No □ N/A □
HRDS6.6.10	Channel 10	17.4.6.3	HRDS6.6:M	Yes □ No □ N/A □
HRDS6.6.11	Channel 11	17.4.6.3	HRDS6.6:M	Yes □ No □ N/A □
HRDS6.6.12	Channel 12	17.4.6.3	HRDS6.6:M	Yes □ No □ N/A □
HRDS6.6.13	Channel 13	17.4.6.3	HRDS6.6:M	Yes □ No □ N/A □
HRDS6.6.14	Channel 14	17.4.6.3	HRDS6.6:M	Yes □ No □ N/A □
*HRDS6.7	China (Radio Administration The Radio Administration of P.R.China)	17.4.6.3	HRDS6:O.3	Yes □ No □ N/A □
HRDS6.7.1	Channel 1	17.4.6.3	HRDS6.7:M	Yes □ No □ N/A □

B.4.9 High Rate, direct sequence PHY functions *(continued)*

Item	PHY feature	References	Status	Support
HRDS6.7.2	Channel 2	17.4.6.3	HRDS6.7:M	Yes □ No □ N/A □
HRDS6.7.3	Channel 3	17.4.6.3	HRDS6.7:M	Yes □ No □ N/A □
HRDS6.7.4	Channel 4	17.4.6.3	HRDS6.7:M	Yes □ No □ N/A □
HRDS6.7.5	Channel 5	17.4.6.3	HRDS6.7:M	Yes □ No □ N/A □
HRDS6.7.6	Channel 6	17.4.6.3	HRDS6.7:M	Yes □ No □ N/A □
HRDS6.7.7	Channel 7	17.4.6.3	HRDS6.7:M	Yes □ No □ N/A □
HRDS6.7.8	Channel 8	17.4.6.3	HRDS6.7:M	Yes □ No □ N/A □
HRDS6.7.9	Channel 9	17.4.6.3	HRDS6.7:M	Yes □ No □ N/A □
HRDS6.7.10	Channel 10	17.4.6.3	HRDS6.7:M	Yes □ No □ N/A □
HRDS6.7.11	Channel 11	17.4.6.3	HRDS6.7:M	Yes □ No □ N/A □
HRDS6.7.12	Channel 12	17.4.6.3	HRDS6.7:M	Yes □ No □ N/A □
HRDS6.7.13	Channel 13	17.4.6.3	HRDS6.7:M	Yes □ No □ N/A □
HRDS7	Hop sequences		HRDS2:M	Yes □ No □ N/A □
HRDS8	Complementary code keying (CCK) bits to symbol mapping			
HRDS8.1	5.5 Mb/s	17.4.6.6	M	Yes □ No □
HRDS8.2	11 Mb/s	17.4.6.6	M	Yes □ No □
*HRDS9	PBCC bits to symbol mappings	17.4.6.7	O	Yes □ No □
HRDS9.1	5.5 Mb/s	17.4.6.7	HRDS9:M	Yes □ No □ N/A □
HRDS9.2	11 Mb/s	17.4.6.7	HRDS9:M	Yes □ No □ N/A □
*HRDS10	CCA functionality	17.4.8.5		
HRDS10.1	CCA Mode 1, energy only (RSSI above threshold)	17.4.8.5	HRDS10:O.4	Yes □ No □ N/A □
HRDS10.2	CCA Mode 4, CS with timer	17.4.8.5	HRDS10:O.4	Yes □ No □ N/A □
HRDS10.3	CCA Mode 5, energy detect with High Rate CS	17.4.8.5	HRDS10:O.4	Yes □ No □ N/A □
HRDS10.4	Hold CCA busy for packet duration of a correctly received PLCP, but carrier lost during reception of MPDU.	17.2.6	M	Yes □ No □
HRDS10.5	Hold CCA busy for packet duration of a correctly received, but out of spec, PLCP.	17.2.6	M	Yes □ No □
HRDS11	Transmit antenna selection	17.4.5.9	O	Yes □ No □
HRDS12	Receive antenna diversity	17.4.5.9, 17.4.5.10	O	Yes □ No □
*HRDS13	Antenna port(s) availability	17.4.6.9	O	Yes □ No □
HRDS13.1	If available (50 Ω impedance)	17.4.6.9	HRDS13:M	Yes □ No □ N/A □
*HRDS14	Transmit power level support	17.4.5.10, 17.4.7.3	O	Yes □ No □
HRDS14.1	If greater than 100 mW capability	17.4.7.3	HRDS14:M	Yes □ No □ N/A □
HRDS15	Spurious emissions conformance	17.4.6.9	M	Yes □ No □
HRDS16	TX-to-RX turnaround time	17.4.6.10	M	Yes □ No □
HRDS17	RX-to-TX turnaround time	17.4.6.11	M	Yes □ No □
HRDS18	Slot time	17.4.6.12	M	Yes □ No □
HRDS19	ED reporting time	17.4.6.11, 17.4.8.5	M	Yes □ No □

B.4.9 High Rate, direct sequence PHY functions *(continued)*

Item	PHY feature	References	Status	Support
HRDS20	Minimum transmit power level	17.4.7.3	M	Yes □ No □
HRDS21	Transmit spectral mask conformance	17.4.7.4	M	Yes □ No □
HRDS22	Transmitted center frequency tolerance	17.4.7.5	M	Yes □ No □
HRDS23	Chip clock frequency tolerance	17.4.7.6	M	Yes □ No □
HRDS24	Transmit power-on ramp	17.4.7.7	M	Yes □ No □
HRDS25	Transmit power-down ramp	17.4.7.7	M	Yes □ No □
HRDS26	RF carrier suppression	17.4.7.8	M	Yes □ No □
HRDS27	Transmit modulation accuracy	17.4.7.9	M	Yes □ No □
HRDS28	Receiver minimum input level sensitivity	17.4.8.2	M	Yes □ No □
HRDS29	Receiver maximum input level	17.4.8.3	M	Yes □ No □
HRDS30	Receiver adjacent channel rejection	17.4.8.4	M	Yes □ No □
HRDS31	MIB	17.3.2, Annex J	M	Yes □ No □
HRDS31.1	PHY object class	17.3.3	M	Yes □ No □

B.4.10 Regulatory Domain Extensions

Item	Protocol capability	References	Status	Support
MD1	Country element Length Country String First Channel Number Maximum Transmit Power Level Number of Channels	8.3.3.2, 8.3.3.10 8.4.2.10	CF8:M	Yes □ No □ N/A □
MD2	Inclusion of the Request information in the Probe Request frame	8.3.3.9	CF8:O	Yes □ No □ N/A □
MD3	Hopping Pattern Parameters Element ID Prime Radix Number of Channels	8.4.2.11	(CF3 or (CF7 and HRDS2)) and CF8:M	Yes □ No □ N/A □
MD4	Hopping Pattern element Format Element ID Random table method	8.4.2.12	(CF3 or (CF7 and HRDS2)) and CF8:M	Yes □ No □ N/A □
MD5	Request element Format Element ID Order of the Requested Elemented IDs	8.4.2.13	CF8:M	Yes □ No □ N/A □

B.4.10 Regulatory Domain Extensions *(continued)*

Item	Protocol capability	References	Status	Support
MD6	Entering a Regulatory Domain Lost Connectivity with its extended service set (ESS) Passive Scanning to learn Beacon information Transmit Probe Request	9.18.2	CF8:M	Yes □ No □ N/A □
MD7	Determination of the Hopping Patterns [Hop Index Method without table, Hop Index Method with table, and hyperbolic congruence code (HCC)/extended HCC (EHCC)] The hopping pattern option is obsolete. Consequently this option may be removed in a later revision of the standard.	8.4.2.12, 9.18.3	(CF3 or (CF7 and HRDS2)) and CF8:M	Yes □ No □ N/A □
MD8	Roaming requires Beacon frame with country element	10.1.4.4	CF8:M	Yes □ No □ N/A □
MD9	Actions to be taken upon the receipt of the Beacon frame	10.1.4.5	CF8:M	Yes □ No □ N/A □
MD10	Ignore improperly formed Request element	8.3.3.10	CF8:O	Yes □ No □ N/A □
MD11	Hopping Pattern set attribute	14.9.2.19	(CF3 or (CF7 and HRDS2)) and CF8:M	Yes □ No □ N/A □
MD12	Operating and Coverage classes	8.4.2.10	RC1:M	Yes □ No □ N/A □
MD13	Reserved First Channel Number	9.18.5	CF15:M	Yes □ No □ N/A □
MD14	Reserved Operating Class	9.18.5	CF15:M	Yes □ No □ N/A □
MD15	Operation with operating classes Multiple classes in Country element Multiple classes in Association and Reassociation frames	 9.18.5 9.18.5	 CF15:M CF15:M	 Yes □ No □ N/A □ Yes □ No □ N/A □

B.4.11 ERP functions

Item	PHY features	References	Status	Support
*ERP1	Transmit and Receive ERP-DSSS data rates 1 and 2 Mb/s and ERP-CCK data rates 5.5 and 11 Mb/s	19.3.2	CF9:M	Yes □ No □ N/A □
ERP1.1	Transmit and receive ERP-OFDM data rates of 6, 12, and 24 Mb/s	19.3.2	CF9:M	Yes □ No □ N/A □

B.4.11 ERP functions *(continued)*

Item	PHY features	References	Status	Support
ERP1.2	Transmit and receive ERP-OFDM data rate of 9 Mb/s	19.3.2	ERP1:O	Yes □ No □ N/A □
ERP1.3	Transmit and receive ERP-OFDM data rate of 18 Mb/s	19.3.2	ERP1:O	Yes □ No □ N/A □
ERP1.4	Transmit and receive ERP-OFDM data rate of 36 Mb/s	19.3.2	ERP1:O	Yes □ No □ N/A □
ERP1.5	Transmit and receive ERP-OFDM data rate of 48 Mb/s	19.3.2	ERP1:O	Yes □ No □ N/A □
ERP1.6	Transmit and receive ERP-OFDM data rate of 54 Mb/s	19.3.2	ERP1:O	Yes □ No □ N/A □
*ERP2	Transmit and receive ERP-PBCC data rate of 22 Mb/s The ERP-PBCC option is obsolete. Consequently this option may be removed in a later revision of the standard.	19.3.2	CF9&HRDS9.1&HRDS9.2:O	Yes □ No □ N/A □
ERP2.1	Transmit and receive ERP-PBCC data rate of 33 Mb/s	19.3.2	ERP2:O	Yes □ No □ N/A □
*ERP3	Transmit and receive DSSS-OFDM data at same rates as ERP-OFDM	19.3.2	CF9:O	Yes □ No □ N/A □
ERP4	Support of ERP3 required PPDU formats as described in reference	19.3.2	CF9:O	Yes □ No □ N/A □
ERP5	Able to transmit and receive long and short DSSS as well as OFDM preambles	19.3.2	CF9:M	Yes □ No □ N/A □
ERP6	Set SERVICE field bits for DSSS-OFDM, ERP-PBCC, locked clocks, and length extension (b0, b2, b3, b5, b6, and b7)	19.3.2.2	CF9:M	Yes □ No □ N/A □
ERP7	Set b1 & b4 of long and short preamble PPDU SERVICE field to 0	19.3.2.2	CF9:M	Yes □ No □ N/A □
ERP8	b2 shall be set to 1 in all long and short preamble PPDU SERVICE fields	19.3.2.2	CF9:M	Yes □ No □ N/A □

B.4.11 ERP functions *(continued)*

Item	PHY features	References	Status	Support
ERP9	Set bits b5, b6, and b7 of the long and short preamble PPDU SERVICE fields as described in the reference	19.3.2.2, 19.3.2.2.3	CF9:M	Yes □ No □ N/A □
ERP10	Use Clause 16 or Clause 17 rates when using protection mechanisms	9.23	CF9:M	Yes □ No □ N/A □
ERP11	SIGNAL field set to 3 Mb/s in all long and short DSSS-OFDM PPDU formats as described in the reference	19.3.2.5	ERP3:M	Yes □ No □ N/A □
ERP12	Calculate DSSS-OFDM length with signal extension	19.3.2.6	ERP3:M	Yes □ No □ N/A □
ERP13	Set ERP-PBCC encoder in state 0 at beginning of PPDU	19.3.3.2	ERP2:M	Yes □ No □ N/A □
ERP14	Set phase of ERP-PBCC relative to header	19.3.3.2	ERP2:M	Yes □ No □ N/A □
ERP15	Use same pulse shape for 22 and 33 Mb/s	19.3.3.2	ERP2:M	Yes □ No □ N/A □
ERP16	Add signal extension of 6 µs	19.3.2.4	CF9:M	Yes □ No □ N/A □
ERP17	Simultaneous CCA on long preamble Barker, short preamble Barker, and OFDM	19.3.5	CF9:M	Yes □ No □ N/A □
ERP18	CCA with energy detect above threshold and CS	19.3.5	CF9:M	Yes □ No □ N/A □
ERP19	Decode as DSSS-OFDM if signal field indicates 3 Mb/s	19.3.6	ERP3:M	Yes □ No □ N/A □
ERP20	Able to automatically detect format of long preamble Barker, short preamble Barker, and OFDM and receive appropriately	19.3.6	CF9:M	Yes □ No □ N/A □
ERP21	Comply with local regulatory frequency allocation requirements	19.4.2	CF9:M	Yes □ No □ N/A □
ERP22	Use frequency plan for 2.4 GHz	19.4.3	CF9:M	Yes □ No □ N/A □
ERP23	Comply with regulatory spurious emissions regulations	19.4.4	CF9:M	Yes □ No □ N/A □
ERP24	Slot time requirements	19.4.5	CF9:M	Yes □ No □ N/A □

B.4.11 ERP functions *(continued)*

Item	PHY features	References	Status	Support
ERP25	Implement Short Slot Time option	19.4.5	CF9:O	Yes □ No □ N/A □
ERP26	Use 10 μs short interframe space (SIFS) time	19.4.7	CF9:M	Yes □ No □ N/A □
ERP27	Comply with regulatory transmit power requirements	19.4.8.2	CF9:M	Yes □ No □ N/A □
ERP28	± 25 PPM frequency tolerance	19.4.8.3	CF9:M	Yes □ No □ N/A □
ERP29	Use locked clocks	19.4.8.3, 19.4.8.4	CF9:M	Yes □ No □ N/A □
ERP30	Tolerate input level of –20 dBm	19.5.4	CF9:M	Yes □ No □ N/A □
ERP31	Use specified transmit mask	19.5.5	CF9:M	Yes □ No □ N/A □
ERP32	Meet sensitivity for all supported data rates	19.5.2	CF9:M	Yes □ No □ N/A □
ERP33	Reject adjacent channels as in Table 18-14 in 18.3.10.2 or in 17.4.8.4 as appropriate	19.5.3	CF9:M	Yes □ No □ N/A □
ERP34	Coherent transition of ERP-DSSS to OFDM	19.7.3, 19.7.3.8	ERP3:M	Yes □ No □ N/A □
ERP35	Same signal shaping of ERP-DSSS and OFDM	19.7.3.2	ERP3:M	Yes □ No □ N/A □
ERP36	Transmit power equal for ERP-DSSS and OFDM segments	19.7.3.3	ERP3:M	Yes □ No □ N/A □
ERP37	Align transition time	19.7.3.4	ERP3:M	Yes □ No □ N/A □
ERP38	Set transition phase to 45 degrees	19.7.3.4	ERP3:M	Yes □ No □ N/A □
ERP39	Calculate ERP-OFDM TXTIME	19.8.3.2	CF9:M	Yes □ No □ N/A □
ERP40	Calculate ERP-PBCC TXTIME	19.8.3.3	ERP2:M	Yes □ No □ N/A □
ERP41	Calculate DSSS-OFDM TXTIME	19.8.3.4	ERP3:M	Yes □ No □ N/A □
ERP42	Revert to long slot time when establishing association with a long slot time STA	8.4.1.4	CF9:M	Yes □ No □ N/A □
ERP43	Support TXVECTOR and RXVECTOR as described in reference	9.3	CF9:M	Yes □ No □ N/A □
ERP44	Terminate single carrier segment smoothly	19.7.3.5	ERP3:M	Yes □ No □ N/A □

B.4.12 Spectrum management extensions

Item	IUT configuration	References	Status	Support
SM1	Country, Power Constraint, and transmit power control (TPC) Report elements included in Beacon and Probe Response frames	8.3.3.2, 8.3.3.10, 8.4.2.10, 8.4.2.14, 8.4.2.17	CF10: M	Yes □ No □ N/A □
SM2	Spectrum Management Capability bit	8.4.1.4	CF10:M	Yes □ No □ N/A □
SM3	Power Capability and Supported Channels elements in Association and Reassociation frames	8.3.3.5, 8.3.3.6, 10.6.1	CF10:M	Yes □ No □ N/A □
SM4	Action frame protocol for spectrum management actions	8.4.1.11, 8.5	CF10:M	Yes □ No □ N/A □
SM4.1	Measurement Request frame	8.5.2.2	CF10:M	Yes □ No □ N/A □
SM4.2	Measurement Report frame	8.5.2.3	CF10:M	Yes □ No □ N/A □
SM4.3	TPC Request frame	8.5.2.4	CF10:M	Yes □ No □ N/A □
SM4.4	TPC Report frame	8.5.2.5	CF10:M	Yes □ No □ N/A □
SM4.5	Channel Switch Announcement frame	8.5.2.6	CF10:M	Yes □ No □ N/A □
SM5	Measurement requests			
SM5.1	Basic request	8.4.2.23.2	CF10:M	Yes □ No □ N/A □
SM5.2	CCA request	8.4.2.23.3	CF10:O	Yes □ No □ N/A □
SM5.3	Receive power indication (RPI) histogram	8.4.2.23.4	CF10:O	Yes □ No □ N/A □
SM5.4	Enabling/disabling requests and reports	8.4.2.23	CF10:M	Yes □ No □ N/A □
SM6	Measurement reports			
SM6.1	Basic report	8.4.2.24.2	CF10:M	Yes □ No □ N/A □
SM6.2	CCA report	8.4.2.24.3	CF10:O	Yes □ No □ N/A □
SM6.3	RPI histogram report	8.4.2.24.4	CF10:O	Yes □ No □ N/A □
SM6.4	Refusal to measure	8.4.2.24	CF10:M	Yes □ No □ N/A □
SM7	Quiet interval			
SM7.1	AP-defined Quiet interval	8.3.3.2, 8.3.3.10, 8.4.2.25, 10.6.2	(CF1 and CF10):M	Yes □ No □ N/A □
SM7.2	STA-defined Quiet interval	8.3.3.2, 8.3.3.10, 8.4.2.25, 10.6.2	(CF2.1 and CF10):M	Yes □ No □ N/A □
SM7.3	STA support for Quiet interval	8.3.3.2, 8.3.3.10, 8.4.2.25, 10.6.2	CF10:M	Yes □ No □ N/A □
SM8	Association control based on spectrum management capability	10.5, 10.6	(CF1 and CF10):M	Yes □ No □ N/A □
SM9	Association control based on transmit power capability	10.8.2	(CF1 and CF10):M	Yes □ No □ N/A □

B.4.12 Spectrum management extensions *(continued)*

Item	IUT configuration	References	Status	Support
SM10	Maximum transmit power levels			
SM10.1	AP determination and communication of local maximum transmit power level	10.8.4	(CF1 and CF10):M	Yes □ No □ N/A □
SM10.2	STA determination and communication of local maximum transmit power level	10.8.4	((CF2.1 or CF2.2) and CF10):M	Yes □ No □ N/A □
SM11	Selection of transmit power	10.8.5	CF10:M	Yes □ No □ N/A □
SM12	Adaptation of transmit power			
SM12.1	TPC report in Beacon and Probe Response frames	10.8.6	CF10:M	Yes □ No □ N/A □
SM13.1	Dynamic transmit power adaptation	10.8.6	CF10:O	Yes □ No □ N/A □
SM13	Testing channels for radars	10.9.4	CF10:M	Yes □ No □ N/A □
SM14	Detecting and discontinuing operations after detection of a radar	10.9.5	CF10:M	Yes □ No □ N/A □
SM15	Requesting and reporting of measurements	10.9.7	CF10:M	Yes □ No □ N/A □
SM16	Autonomous reporting of radars	10.9.7	CF10:M	Yes □ No □ N/A □
SM17	IBSS dynamic frequency selection (DFS) element including channel map	8.4.2.26	(CF2.2 and CF10):M	Yes □ No □ N/A □
SM18	DFS owner function	10.9.8	(CF2.2 and CF10):M	Yes □ No □ N/A □
SM19	DFS owner recovery procedure	10.9.8	(CF2.2 and CF10):M	Yes □ No □ N/A □
SM20	Channel switch procedure			
SM20.1	Transmission of channel switch announcement and channel switch procedure by an AP	10.9.8	(CF1 and CF10):M	Yes □ No □ N/A □
SM20.2	Transmission of channel switch announcement and channel switch procedure by a STA	10.9.8	(CF2.1 and CF10):M	Yes □ No □ N/A □
SM20.3	Reception of channel switch announcement and channel switch procedure by a STA	10.9.8	CF10:M	Yes □ No □ N/A □

B.4.13 Operating Classes extensions

Item	Protocol capability	References	Status	Support
RC1	Operating and coverage classes	8.4.2.10	CF8&CF11:M	Yes □ No □ N/A □
RC2	Operating and coverage classes (20 MHz channel spacing)	8.4.2.10, 18.3.8.7	CF8&CF11:M	Yes □ No □ N/A □
RC3	Operating and coverage classes (10 MHz channel spacing)	8.4.2.10, 18.3.8.7	CF8&CF11& OF1.7:M	Yes □ No □ N/A □
RC4	Operating and coverage classes (5 MHz channel spacing)	8.4.2.10, 18.3.8.7	CF8&CF11& OF1.8:M	Yes □ No □ N/A □
RC5	Coverage classes 0–31	9.18.6	CF15:M	Yes □ No □ N/A □
	Coverage class operation when not associated	9.18.6	CF15:M	Yes □ No □ N/A □
RC6	Power level, equivalent maximum transmit power level and operating class	9.18.6	CF15:M	Yes □ No □ N/A □
	Power level operation when not associated	9.18.6	CF15:M	Yes □ No □ N/A □
RC7	Power level, different maximum transmit power level and operating class	9.18.6	CF15:M	Yes □ No □ N/A □
	Power level operation when not associated	9.18.6	CF15:M	Yes □ No □ N/A □

B.4.14 QoS base functionality

Item	Protocol capability	References	Status	Support
QB1	QoS frame format	8.3.1.2–8.3.1.4, 8.3.2.1, 8.3.3.2, 8.3.3.5–8.3.3.8, 8.3.3.10, 8.3.3.13	CF12:M	Yes □ No □ N/A □
QB2	Per traffic identifier (TID) duplicate detection	8.2.4.4, 8.2.4.5, 9.3.2.10	CF12:M	Yes □ No □ N/A □
QB3	Decode of no-acknowledgment policy in QoS data frames	8.2.4.5.4, 9.19.2.4, 9.19.2.5, 9.19.4.2, 9.19.4.3	CF12:M	Yes □ No □ N/A □
QB4	Block Acknowledgments (Block Acks)			
QB4.1	Immediate Block Ack	8.3.1.8.1, 8.3.1.8.2, 8.3.1.9.1, 8.3.1.9.2, 8.5.5, 9.21 (except 9.21.7 and 9.21.8), 10.5	CF12:O CF16:M	Yes □ No □ N/A □

B.4.14 QoS base functionality *(continued)*

Item	Protocol capability	References	Status	Support
*QB4.2	Delayed Block Ack	8.3.1.8.1, 8.3.1.8.2, 8.3.1.9.1, 8.3.1.9.2, 8.5.5, 9.21 (except 9.21.7 and 9.21.8), 10.5	CF12:O	Yes □ No □ N/A □
QB4.3	Compressed Block Ack	8.3.1.8.3	CF12:O CF16:M	Yes □ No □ N/A □
QB4.4	MultiTID Block Ack	8.3.1.8.4	CF12:O CF16:M	Yes □ No □ N/A □
QB5	Automatic power save delivery (APSD)	8.5.3, 10.2.1	(CF1 and CF12):O (CF2 and CF12):O	Yes □ No □ N/A □
QB6	Direct-link setup (DLS)	8.4.2.21, 8.5.4, 6.3.14, 10.7	(CF1 AND CF12):M (CF2.1 AND CF12):O	Yes □ No □ N/A □

B.4.15 QoS enhanced distributed channel access (EDCA)

Item	Protocol capability	References	Status	Support
QD1	Support for four transmit queues with a separate channel access entity associated with each	9.2.4.2, 9.19.2.1	CF12:M	Yes □ No □ N/A □
QD2	Per-channel access function differentiated channel access	9.19.2.2, 9.19.2.3, 9.19.2.5	CF12:M	Yes □ No □ N/A □
QD3	Multiple frame transmission support	9.19.2.4	CF12:O	Yes □ No □ N/A □
QD4	Maintenance of within-queue ordering, exhaustive retransmission when sending non-QoS data frames	9.19.2.6	CF12:M	Yes □ No □ N/A □
QD5	Interpretation of admission control mandatory (ACM) bit in EDCA Parameter Set element	8.4.2.15, 9.19.4.2	(CF2.1 & CF12):M	Yes □ No □ N/A □
QD6	Contention-based admission control	9.19.4.2, 8.4.2.16, 8.4.2.17, 8.5.3.2–8.5.3.4, 10.4	(CF1 & CF12):O (CF2.1 & CF12):O	Yes □ No □ N/A □
QD7	Power management in an infrastructure BSS or in an IBSS	10.2	(CF1 and CF12):O (CF2 and CF12):O	Yes □ No □ N/A □
QD8	Default EDCA parameters for communications outside context of BBS	8.4.2.31, 9.19.2.2	CF2.3:M	Yes □ No □ N/A □

B.4.16 QoS hybrid coordination function (HCF) controlled channel access (HCCA)

Item	Protocol Capability	References	Status	Support
QP1	Traffic specification (TSPEC) and associated frame formats	8.5.3	(CF1 and CF12):M (CF2 and CF12):M	Yes □ No □ N/A □
QP2	HCCA rules	9.2.4.3, 9.19.3, 9.19.3.2–9.19.3.5	(CF1 and CF12):M (CF2 and CF12):M	Yes □ No □ N/A □
QP3	HCCA schedule generation and management	9.19.4	(CF1 & CF12):M	Yes □ No □ N/A □
QP4	HCF frame exchange sequences	9.19.2, 9.4.3	(CF1 and CF12):M (CF2 and CF12):M	Yes □ No □ N/A □
QP5	Traffic stream (TS) management	10.4	(CF1 and CF12):M (CF2 and CF12):M	Yes □ No □ N/A □
QP6	Minimum TSPEC parameter set	9.19.4	(CF1 and CF12):M (CF2 and CF12):M	Yes □ No □ N/A □
QP7	Power management in an infrastructure BSS	10.2.1.5, 10.2.1.6, 10.2.1.7, 10.2.1.8, 10.2.1.9, 10.2.1.10, 10.2.1.11	(CF1 and CF12):M (CF2 and CF12):M	Yes □ No □ N/A □

B.4.17 Radio Management extensions

Item	Protocol Capability	References	Status	Support
	Are the following Radio Measurement capabilities supported?			
RM1	Radio Measurement Capability	8.4.1.4	CF13:M	Yes □ No □ N/A □
RM2	Action frame protocol for measurements	8.5	CF13:M	Yes □ No □ N/A □
RM2.1	Radio Measurement Request frame	8.5.7.2	CF13:M	Yes □ No □ N/A □
RM2.2	Radio Measurement Report frame	8.5.7.3	CF13:M	Yes □ No □ N/A □
RM2.3	Link Measurement Request frame	8.5.7.4	CF13:M	Yes □ No □ N/A □
RM2.4	Link Measurement Report frame	8.5.7.5	CF13:M	Yes □ No □ N/A □
RM2.5	Neighbor Report Request			
RM2.5.1	Generate and transmit Neighbor Report Request	8.5.7.6	(CF13 AND CF2.1):M	Yes □ No □ N/A □
RM2.5.2	Receive and process Neighbor Report Request	8.5.7.6	(CF13 AND CF1):M	Yes □ No □ N/A □
RM2.6	Neighbor Report Response			

B.4.17 Radio Management extensions *(continued)*

Item	Protocol Capability	References	Status	Support
RM2.6.1	Generate and transmit Neighbor Report Response	8.5.7.7, 8.4.2.39	(CF13 AND CF1):M	Yes □ No □ N/A □
RM2.6.2	Receive and process Neighbor Report Response	7.4.8.5.7.7, 8.4.2.39	(CF13 AND CF2.1):M	Yes □ No □ N/A □
RM3	General protocol for requesting and reporting of measurements	8.4.2.23, 8.4.2.24, 10.11, 10.11.6	CF13:M	Yes □ No □ N/A □
RM3.1	Parallel Measurements	8.4.2.23, 10.11.6, 8.4.2.24	CF13:M	Yes □ No □ N/A □
RM3.2	Use of Enable, Request and Report bits to enable/ disable measurement requests and triggered autonomous reports Measurement Requests	8.4.2.23, 10.11.8, 10.11.6	CF13:M	Yes □ No □ N/A □
RM3.3	Enable Autonomous Report	8.4.2.23, 10.11.8	CF13:M	Yes □ No □ N/A □
RM3.4	Duration Mandatory	8.4.2.23, 10.11.4	CF13:M	Yes □ No □ N/A □
RM3.5	Incapable Indication	8.4.2.24	CF13:M	Yes □ No □ N/A □
RM3.6	Refused Indication	8.4.2.24, 10.11.5	CF13:M	Yes □ No □ N/A □
RM3.7	Repeated Measurement	8.5.7.2, 10.11.7	CF13:M	Yes □ No □ N/A □
RM3.8	Measurement pause	8.4.2.23.12, 10.11.9.7	CF13:M	Yes □ No □ N/A □
RM4	Beacon Measurement Type	10.11, 10.11.9.1	CF13:M	Yes □ No □ N/A □
RM4.1	Beacon Request	8.4.2.23.7	CF13:M	Yes □ No □ N/A □
RM4.2	Passive Measurement mode	8.4.2.23.7, 10.11.9.1	CF13:M	Yes □ No □ N/A □
RM4.3	Active Measurement mode	8.4.2.23.7, 10.11.9.1	CF13:M	Yes □ No □ N/A □
RM4.4	Beacon table mode	8.4.2.23.7, 10.11.9.1	CF13:M	Yes □ No □ N/A □
RM4.5	Reporting Conditions	8.4.2.23.7	CF13:O	Yes □ No □ N/A □
RM4.6	Beacon Report	8.4.2.23.7	CF13:M	Yes □ No □ N/A □
RM4.7	Reporting Detail	8.4.2.23.7, 8.4.2.24.7, 8.4.2.38	CF13:O	Yes □ No □ N/A □
* RM5	Frame Measurement Type	10.11, 10.11.9.2	CF13:O	Yes □ No □ N/A □
RM5.1	Frame request	8.4.2.23.8	(CF13 AND RM5):M	Yes □ No □ N/A □
RM5.2	Frame Report	8.4.2.24.8	(CF13 AND RM5):M	Yes □ No □ N/A □

B.4.17 Radio Management extensions *(continued)*

Item	Protocol Capability	References	Status	Support
RM6	Channel Load Measurement Type	10.11, 10.11.9.3	CF13:M	Yes □ No □ N/A □
RM6.1	Channel Load Request	8.4.2.23.5	CF13:M	Yes □ No □ N/A □
RM6.2	Channel Load Report	8.4.2.24.5	CF13:M	Yes □ No □ N/A □
RM7	Noise Histogram Measurement Type	10.11, 10.11.9.4	CF13:M	Yes □ No □ N/A □
RM7.1	Noise Histogram Request	8.4.2.23.6	CF13:M	Yes □ No □ N/A □
RM7.2	Noise Histogram Report	8.4.2.24.6	CF13:M	Yes □ No □ N/A □
RM8	STA Statistics Measurement Type	10.11, 10.11.9.5	CF13:M	Yes □ No □ N/A □
RM8.1	STA Statistics Request	8.4.2.23.9	CF13:M	Yes □ No □ N/A □
RM8.2	STA Statistics Report	8.4.2.24.9	CF13:M	Yes □ No □ N/A □
RM9	LCI Measurement Type	10.11, 10.11.9.6	CF13:M	Yes □ No □ N/A □
RM9.1	LCI Request	8.4.2.23.10	CF13:M	Yes □ No □ N/A □
RM9.1.1	Location Subject	8.4.2.23.10	CF13:M	Yes □ No □ N/A □
RM9.1.1.1	Location Subject third party	8.4.2.23.10	CF13:O	Yes □ No □ N/A □
RM9.1.2	Latitude Requested Resolution	8.4.2.23.10	CF13:M	Yes □ No □ N/A □
RM9.1.3	Longitude Requested Resolution	8.4.2.23.10	CF13:M	Yes □ No □ N/A □
RM9.1.4	Altitude Requested Resolution	8.4.2.23.10	CF13:M	Yes □ No □ N/A □
RM9.2	LCI Report	8.4.2.24.10	CF13:M	Yes □ No □ N/A □
RM9.3	Azimuth	10.11, 10.11.9.6	CF13:O	Yes □ No □ N/A □
RM9.3.1	Azimuth Request	8.4.2.23.10	CF13:O	Yes □ No □ N/A □
RM9.3.2	Azimuth Response	8.4.2.24.10	CF13:O	Yes □ No □ N/A □
*RM10	Transmit Stream/ Category Measurement Type	10.11, 10.11.9.8	(CF13 AND CF12):O	Yes □ No □ N/A □
RM10.1	Transmit Stream/ Category Measurement Request	8.4.2.23.11	RM10:M	Yes □ No □ N/A □
RM10.2	Transmit Stream/ Category Measurement Report	8.4.2.24.11	RM10:M	Yes □ No □ N/A □
RM10.3	Triggered Transmit Stream/Category Measurement Report	8.4.2.24.11, 10.11.9.8	RM10:O	Yes □ No □ N/A □
RM11	AP Channel Report	8.4.2.10, 8.4.2.38	(CF13 AND CF1):M	Yes □ No □ N/A □

B.4.17 Radio Management extensions *(continued)*

Item	Protocol Capability	References	Status	Support
RM11.1	Generate and transmit AP Channel Report	8.3.3.2, 8.3.3.10, 8.4.2.38	(CF13 AND CF1):M	Yes □ No □ N/A □
RM11.2	Receive and process AP Channel Report	8.3.3.2, 8.3.3.10, 8.4.2.38	(CF13 AND CF2.1):M	Yes □ No □ N/A □
RM12	Neighbor Report Procedure	10.11, 10.11.10	CF13:M	Yes □ No □ N/A □
RM12.1	Neighbor Report Procedure	10.11.10.2, 10.11.10.3	CF13:M	Yes □ No □ N/A □
RM12.2	TSF Offset in Neighbor Report	8.4.2.39, 10.11.10.3	CF13:O	Yes □ No □ N/A □
RM13	RCPI Measurement			
RM13.1	RCPI Measurement for DSSS PHY at 2.4 GHz	16.4.8.6	(CF13 AND CF4):M	Yes □ No □ N/A □
RM13.2	RCPI Measurement for OFDM PHY at 5 GHz	18.2.3.6, 18.3.10.7, 18.5.4.4, 18.5.5.9	(CF13 AND CF6):M	Yes □ No □ N/A □
RM13.3	RCPI Measurement for HR DSSS PHY at 2.4 GHz	17.4.5.17, 17.4.8.6	(CF13 AND CF7):M	Yes □ No □ N/A □
RM13.4	RCPI Measurement for Extended Rate PHY at 2.4 Ghz	19.9.5.15	(CF13 AND CF9):M	Yes □ No □ N/A □
RM14	RCPI Measurement during Active Scanning			
RM14.1	Respond with RCPI element when requested	10.1.4.3.3	(CF13 AND CF12 AND CF1):M	Yes □ No □ N/A □
RM14.2	Measurement of RCPI on Probe Request frames	10.1.4.3.3	(CF13 AND CF12 AND CF1):O	Yes □ No □ N/A □
RM15	RSNI Measurement	8.4.2.43	(CF13 AND RM13):M	Yes □ No □ N/A □
RM16	TPC Information in Beacon and Probe Response frames			
RM16.1	Country and TPC Report elements included in Beacon and Probe Response frames	8.3.3.2, 8.3.3.10, 8.4.2.10, 8.4.2.19, 10.8	CF13:M	Yes □ No □ N/A □
RM16.2	Power Constraint element included in Beacon and Probe Response frames	8.3.3.2, 8.3.3.10, 8.4.2.16	CF13:O	Yes □ No □ N/A □
RM17	Power Capability elements in Association and Reassociation frames	8.3.3.5, 8.3.3.6, 10.9.2	CF13:M	Yes □ No □ N/A □
RM18	Management Information Base			

B.4.17 Radio Management extensions *(continued)*

Item	Protocol Capability	References	Status	Support
RM18.1	dot11RadioMeasurement	Annex C	(CF13 AND CF1):M	Yes □ No □ N/A □
RM18.2	dot11SMTRMRequest	Annex C	(CF13 AND CF1):O	Yes □ No □ N/A □
RM18.3	dot11SMTRMReport	Annex C	(CF13 AND CF1):O	Yes □ No □ N/A □
RM18.4	dot11SMTRMConfig	Annex C	(CF13 AND CF1):O	Yes □ No □ N/A □
RM19	Measurement Pilot Frame	8.4.1.18, 8.4.2.48, 6.3.32, 10.8, 10.11.14, 10.11.15	CF13:O	Yes □ No □ N/A □
RM20	BSS Average Access Delay elements included in Beacon and Probe Response frames	8.3.3.2, 8.3.3.10, 8.4.2.41	(CF1AND CF13):M	Yes □ No □ N/A □
RM21	Antenna elements included in Beacon and Probe Response frames	8.3.3.2, 8.3.3.10, 8.4.2.42	CF13:M	Yes □ No □ N/A □
RM22	Measurement Pilot Transmission element and Multiple BSSID element, if required, included in Probe Response frame	8.3.3.10, 8.4.2.44, 8.4.2.48	CF13:O	Yes □ No □ N/A □
RM23	Quiet interval			
RM23.1	AP-defined Quiet Interval	8.3.3.2, 8.3.3.10, 8.4.2.25, 10.9.3	(CF1 AND CF13):M	Yes □ No □ N/A □
RM23.2	STA-defined Quiet Interval	8.3.3.2, 8.3.3.10, 8.4.2.25, 10.9.3	(CF2.1 AND CF13):M	Yes □ No □ N/A □
RM23.3	STA support for Quiet Interval	8.3.3.2, 8.3.3.10, 8.4.2.25, 10.9.3	CF13:M	Yes □ No □ N/A □
RM24	BSS Available Admission Capacity	8.4.2.45	(CF1 AND CF12 AND CF13):M	Yes □ No □ N/A □
RM25	BSS AC Access Delay	8.3.3.2, 8.3.3.10, 8.4.2.46	(CF1 AND CF12 AND CF13):M	Yes □ No □ N/A □

B.4.18 DSE functions

Item	Protocol capability	References	Status	Support
*DSE1	Fixed STA operation with RegLoc	10.12.3	CF15:O.1	Yes □ No □ N/A □
*DSE2	Enabling STA operation with RegLoc	10.12.3	CF15:O.1	Yes □ No □ N/A □
DSE2.1	Enabling STA creation of DSE service area	10.12.4	DSE2:M	Yes □ No □ N/A □
DSE2.2	Enabling STA operation with DSE	10.12.3	DSE2:M	Yes □ No □ N/A □
*DSE3	Dependent STA operation with DSE	10.12.5	CF15:O.1	Yes □ No □ N/A □
DSE3.1	Dependent STA enablement	10.12.5	DSE3:M	Yes □ No □ N/A □
DSE3.2	Dependent STA DSE time to enablement	10.12.5	DSE3:M	Yes □ No □ N/A □
DSE3.3	Dependent STA DSE time to not transmit	10.12.5	DSE3:M	Yes □ No □ N/A □
DSE3.4	Dependent STA DSE Registered Location Announcement frame	10.12.5	DSE3:M	Yes □ No □ N/A □
DSE3.5	Dependent STA MLME-ASSOCIATE.response primitive DSE	6.3.7.5	DSE3:M	Yes □ No □ N/A □
DSE3.6	Dependent STA MLME-REASSOCIATE.response primitive DSE	6.3.8.5	DSE3:M	Yes □ No □ N/A □
DSE4	DSE request report procedure Transmission of DSE measurement request by an AP Transmission of DSE measurement report by a STA	10.12.5 10.12.5	(CF15&CF1):M (CF15&CF2.1):M	Yes □ No □ N/A □ Yes □ No □ N/A □
DSE5	STA association procedure Transmission of Association Request frame with Supported Operating Classes element by a STA Transmission of Association Response frame with Supported Operating Classes element by an AP	9.18.5, 10.3.5.2 9.18.5, 10.3.5.3	(CF15&CF2.1):M (CF15&CF1):M	Yes □ No □ N/A □ Yes □ No □ N/A □
DSE6	STA reassociation procedure Transmission of Reassociation Request frame with Supported Operating Classes element by a STA Transmission of Reassociation Response frame with Supported Operating Classes element by an AP	9.18.5, 10.3.5.4 9.18.5, 10.3.5.5	(CF15&CF2.1):M (CF15&CF1):M	Yes □ No □ N/A □ Yes □ No □ N/A □
DSE7	Probe request procedure Transmission of Probe Request frame with Supported Operating Classes element by a STA	10.10.1	CF15&CF2.1:M	Yes □ No □ N/A □
DSE8	Probe response procedure Transmission of Probe Response frame with Supported Operating Classes element by an AP	10.10.1	CF15&CF1:M	Yes □ No □ N/A □

B.4.18 DSE functions *(continued)*

Item	Protocol capability	References	Status	Support
DSE9	Extended channel switch procedure	10.10.3	(CF15&CF1):M	Yes □ No □ N/A □
	Transmission of extended channel switch announcement and channel switch procedure by an AP	10.10.3		
	Transmission of extended channel switch announcement and channel switch procedure by a STA	10.10.3	(CF15&CF2.1): M	Yes □ No □ N/A □
	Reception of extended channel switch announcement and channel switch procedure by a STA		CF15:M	Yes □ No □ N/A □
DSE10	DSE power constraint procedure	10.12.5	(CF15&CF1):M	Yes □ No □ N/A □
	Transmission of DSE power constraint announcement by an enabling STA	10.12.5		
	Reception of DSE power constraint announcement by a dependent STA		CF15:M	Yes □ No □ N/A □

B.4.19 High-throughput (HT) features

B.4.19.1 HT MAC features

Item	Protocol capability	References	Status	Support
	Are the following MAC protocol features supported?			
HTM1	HT capabilities signaling			
HTM1.1	HT Capabilities element	8.4.2.58.1	CF16:M	Yes □ No □ N/A □
HTM1.2	Signaling of STA capabilities in Probe Request, (Re)Association Request frames	8.4.2.58, 8.3.3.9, 8.3.3.5, 8.3.3.7	(CF16 and CF2):M	Yes □ No □ N/A □
HTM1.3	Signaling of STA and BSS capabilities in Beacon, Probe Response, (Re)Association Response frames	8.4.2.58, 8.3.3.2, 8.3.3.10, 8.3.3.6, 8.3.3.8	(CF16 and CF1):M	Yes □ No □ N/A □
HTM2	Signaling of HT operation	8.4.2.59	(CF16 and CF1):M	Yes □ No □ N/A □
HTM3	MPDU aggregation			
HTM3.1	Reception of A-MPDU	8.4.2.58.3, 11.4, 9.12.2	CF16:M	Yes □ No □ N/A □
HTM3.2	A-MPDU format	8.6.1	CF16:M	Yes □ No □ N/A □
HTM3.3	A-MPDU contents	8.6.3	CF16:M	Yes □ No □ N/A □
HTM3.4	A-MPDU frame exchange sequences	9.19.2.4	CF16:M	Yes □ No □ N/A □
HTM3.5	Transmission of A-MPDU	8.4.2.58.3, 11.4	CF16:O	Yes □ No □ N/A □
HTM4	MSDU aggregation			
HTM4.1	Reception of A-MSDUs	8.2.4.5, 8.3.2.2	CF16:M	Yes □ No □ N/A □
HTM4.2	A-MSDU format	8.3.2.2	CF16:M	Yes □ No □ N/A □
HTM4.3	A-MSDU content	8.3.2.2	CF16:M	Yes □ No □ N/A □
HTM4.4	Transmission of A-MSDUs	8.3.2.2, 8.2.4.5	CF16:O	Yes □ No □ N/A □

B.4.19.1 HT MAC features *(continued)*

Item	Protocol capability	References	Status	Support
HTM5	Block Ack			
HTM5.1	Block Ack mechanism	8.3.1.8, 8.3.1.9, 8.4.1.14, 9.21, 10.15	CF16:M	Yes □ No □ N/A □
HTM5.2	Use of compressed bitmap between HT STAs	8.3.1.9.3, 9.21.6,	CF16:M	Yes □ No □ N/A □
HTM5.3	HT-immediate Block Ack extensions	9.21.7	CF16:M	Yes □ No □ N/A □
HTM5.4	HT-delayed Block Ack extensions	9.21.8	CF16 and QB4.2:M	Yes □ No □ N/A □
HTM5.5	Multiple TID Block Ack	8.3.1.8.4, 8.3.1.9.4, 9.26.1.7	PC37:M	Yes □ No □ N/A □
HTM6	Protection mechanisms for different HT PHY options			
HTM6.1	Protection of RIFS PPDUs in the presence of non-HT STAs	9.23.3.3	CF16:M	Yes □ No □ N/A □
HTM6.1a	Protection of RIFS PPDUs in an IBSS	9.23.3.3	CF16:M	Yes □ No □ N/A □
HTM6.2	Protection of HT-greenfield PPDUs in the presence of non-HT STAs	9.23.3.1	HTP1.3:M	Yes □ No □ N/A □
HTM6.2a	Protection of HT-greenfield PPDUs in an IBSS	9.23.3.1	CF16:M	Yes □ No □ N/A □
*HTM7	L-SIG TXOP protection mechanism	9.23.5	CF16:O	Yes □ No □ N/A □
HTM7.1	Update NAV according to L-SIG	9.23.5.4	HTM7:M	Yes □ No □ N/A □
HTM8	Duration/ID rules for A-MPDU and TXOP	8.2.4.2	CF16:O	Yes □ No □ N/A □
HTM9	Truncation of TXOP as TXOP holder	9.19.2.7	CF16:O	Yes □ No □ N/A □
HTM10	Reception of +HTC frames	8.2.4.1.10, 8.4.2.58.5, 9.9	CF16:O	Yes □ No □ N/A □
*HTM11	Reverse direction (RD) aggregation exchanges	9.25	CF16:O	Yes □ No □ N/A □
HTM11.1	Constraints regarding responses	9.25.4	HTM11:M	Yes □ No □ N/A □
HTM12	Link adaptation			
HTM12.1	Use of the HT Control field for link adaptation in immediate response exchange	8.2.4.6, 8.3.3.14, 9.28.2	CF16:O	Yes □ No □ N/A □
HTM12.2	Link adaptation using explicit feedback mechanism	8.3.3.14, 9.29.3	CF16:O	Yes □ No □ N/A □
HTM13	Transmit beamforming			
*HTM13.1	Transmission of beamformed PPDUs	9.29	CF16:O	Yes □ No □ N/A □
*HTM13.2	Reception of beamformed PPDUs	9.29	CF16:O	Yes □ No □ N/A □
*HTM13.3	Initiate transmit beamforming frame exchange with implicit feedback	9.29.2	HTM13.1:O	Yes □ No □ N/A □
HTM13.3.1	Reception of sounding PPDUs	9.29.2	HTM13.3:M	Yes □ No □ N/A □
*HTM13.4	Response to transmit beamforming frame exchange with implicit feedback	9.29.2	HTM13.2:O	Yes □ No □ N/A □

B.4.19.1 HT MAC features *(continued)*

Item	Protocol capability	References	Status	Support
HTM13.4.1	Transmission of sounding PPDUs	9.29.2	HTM13.4:M	Yes □ No □ N/A □
*HTM13.5	Initiate transmit beamforming frame exchange with explicit feedback	8.5.12.6, 9.29.3	HTM13.1:O	Yes □ No □ N/A □
HTM13.5.1	Transmission of sounding PPDUs	9.29.3	HTM13.5:M	Yes □ No □ N/A □
*HTM13.6	Respond to transmit beamforming frame exchange with explicit feedback	9.29.3	HTM13.2:O	Yes □ No □ N/A □
HTM13.6.1	Transmission of Action No Ack +HTC frame including Action payload of type CSI	9.29.3, 8.5.12.6	HTM13.6:O.1	Yes □ No □ N/A □
HTM13.6.2	Transmission of Action No Ack +HTC frame including Action payload of type “Noncompressed beamforming”	9.29.3, 8.5.12.7	HTM13.6:O.1	Yes □ No □ N/A □
HTM13.6.3	Transmission of Action No Ack +HTC frame including Action payload of type “Compressed beamforming”	9.29.3, 8.5.12.8	HTM13.6:O.1	Yes □ No □ N/A □
*HTM13.7	Calibration procedure	8.3.3.14, 9.29.2.4	HTM13:O	Yes □ No □ N/A □
HTM14	Antenna selection (ASEL)	8.2.4.6, 8.4.2.58.7, 8.5.12.9, 9.30	CF16:O	Yes □ No □ N/A □
*HTM15	Null data packet (NDP)	9.31	CF16:O	Yes □ No □ N/A □
HTM16	Space-time block coding (STBC) support			
HTM16.1	STBC beacon transmission	10.1.3.2	HTP2.11:O	Yes □ No □ N/A □
HTM16.2	Dual CTS protection	9.3.2.7	HTP2.11:O	Yes □ No □ N/A □
HTM17	SM power save support			
*HTM17.1	AP support for dynamic and static SM power save mode	10.2.4	(CF16 and CF1):M	Yes □ No □ N/A □
*HTM17.2	STA support for dynamic and static SM power save mode	10.2.4	(CF16 and CF2):O	Yes □ No □ N/A □
HTM17.3	Transmit SM Power Save state information using HT capabilities, or SM Power Save frame	8.5.12.3, 10.2.4	(HTM17.1 OR HTM17.2):M	Yes □ No □ N/A □
HTM17.4	Receive SM Power Save state information and support frame exchanges with SM Power Save STAs	10.2.4	CF16:M	Yes □ No □ N/A □
HTM18	Mechanisms for coexistence of 20 MHz and 40 MHz channels	10.15	CF16:M	Yes □ No □ N/A □
HTM19	Channel selection methods for 20/40 MHz operation	10.15.3	(HTP2.3.4 and CF1):M	Yes □ No □ N/A □
HTM20	20/40 MHz operation	10.15	HTP2.3.4:M	Yes □ No □ N/A □
HTM21	Phased coexistence operation (PCO)			
*HTM21.1	PCO capability at AP	10.16	(CF16 and CF1):O	Yes □ No □ N/A □

B.4.19.1 HT MAC features *(continued)*

Item	Protocol capability	References	Status	Support
HTM21.1.1	Rules for operation at a PCO active AP	8.5.12.5, 10.16.2	HTM21.1:M	Yes □ No □ N/A □
*HTM21.2	STA support for PCO mode	10.16	(CF16 and CF2):O	Yes □ No □ N/A □
HTM21.2.1	Rules for operation at PCO active STA	8.5.12.5, 10.16.3	HTM21.2:M	Yes □ No □ N/A □
HTM22	Management information base (MIB)			
HTM22.1	dot11PhyHTComplianceGroup	Annex C	CF16:M	Yes □ No □ N/A □
HTM22.2	dot11PhyMCSGroup	Annex C	CF16:M	Yes □ No □ N/A □

B.4.19.2 HT PHY features

Item	Protocol capability	References	Status	Support
	Are the following PHY protocol features supported?			
HTP1	PHY operating modes			
HTP1.1	Operation according to 18 and/or Clause 19	20.1.4	CF16:M	Yes □ No □ N/A □
HTP1.2	HT-mixed format	20.1.4	CF16:M	Yes □ No □ N/A □
*HTP1.3	HT-greenfield format	20.1.4	CF16:O	Yes □ No □ N/A □
HTP2	PLCP frame format			
HTP2.1	HT-mixed format PLCP format	20.3.2	CF16:M	Yes □ No □ N/A □
HTP2.2	HT-greenfield PLCP format	20.3.2	HTP1.3:M	Yes □ No □ N/A □
HTP2.3	Modulation and coding schemes (MCS)			
HTP2.3.1	MCS 0 to MCS 7 in 20 MHz with 800 ns guard interval (GI)			
HTP2.3.1.1	Support for 20 MHz with 800 ns GI MCS index 0	20.3.5, 20.6	CF16:M	Yes □ No □ N/A □
HTP2.3.1.2	Support for 20 MHz with 800 ns GI MCS index 1	20.3.5, 20.6	CF16:M	Yes □ No □ N/A □
HTP2.3.1.3	Support for 20 MHz with 800 ns GI MCS index 2	20.3.5, 20.6	CF16:M	Yes □ No □ N/A □
HTP2.3.1.4	Support for 20 MHz with 800 ns GI MCS index 3	20.3.5, 20.6	CF16:M	Yes □ No □ N/A □
HTP2.3.1.5	Support for 20 MHz with 800 ns GI MCS index 4	20.3.5, 20.6	CF16:M	Yes □ No □ N/A □
HTP2.3.1.6	Support for 20 MHz with 800 ns GI MCS index 5	20.3.5, 20.6	CF16:M	Yes □ No □ N/A □
HTP2.3.1.7	Support for 20 MHz with 800 ns GI MCS index 6	20.3.5, 20.6	CF16:M	Yes □ No □ N/A □
HTP2.3.1.8	Support for 20 MHz with 800 ns GI MCS index 7	20.3.5, 20.6	CF16:M	Yes □ No □ N/A □

B.4.19.2 HT PHY features *(continued)*

Item	Protocol capability	References	Status	Support
HTP2.3.2	MCS 8 to MCS 15 in 20 MHz with 800 ns GI			
HTP2.3.2.1	Support for 20 MHz with 800 ns GI MCS index 8	20.3.5, 20.6	(CF16 and CF1):M	Yes □ No □ N/A □
HTP2.3.2.2	Support for 20 MHz with 800 ns GI MCS index 9	20.3.5, 20.6	(CF16 and CF1):M	Yes □ No □ N/A □
HTP2.3.2.3	Support for 20 MHz with 800 ns GI MCS index 10	20.3.5, 20.6	(CF16 and CF1):M	Yes □ No □ N/A □
HTP2.3.2.4	Support for 20 MHz with 800 ns GI MCS index 11	20.3.5, 20.6	(CF16 and CF1):M	Yes □ No □ N/A □
HTP2.3.2.5	Support for 20 MHz with 800 ns GI MCS index 12	20.3.5, 20.6	(CF16 and CF1):M	Yes □ No □ N/A □
HTP2.3.2.6	Support for 20 MHz with 800 ns GI MCS index 13	20.3.5, 20.6	(CF16 and CF1):M	Yes □ No □ N/A □
HTP2.3.2.7	Support for 20 MHz with 800 ns GI MCS index 14	20.3.5, 20.6	(CF16 and CF1):M	Yes □ No □ N/A □
HTP2.3.2.8	Support for 20 MHz with 800 ns GI MCS index 15	20.3.5, 20.6	(CF16 and CF1):M	Yes □ No □ N/A □
HTP2.3.3	Transmit and receive support for 400 ns GI	20.3.5, 20.6	CF16:O	Yes □ No □ N/A □
*HTP2.3.4	Operation at 40 MHz	20.3.5, 20.6	CF16:O	Yes □ No □ N/A □
HTP2.3.5	Support for MCS with indices 16 to 76			
HTP2.3.5.1	Support for MCS with index 16	20.3.5, 20.6	CF16:O	Yes □ No □ N/A □
HTP2.3.5.2	Support for MCS with index 17	20.3.5, 20.6	CF16:O	Yes □ No □ N/A □
HTP2.3.5.3	Support for MCS with index 18	20.3.5, 20.6	CF16:O	Yes □ No □ N/A □
HTP2.3.5.4	Support for MCS with index 19	20.3.5, 20.6	CF16:O	Yes □ No □ N/A □
HTP2.3.5.5	Support for MCS with index 20	20.3.5, 20.6	CF16:O	Yes □ No □ N/A □
HTP2.3.5.6	Support for MCS with index 21	20.3.5, 20.6	CF16:O	Yes □ No □ N/A □
HTP2.3.5.7	Support for MCS with index 22	20.3.5, 20.6	CF16:O	Yes □ No □ N/A □
HTP2.3.5.8	Support for MCS with index 23	20.3.5, 20.6	CF16:O	Yes □ No □ N/A □
HTP2.3.5.9	Support for MCS with index 24	20.3.5, 20.6	CF16:O	Yes □ No □ N/A □
HTP2.3.5.10	Support for MCS with index 25	20.3.5, 20.6	CF16:O	Yes □ No □ N/A □
HTP2.3.5.11	Support for MCS with index 26	20.3.5, 20.6	CF16:O	Yes □ No □ N/A □
HTP2.3.5.12	Support for MCS with index 27	20.3.5, 20.6	CF16:O	Yes □ No □ N/A □
HTP2.3.5.13	Support for MCS with index 28	20.3.5, 20.6	CF16:O	Yes □ No □ N/A □
HTP2.3.5.14	Support for MCS with index 29	20.3.5, 20.6	CF16:O	Yes □ No □ N/A □
HTP2.3.5.15	Support for MCS with index 30	20.3.5, 20.6	CF16:O	Yes □ No □ N/A □
HTP2.3.5.16	Support for MCS with index 31	20.3.5, 20.6	CF16:O	Yes □ No □ N/A □
HTP2.3.5.17	Support for MCS with index 32	20.3.5, 20.6	CF16:O	Yes □ No □ N/A □
HTP2.3.5.18	Support for MCS with index 33	20.3.5, 20.6	CF16:O	Yes □ No □ N/A □
HTP2.3.5.19	Support for MCS with index 34	20.3.5, 20.6	CF16:O	Yes □ No □ N/A □
HTP2.3.5.20	Support for MCS with index 35	20.3.5, 20.6	CF16:O	Yes □ No □ N/A □

B.4.19.2 HT PHY features *(continued)*

Item	Protocol capability	References	Status	Support
HTP2.3.5.21	Support for MCS with index 36	20.3.5, 20.6	CF16:O	Yes ☐ No ☐ N/A ☐
HTP2.3.5.22	Support for MCS with index 37	20.3.5, 20.6	CF16:O	Yes ☐ No ☐ N/A ☐
HTP2.3.5.23	Support for MCS with index 38	20.3.5, 20.6	CF16:O	Yes ☐ No ☐ N/A ☐
HTP2.3.5.24	Support for MCS with index 39	20.3.5, 20.6	CF16:O	Yes ☐ No ☐ N/A ☐
HTP2.3.5.25	Support for MCS with index 40	20.3.5, 20.6	CF16:O	Yes ☐ No ☐ N/A ☐
HTP2.3.5.26	Support for MCS with index 41	20.3.5, 20.6	CF16:O	Yes ☐ No ☐ N/A ☐
HTP2.3.5.27	Support for MCS with index 42	20.3.5, 20.6	CF16:O	Yes ☐ No ☐ N/A ☐
HTP2.3.5.28	Support for MCS with index 43	20.3.5, 20.6	CF16:O	Yes ☐ No ☐ N/A ☐
HTP2.3.5.29	Support for MCS with index 44	20.3.5, 20.6	CF16:O	Yes ☐ No ☐ N/A ☐
HTP2.3.5.30	Support for MCS with index 45	20.3.5, 20.6	CF16:O	Yes ☐ No ☐ N/A ☐
HTP2.3.5.31	Support for MCS with index 46	20.3.5, 20.6	CF16:O	Yes ☐ No ☐ N/A ☐
HTP2.3.5.32	Support for MCS with index 47	20.3.5, 20.6	CF16:O	Yes ☐ No ☐ N/A ☐
HTP2.3.5.33	Support for MCS with index 48	20.3.5, 20.6	CF16:O	Yes ☐ No ☐ N/A ☐
HTP2.3.5.34	Support for MCS with index 49	20.3.5, 20.6	CF16:O	Yes ☐ No ☐ N/A ☐
HTP2.3.5.35	Support for MCS with index 50	20.3.5, 20.6	CF16:O	Yes ☐ No ☐ N/A ☐
HTP2.3.5.36	Support for MCS with index 51	20.3.5, 20.6	CF16:O	Yes ☐ No ☐ N/A ☐
HTP2.3.5.37	Support for MCS with index 52	20.3.5, 20.6	CF16:O	Yes ☐ No ☐ N/A ☐
HTP2.3.5.38	Support for MCS with index 53	20.3.5, 20.6	CF16:O	Yes ☐ No ☐ N/A ☐
HTP2.3.5.39	Support for MCS with index 54	20.3.5, 20.6	CF16:O	Yes ☐ No ☐ N/A ☐
HTP2.3.5.40	Support for MCS with index 55	20.3.5, 20.6	CF16:O	Yes ☐ No ☐ N/A ☐
HTP2.3.5.41	Support for MCS with index 56	20.3.5, 20.6	CF16:O	Yes ☐ No ☐ N/A ☐
HTP2.3.5.42	Support for MCS with index 57	20.3.5, 20.6	CF16:O	Yes ☐ No ☐ N/A ☐
HTP2.3.5.43	Support for MCS with index 58	20.3.5, 20.6	CF16:O	Yes ☐ No ☐ N/A ☐
HTP2.3.5.44	Support for MCS with index 59	20.3.5, 20.6	CF16:O	Yes ☐ No ☐ N/A ☐
HTP2.3.5.45	Support for MCS with index 60	20.3.5, 20.6	CF16:O	Yes ☐ No ☐ N/A ☐
HTP2.3.5.46	Support for MCS with index 61	20.3.5, 20.6	CF16:O	Yes ☐ No ☐ N/A ☐
HTP2.3.5.47	Support for MCS with index 62	20.3.5, 20.6	CF16:O	Yes ☐ No ☐ N/A ☐
HTP2.3.5.48	Support for MCS with index 63	20.3.5, 20.6	CF16:O	Yes ☐ No ☐ N/A ☐
HTP2.3.5.49	Support for MCS with index 64	20.3.5, 20.6	CF16:O	Yes ☐ No ☐ N/A ☐
HTP2.3.5.50	Support for MCS with index 65	20.3.5, 20.6	CF16:O	Yes ☐ No ☐ N/A ☐
HTP2.3.5.51	Support for MCS with index 66	20.3.5, 20.6	CF16:O	Yes ☐ No ☐ N/A ☐
HTP2.3.5.52	Support for MCS with index 67	20.3.5, 20.6	CF16:O	Yes ☐ No ☐ N/A ☐
HTP2.3.5.53	Support for MCS with index 68	20.3.5, 20.6	CF16:O	Yes ☐ No ☐ N/A ☐
HTP2.3.5.54	Support for MCS with index 69	20.3.5, 20.6	CF16:O	Yes ☐ No ☐ N/A ☐
HTP2.3.5.55	Support for MCS with index 70	20.3.5, 20.6	CF16:O	Yes ☐ No ☐ N/A ☐
HTP2.3.5.56	Support for MCS with index 71	20.3.5, 20.6	CF16:O	Yes ☐ No ☐ N/A ☐
HTP2.3.5.57	Support for MCS with index 72	20.3.5, 20.6	CF16:O	Yes ☐ No ☐ N/A ☐
HTP2.3.5.58	Support for MCS with index 73	20.3.5, 20.6	CF16:O	Yes ☐ No ☐ N/A ☐
HTP2.3.5.59	Support for MCS with index 74	20.3.5, 20.6	CF16:O	Yes ☐ No ☐ N/A ☐
HTP2.3.5.60	Support for MCS with index 75	20.3.5, 20.6	CF16:O	Yes ☐ No ☐ N/A ☐

B.4.19.2 HT PHY features *(continued)*

Item	Protocol capability	References	Status	Support
HTP2.3.5.61	Support for MCS with index 76	20.3.5, 20.6	CF16:O	Yes □ No □ N/A □
HTP2.4	PHY timing parameters			
HTP2.4.1	Values in non-HT 20 MHz channel	20.3.6	CF16:M	Yes □ No □ N/A □
HTP2.4.2	Values in 20 MHz HT channel	20.3.6	CF16:M	Yes □ No □ N/A □
HTP2.4.3	Values in 40 MHz channel	20.3.6	HTP2.3.4:M	Yes □ No □ N/A □
HTP2.5	HT Preamble field definition and coding			
HTP2.5.1	HT-mixed format preamble	20.3.9.2	CF16:M	Yes □ No □ N/A □
HTP2.5.2	HT-greenfield preamble	20.3.9.5	HTP1.3:M	Yes □ No □ N/A □
HTP2.5.3	Extension HT Long Training fields (HT-ELTFs)	20.3.9.4.6	CF16:O	Yes □ No □ N/A □
HTP2.6	HT Data field definition and coding	20.3.11	CF16:M	Yes □ No □ N/A □
HTP2.6.1	Use of LDPC codes	20.3.11.7	CF16:O	Yes □ No □ N/A □
HTP2.7	Beamforming	20.3.12	CF16:O	Yes □ No □ N/A □
HTP2.8	Sounding PPDUs			
HTP2.8.1	HT preamble format for sounding PPDUs	20.3.13	CF16:O	Yes □ No □ N/A □
HTP2.8.2	Sounding with an NDP	20.3.13.2	HTM15:O	Yes □ No □ N/A □
HTP2.8.3	Sounding PPDU for calibration	20.3.13.3	HTM14.7:M	Yes □ No □ N/A □
HTP2.9	Channel numbering and channelization			
HTP2.9.1	Channel allocation for 20 MHz channels at 5 GHz	18.3.8.4	CF16:M	Yes □ No □ N/A □
HTP2.9.2	Channel allocation for 20 MHz channels at 2.4 GHz	19.4.3	CF16:M	Yes □ No □ N/A □
HTP2.9.3	Channel allocation for 40 MHz channels at 5 GHz	20.3.15.3	HTP2.3.4:M	Yes □ No □ N/A □
HTP2.9.4	Channel allocation for 40 MHz channels at 2.4 GHz	20.3.15.2	HTP2.3.4:M	Yes □ No □ N/A □
HTP2.10	PMD transmit specification			
HTP2.10.1	PMD transmit specification for 20 MHz channel	20.3.20	CF16:M	Yes □ No □ N/A □
HTP2.10.2	PMD transmit specification for 40 MHz channel	20.3.20	HTP2.3.4:M	Yes □ No □ N/A □
*HTP2.11	Space-time block coding (STBC)	20.3.11.9.2	CF16:O	Yes □ No □ N/A □
HTP2.12	PMD receive specification			
HTP2.12.1	PMD receive specification for 20 MHz channel	20.3.21	CF16:M	Yes □ No □ N/A □
HTP2.12.2	PMD receive specification for 40 MHz channel	20.3.21	HTP2.3.4:M	Yes □ No □ N/A □
HTP2.13	PPDU reception with RIFS	20.3.21.7	CF16:M	Yes □ No □ N/A □

B.4.20 Tunneled direct-link setup extensions

Item	Protocol capability	References	Status	Support
TDLS1	Tunneled direct-link setup	8.5.13, 10.22	CF2&CF18:M	Yes □ No □ N/A □
TDLS1.1	TDLS setup	8.4.2.64, 8.5.13.2, 8.5.13.3, 8.5.13.4, 10.22.4	CF2&CF18:M	Yes □ No □ N/A □
TDLS1.2	TDLS teardown	8.4.2.64, 8.5.13.5, 10.22.5	CF2&CF18:M	Yes □ No □ N/A □
TDLS1.3	TDLS Peer Key Handshake	11.6.9	CF2&CF18:M	Yes □ No □ N/A □
TDLS1.4	TDLS Peer PSM	8.4.2.64, 8.4.2.65, 8.5.13.9, 8.5.13.10, 10.2.1.14	CF2&CF18:O	Yes □ No □ N/A □
TDLS 1.5	TDLS Peer U-APSD	8.4.2.64, 8.4.2.67, 8.4.2.68, 8.5.13.6, 8.5.13.11, 10.2.1.15	CF2&CF18:O	Yes □ No □ N/A □
TDLS 1.6	TDLS Channel Switching	8.4.2.64, 8.4.2.66, 8.5.13.7, 8.5.13.8, 10.22.6	CF2&CF8& CF11&CF18:O	Yes □ No □ N/A □
TDLS1.7	TDLS Discovery	8.4.2.64, 8.5.13.12, 8.5.8.16, 10.22.3	CF2&CF8& CF11&CF18:O	Yes □ No □ N/A □

B.4.21 WNM extensions

Item	Protocol capability	References	Status	Support
WNM1	Extended Capabilities information element	8.4.2.29	CF19:M	Yes □ No □ N/A □
WNM2	STA Statistics (Triggered) and Multicast Diagnostics	10.11.8	CF19:M	Yes □ No □ N/A □
WNM2.1	Protocol for Triggered Measurements	10.11.8	CF19:M	Yes □ No □ N/A □
WNM2.2	Triggered STA Statistics	8.4.2.23.9, 8.4.2.24.9, 8.5.7.2,8.5.7.3, 10.11.9.5	CF19:O	Yes □ No □ N/A □
WNM2.3	Multicast Diagnostics	8.4.2.23.13, 8.4.2.24.12, 8.5.7.2,8.5.7.3, 10.11.19	CF19:M	Yes □ No □ N/A □
WNM3	Event	10.23.2	CF19:M	Yes □ No □ N/A □
WNM3.1	Event Request frame	8.4.2.69, 8.5.14.2	(CF19 & CF1):M	Yes □ No □ N/A □
WNM3.2	Event Report frame	8.4.2.70, 8.5.14.3	(CF19 & CF2):M	Yes □ No □ N/A □
WNM4	Diagnostic	10.23.3	CF19:M	Yes □ No □ N/A □
WNM4.1	Diagnostic Request frame	8.4.2.71, 8.5.14.4	(CF19 & CF1):M	Yes □ No □ N/A □
WNM4.2	Diagnostic Report frame	8.4.2.72, 8.5.14.5	(CF19 & CF2):M	Yes □ No □ N/A □
WNM4.3	Configuration Profile Diagnostic Type	8.4.2.72.3, 10.23.3.2	CF19:M	Yes □ No □ N/A □
WNM4.4	Manufacturer Information STA Report Diagnostic Type	8.4.2.72.2, 10.23.3.3	CF19:M	Yes □ No □ N/A □
WNM4.5	Association Diagnostic Type	8.4.2.71.2, 8.4.2.72.4, 10.23.3.4	CF19:M	Yes □ No □ N/A □
WNM4.6	IEEE 802.1X Authentication Diagnostic Type	8.4.2.71.3, 8.4.2.72.5, 10.23.3.5	(CF19 & PC34):M	Yes □ No □ N/A □
WNM5	Location	10.23.4, 8.4.2.73	CF19:M	Yes □ No □ N/A □
WNM5.1	Location Civic Request/Report	10.11.9.9	CF19:M	Yes □ No □ N/A □
WNM5.2	Location Identifier Request/Report	10.11.9.10	CF19:M	Yes □ No □ N/A □
WNM5.3	Location Track Notification	10.23.4, 8.5.8.17	CF19:O	Yes □ No □ N/A □
WNM5.3.1	Time of Departure Notifications	10.23.4, 8.4.2.73	CF19:O	Yes □ No □ N/A □
WNM5.3.2	Motion Detection Notifications	10.23.4, 8.4.2.73	CF19:O	Yes □ No □ N/A □
WNM5.4	Location Configuration Request frame	8.5.14.6, 8.4.2.73	CF19:M	Yes □ No □ N/A □

B.4.21 WNM extensions *(continued)*

Item	Protocol capability	References	Status	Support
WNM5.4.1	Normal Indication	8.5.14.6, 8.4.2.73	CF19:O	Yes □ No □ N/A □
WNM5.4.2	Motion Indication	8.5.14.6, 8.4.2.73	CF19:O	Yes □ No □ N/A □
WNM5.5	Location Configuration Response frame	8.5.14.7, 8.4.2.73	CF19:M	Yes □ No □ N/A □
*WNM6	Multiple BSSID Support	10.1.3.6, 10.1.4, 10.11.14	CF19:O	Yes □ No □ N/A □
WNM6.1	Multiple BSSID element	8.4.2.48	WNM6:M	Yes □ No □ N/A □
WNM6.2	Multiple BSSID-index element	8.4.2.76	WNM6:M	Yes □ No □ N/A □
WNM7	BSS Transition Management	10.23.6	CF19:O	Yes □ No □ N/A □
WNM7.1	Neighbor Report element	8.4.2.39	(CF19 & CF1):M	Yes □ No □ N/A □
WNM7.2	BSS Transition Management Query frame	8.5.14.8	(CF19 & CF1):M	Yes □ No □ N/A □
WNM7.3	BSS Transition Management Request frame	8.5.14.9	(CF19 & CF2):M	Yes □ No □ N/A □
WNM7.4	BSS Transition Management Response frame	8.5.14.10	(CF19 & CF2):M	Yes □ No □ N/A □
*WNM8	FMS	10.2.1.16	CF19:O	Yes □ No □ N/A □
WNM8.1	FMS Request frame	8.5.14.11	(CF2 & WNM8):M	Yes □ No □ N/A □
WNM8.2	FMS Response frame	8.5.14.12	(CF1 & WNM8):M	Yes □ No □ N/A □
WNM9	Proxy ARP	10.23.13	CF19:O	Yes □ No □ N/A □
*WNM10	Collocated Interference Reporting	10.23.9	CF19:O	Yes □ No □ N/A □
WNM10.1	Collocated Interference Request frame	8.5.14.13	WNM10:M	Yes □ No □ N/A □
WNM10.2	Collocated Interference Report frame	8.5.14.14	WNM10:M	Yes □ No □ N/A □
*WNM11	BSS Max idle period	10.23.12	CF19:M	Yes □ No □ N/A □
WNM11.1	BSS Max Idle Period element	8.4.2.81	WNM11:M	Yes □ No □ N/A □
*WNM12	TFS	10.23.11	CF19:O	Yes □ No □ N/A □
WNM12.1	TFS Request frame	8.4.2.82, 8.5.14.15	WNM12:M	Yes □ No □ N/A □
WNM12.2	TFS Response frame	8.4.2.83, 8.5.14.16	WNM12:M	Yes □ No □ N/A □
WNM12.3	TFS Notify frame	8.5.14.17	(CF1 & WNM12):M, (CF2 & WNM12):O	Yes □ No □ N/A □

B.4.21 WNM extensions *(continued)*

Item	Protocol capability	References	Status	Support
*WNM13	WNM-Sleep Mode	10.2.1.18	WNM12:O	Yes □ No □ N/A □
WNM13.1	WNM-Sleep Mode Request frame	8.4.2.84, 8.5.14.18	WNM13:M	Yes □ No □ N/A □
WNM13.2	WNM-Sleep Mode Response frame	8.4.2.84, 8.5.14.19	WNM13:M	Yes □ No □ N/A □
*WNM14	TIM Broadcast	10.2.1.17	CF19:O	Yes □ No □ N/A □
WNM14.1	TIM Broadcast Request frame	8.4.2.85, 8.5.14.20,	WNM14:M	Yes □ No □ N/A □
WNM14.2	TIM Broadcast Response frame	8.4.2.86, 8.5.14.21	WNM14:M	Yes □ No □ N/A □
WNM14.3	TIM Broadcast frame	8.5.15.2	WNM14:M	Yes □ No □ N/A □
*WNM15	QoS Traffic Capability	10.23.9	(CF19 & CF2):O	Yes □ No □ N/A □
WNM15.1	QoS Traffic Capability element	8.4.2.80	WNM15:M	Yes □ No □ N/A □
WNM15.2	QoS Traffic Capability update frame	8.5.14.22	WNM15:M	Yes □ No □ N/A □
WNM16	AC Station Count	10.23.10	(CF19 & CF2):O	Yes □ No □ N/A □
WNM17	Timing Measurement	10.23.5	CF19:O	Yes □ No □ N/A □
WNM17.1	Timing Measurement Request	8.5.14.27	WNM17:M	Yes □ No □ N/A □
WNM17.2	Timing Measurement	8.5.15.3	WNM17:M	Yes □ No □ N/A □
*WNM18	Channel Usage	10.23.14	CF19:O	Yes □ No □ N/A □
WNM18.1	Channel Usage Request frame	8.4.2.88, 8.5.14.23	WNM18:M	Yes □ No □ N/A □
WNM18.2	Channel Usage Response frame	8.4.2.88, 8.5.14.24	WNM18:M	Yes □ No □ N/A □
*WNM19	DMS	10.23.15	(CF19 & CF16):O	Yes □ No □ N/A □
WNM19.1	DMS Request frame	8.4.2.90, 8.5.14.25	WNM19:M	Yes □ No □ N/A □
WNM19.2	DMS Response frame	8.4.2.91, 8.5.14.26	WNM19:M	Yes □ No □ N/A □
WNM20	UTC TSF Offset	10.21.3, 8.4.2.63, 8.4.2.89	CF19:O	Yes □ No □ N/A □
WNM21	U-APSD Coexistence	8.4.2.93, 10.2.1.5.2	CF19:O	Yes □ No □ N/A □
WNM22	WNM-Notification	10.23.16	(CF19 & CF16):O	Yes □ No □ N/A □
WNM22.1	WNM-Notification Request frame	8.5.14.28	WNM21:M	Yes □ No □ N/A □
WNM22.2	WNM-Notification Response frame	8.5.14.29	WNM21:M	Yes □ No □ N/A □

B.4.22 Interworking (IW) with external networks extensions

Item	Protocol capability	References	Status	Support
	Are the following Interworking with External Networks capabilities supported?			
IW1	Interworking capabilities and Information	8.4.2.94, 10.24.2	CF20:M	Yes □ No □ N/A □
IW1.1	Interworking element	8.4.2.94	IW1:M	Yes □ No □ N/A □
IW1.2	Access network type	8.4.2.94	IW1:M	Yes □ No □ N/A □
IW1.3	Venue type	8.4.2.94	IW1:M	Yes □ No □ N/A □
IW1.4	HESSID	8.4.2.94	IW1:M	Yes □ No □ N/A □
IW2	Generic Advertisement Service	10.24.3	CF20:M	Yes □ No □ N/A □
IW2.1	Advertisement Protocol element	8.4.2.95	IW2:M	Yes □ No □ N/A □
*IW2.2	GAS Protocol	10.24.3.1	IW2:M	Yes □ No □ N/A □
*IW2.2.1	GAS frames	8.5.8	IW2:M	Yes □ No □ N/A □
IW2.2.2	Access Network Query Protocol	8.4.4	IW2.2:M	Yes □ No □ N/A □
IW2.2.3	Query List	8.4.4.2	IW2.2.1:M	Yes □ No □ N/A □
IW2.2.4	Capability List	8.4.4.3	IW2.2.1:M	Yes □ No □ N/A □
IW2.2.5	Venue Name	8.4.4.4	IW2.2.1:O	Yes □ No □ N/A □
IW2.2.6	Emergency Call Number	8.4.4.5	IW2.2.1:O	Yes □ No □ N/A □
IW2.2.7	Network Authentication Type	8.4.4.6	IW2.2.1:O	Yes □ No □ N/A □
IW2.2.8	Roaming Consortium	8.4.4.7	IW2.2.1:O	Yes □ No □ N/A □
IW2.2.9	IP Address Type Availability	8.4.4.9	IW2.2.1:O	Yes □ No □ N/A □
IW2.2.10	NAI Realm	8.4.4.10	IW2.2.1:O	Yes □ No □ N/A □
IW2.2.11	3GPP Cellular Network	8.4.4.11	IW2.2.1:O	Yes □ No □ N/A □
IW2.2.12	AP Geospatial Location	8.4.4.12	IW2.2.1:O	Yes □ No □ N/A □
IW2.2.13	AP Civic Location	8.4.4.13	IW2.2.1:O	Yes □ No □ N/A □
IW2.2.14	AP Location Public Identifier URI	8.4.4.14	IW2.2.1:O	Yes □ No □ N/A □
IW2.2.15	Domain Name	8.4.4.15	IW2.2.1:O	Yes □ No □ N/A □
IW2.2.16	Emergency Alert URI	8.4.4.16	IW2.2.1:O	Yes □ No □ N/A □
IW2.2.17	Emergency NAI	8.4.4.17	IW2.2.1:O	Yes □ No □ N/A □
IW2.2.18	Vendor Specific	8.4.4.8	IW2.2.1:O	Yes □ No □ N/A □
IW2.2.19	MIH IS	8.4.2.95, 10.24.4	IW2:O	Yes □ No □ N/A □
IW2.2.20	MIH Event and Command Services Discovery	8.4.2.95, 10.24.4	IW2.2:O	Yes □ No □ N/A □
IW2.2.21	Emergency Alert System (EAS)	8.4.2.95, 8.4.2.99	IW2.2:O	Yes □ No □ N/A □
IW2.2.22	Advertisement Protocol ID, Vendor Specific	8.4.2.95	IW2.2:O	Yes □ No □ N/A □
IW2.2.23	TDLS Capability	8.4.4.18	IW2.21:O	Yes □ No □ N/A □
IW2.2.24	Neighbor Report	8.4.4.19	IW2.21:O	Yes □ No □ N/A □
IW2.3	GAS Initial Request frame	8.5.8.12	IW2:M	Yes □ No □ N/A □

B.4.22 Interworking (IW) with external networks extensions *(continued)*

Item	Protocol capability	References	Status	Support
IW2.4	GAS Initial Response frame	8.5.8.13	IW2:M	Yes □ No □ N/A □
IW2.5	GAS Comeback Request frame	8.5.8.14	IW2:M	Yes □ No □ N/A □
IW2.6	GAS Comeback Response frame	8.5.8.15	IW2:M	Yes □ No □ N/A □
IW3	QoS Mapping from External Networks	10.24.9, 9.19.4.2, 9.19.4.3	CF20:O	Yes □ No □ N/A □
IW3.1	QoS Map Set element	8.4.2.97	IW3:M	Yes □ No □ N/A □
IW3.2	Transport of QoS Map Set	10.24.9	IW3:M	Yes □ No □ N/A □
IW3.3	QoS Map Configure	8.5.3.6	IW3:M	Yes □ No □ N/A □
IW4	MIH Support	6.4, 10.24.4	CF20:O	Yes □ No □ N/A □
IW4.1	MAC State Generic Convergence Function Support	6.4	IW4:M	Yes □ No □ N/A □
IW4.2	Informational events	6.4.5	IW4:M	Yes □ No □ N/A □
IW4.3	ESS status reporting	6.4.7	IW4:M	Yes □ No □ N/A □
IW4.4	Network configuration	6.4.8	IW4:M	Yes □ No □ N/A □
IW4.5	Network events	6.4.9	IW4:M	Yes □ No □ N/A □
IW4.6	Network command interface	6.4.10	IW4:M	Yes □ No □ N/A □
IW4.7	Mobility management	6.4.11	IW4:M	Yes □ No □ N/A □
IW4.8	Network configuration	6.4.8	IW4:M	Yes □ No □ N/A □
IW5	Extended channel switch enabled	8.4.2.60, 10.1.4	(CF15 AND DSE9):M	Yes □ No □ N/A □
IW6	Expedited Bandwidth Request	8.4.2.96	CF20:O	Yes □ No □ N/A □
IW7	SSPN Interface	10.24.5	CF20:O	Yes □ No □ N/A □

B.4.23 Mesh protocol capabilities

B.4.23.1 General mesh support

Item	Protocol capability	Reference	Status	Support
*MP1	Support of mesh capability	4.3.15, 13.1	CF21:M	Yes □ No □ N/A □
MP1.1	Mesh BSS scanning	13.2.2, 13.2.6	MP1:M	Yes □ No □ N/A □
MP1.2	Candidate peer mesh STA determination	13.2.7	MP1:M	Yes □ No □ N/A □
MP1.3	Active mesh profile determination	13.2.3, 13.2.4	MP1:M	Yes □ No □ N/A □
MP1.4	Establishing a mesh BSS	13.2.8	MP1:M	Yes □ No □ N/A □
MP1.5	Becoming a member of a mesh BSS	13.2.8	MP1:M	Yes □ No □ N/A □
MP1.6	Announcement of mesh profile and supplemental information for the mesh discovery	13.2.3, 13.2.5	MP1:M	Yes □ No □ N/A □

B.4.23.1 General mesh support *(continued)*

Item	Protocol capability	Reference	Status	Support
*MP2	Mesh peering management (MPM) framework	13.3	CF21:M	Yes □ No □ N/A □
*MP2.1	Mesh peering management (MPM) protocol	13.3	MP2:M	Yes □ No □ N/A □
MP2.1.1	Processing of Mesh Peering Open frame	13.3.6	MP2.1:M	Yes □ No □ N/A □
MP2.1.2	Processing of Mesh Peering Confirm frame	13.3.7	MP2.1:M	Yes □ No □ N/A □
MP2.1.3	Processing of Mesh Peering Close frame	13.3.8	MP2.1:M	Yes □ No □ N/A □
MP2.1.4	MPM finite state machine	13.4	MP2.1:M	Yes □ No □ N/A □
*MP2.2	Authenticated mesh peering exchange (AMPE)	13.5	MP2:O	Yes □ No □ N/A □
MP2.2.1	Mesh authentication using SAE	13.3.3, 11.3	MP2.2:M	Yes □ No □ N/A □
MP2.2.2	Mesh authentication using IEEE 802.1X	13.3.3, 4.10	MP2.2:O	Yes □ No □ N/A □
MP2.2.3	Protected Mesh Peering Management frame processing	13.5.3, 13.5.5	MP2.2:M	Yes □ No □ N/A □
MP2.2.4	AMPE finite state machine	13.5.6	MP2.2:M	Yes □ No □ N/A □
MP2.2.5	MGTK distribution	13.5.4	MP2.2:M	Yes □ No □ N/A □
MP2.2.6	MGTK update	13.6	MP2.2:O	Yes □ No □ N/A □
MP3	Mesh STA beaconing	13.13.3	CF21:M	Yes □ No □ N/A □
*MP4	Mesh STA synchronization	13.13.2	CF21:M	Yes □ No □ N/A □
*MP4.1	Neighbor offset synchronization method	13.13.2	MP4:M	Yes □ No □ N/A □
MP4.1.1	Calculation of TSF offset	13.13.2.2.2	MP4.1:M	Yes □ No □ N/A □
MP4.1.2	Clock drift adjustment	13.13.2.2.3	MP4.1:M	Yes □ No □ N/A □
*MP4.2	Mesh beacon collision avoidance (MBCA)	13.13.4	MP4:O	Yes □ No □ N/A □
MP4.2.1	Beacon timing advertisement	13.13.4.2	MP4.2:M	Yes □ No □ N/A □
MP4.2.2	TBTT selection	13.13.4.3	MP4.2:M	Yes □ No □ N/A □
MP4.2.3	TBTT adjustment	13.13.4.4	MP4.2:M	Yes □ No □ N/A □
MP4.2.4	Frame transmission across reported TBTT	13.13.4.5	MP4.2:O	Yes □ No □ N/A □
MP4.2.5	Delayed beacon transmission	13.13.4.6	MP4.2:O	Yes □ No □ N/A □
*MP5	MCCA	9.20.3	CF21:O	Yes □ No □ N/A □
MP5.1	MCCAOP Advertisement	9.20.3.7	MP5:M	Yes □ No □ N/A □
MP5.2	Neighbor MCCAOP Recognition	9.20.3.4–9.20.3.5	MP5:M	Yes □ No □ N/A □
MP5.3	MCCAOP Setup	9.20.3.6	MP5:M	Yes □ No □ N/A □
MP5.4	Access during MCCAOPs	9.20.3.9	MP5:M	Yes □ No □ N/A □
MP5.5	MCCAOP teardown	9.20.3.8	MP5:M	Yes □ No □ N/A □
*MP6	Intra mesh congestion control	13.12	CF21:O	Yes □ No □ N/A □

B.4.23.1 General mesh support *(continued)*

Item	Protocol capability	Reference	Status	Support
MP6.1	Local congestion monitoring and detection	13.12	MP6:M	Yes □ No □ N/A □
MP6.2	Congestion control signaling	13.12	MP6:M	Yes □ No □ N/A □
MP6.3	Local rate control	13.12	MP6:M	Yes □ No □ N/A □
*MP7	MBSS channel switch procedure	10.9.8, 10.10.3	CF21:M	Yes □ No □ N/A □
MP7.1	Transmission of channel switch advertisement	10.9.8, 10.10.3	MP7:M	Yes □ No □ N/A □
MP7.2	Propagation of channel switch advertisement	10.9.8, 10.10.3	MP7:M	Yes □ No □ N/A □
*MP8	Mesh power save operation (operation in light or deep sleep mode)	13.14	CF21:O	Yes □ No □ N/A □
MP8.1	Link-specific mesh power mode setting	13.14.2.2, 13.14.8	MP8:M	Yes □ No □ N/A □
MP8.2	Nonpeer mesh power mode setting	13.14.2.3	MP8:M	Yes □ No □ N/A □
MP8.3	Light sleep mode operation	13.14.8.4	MP8:M	Yes □ No □ N/A □
MP8.4	Deep sleep mode operation	13.14.8.5	MP8:M	Yes □ No □ N/A □
MP8.5	STA power state transitions	13.14.3	MP8:M	Yes □ No □ N/A □
MP8.6	Mesh awake window operation	13.14.6	MP8:M	Yes □ No □ N/A □
*MP9	Mesh power save support	13.14	CF21:M	Yes □ No □ N/A □
MP9.1	TIM transmission	13.14.4	MP9:M	Yes □ No □ N/A □
MP9.2	Link-specific mesh power modes determination	13.14.2	MP9:M	Yes □ No □ N/A □
MP9.3	Group addressed frame transmission	13.14.7	MP9:M	Yes □ No □ N/A □
MP9.4	Frame transmission to a mesh STA in light sleep mode	13.14.7, 13.14.9	MP9:M	Yes □ No □ N/A □
MP9.5	Frame transmission to a mesh STA in deep sleep mode	13.14.7, 13.14.9	MP9:M	Yes □ No □ N/A □
MP10	Airtime link metric computation	13.9	CF21:M	Yes □ No □ N/A □
*MP11	Link metric reporting	13.8.3	CF21:M	Yes □ No □ N/A □
MP11.1	Autonomous link metric reporting	13.8.3	MP11:O	Yes □ No □ N/A □
MP11.2	Link metric reporting upon request	13.8.3	MP11:M	Yes □ No □ N/A □
*MP12	Proxy operation	13.11.4	CF21:O	Yes □ No □ N/A □
MP12.1	Data forwarding at proxy mesh gate	13.11.3	MP12:M	Yes □ No □ N/A □
MP12.2	Maintenance of proxy information	13.11.4.2	CF21:M	Yes □ No □ N/A □

B.4.23.1 General mesh support *(continued)*

Item	Protocol capability	Reference	Status	Support
MP12.3	Proxy update using Proxy Update and Proxy Update Confirmation frames	13.11.4	CF21:M	Yes □ No □ N/A □
MP12.4	Proxy update using HWMP Mesh Path Selection frames	13.10.9, 13.10.10, 13.10.11	HWM1:M	Yes □ No □ N/A □
*MP13	Gate announcement	13.11.2	CF21:O	Yes □ No □ N/A □
MP13.1	GANN transmission	13.11.2	MP13:O	Yes □ No □ N/A □
MP13.2	GANN reception and propagation	13.11.2	CF21:M	Yes □ No □ N/A □
*MP14	Mesh Control field handling	8.2.4.7.3	CF21:M	Yes □ No □ N/A □
MP14.1	Address Extension recognition	8.2.4.7.3, 9.32.3	MP14:M	Yes □ No □ N/A □
MP14.2	Mesh TTL handling	8.2.4.7.3, 9.32.4, 9.32.5, 9.32.6	MP14:M	Yes □ No □ N/A □
MP14.3	Mesh Sequence Number handling	8.2.4.7.3, 9.32.4, 9.32.5, 9.32.6, 9.32.7	MP14:M	Yes □ No □ N/A □
*MP15	MSDU/MMPDU forwarding	9.32	CF21:O	Yes □ No □ N/A □
MP15.1	Individually addressed MSDU forwarding	9.32.4	MP15:M	Yes □ No □ N/A □
MP15.2	Group addressed MSDU forwarding	9.32.5	MP15:M	Yes □ No □ N/A □
MP15.3	MMPDU forwarding	9.32.6	MP15:M	Yes □ No □ N/A □
MP15.4	Detection of duplicate MSDUs/ MMPDUs	9.32.7	CF21:M	Yes □ No □ N/A □
MP15.5	Treatment of unknown destination	9.32.9	CF21:M	Yes □ No □ N/A □

B.4.23.2 HWMP path selection protocol capabilities

Item	Protocol capability	Reference	Status	Support
*HWM1	Hybrid wireless mesh protocol (HWMP)	13.10	CF21:M	Yes □ No □ N/A □
*HWM1.1	On-demand path selection	13.10.3	HWM1:M	Yes □ No □ N/A □
HWM1.1.1	PREQ processing for on-demand path selection	13.10.9	HWM1.1:M	Yes □ No □ N/A □
HWM1.1.2	PREP processing for on-demand path selection	13.10.10	HWM1.1:M	Yes □ No □ N/A □
HWM1.1.3	PERR processing for on-demand path selection	13.10.11	HWM1.1:M	Yes □ No □ N/A □
*HWM1.2	Proactive tree building	13.10.4	HWM1:M	Yes □ No □ N/A □
HWM1.2.1	PREQ processing for proactive tree building	13.10.9	HWM1.2:M	Yes □ No □ N/A □
HWM1.2.2	PREP processing for proactive tree building	13.10.10	HWM1.2:M	Yes □ No □ N/A □

B.4.23.2 HWMP path selection protocol capabilities *(continued)*

Item	Protocol capability	Reference	Status	Support
HWM1.2.3	PERR processing for proactive tree building	13.10.11	HWM1.2:M	Yes □ No □ N/A □
HWM1.2.4	RANN processing	13.10.12	HWM1.2:M	Yes □ No □ N/A □
HWM2	Maintenance of forwarding information	9.32.2, 13.10.8.4	MP15:M	Yes □ No □ N/A □

Annex C

(normative)

ASN.1 encoding of the MAC and PHY MIB

C.1 General

The MIB for the current revision of IEEE STD 802.11 is available online at the following URL: http://www.ieee802.org/11/802.11mib.txt.

C.2 Guidelines for 802.11 MIB Authors/Editors

The MIB may be compiled using the "smitools" package from the Institute of Operating Systems and Computer Networks at the Technical University of Braunschweig, Germany. These tools may be accessed online using the following URL: http://www.ibr.cs.tu-bs.de/bin/smitools.cgi.

Using this tool, the MIB should compile without generating warnings of severity 3 or lower.

Specific points that authors and editors should pay attention to:

- The MIB should compile as specified above prior to submission to IEEE-SA RevCom.
- Use only the 7-bit character-set. In particular, use only straight single and double quotes.
- Do not use symbols such as µ in units, instead spell out the units fully, i.e., "microseconds."
- Follow the guidelines for writing MIB modules in http://www.rfc-editor.org/rfc/rfc4181.txt (including any updates thereto).
- Pay attention to IETF recommendations on object types.
- When adding a lot of new objects, consider adding a new module.
- Consider seeking advice from an IETF MIB Doctor when creating significant new material. The IEEE has an arrangement with the IETF whereby the IEEE may ask for a MIB Advisor from the IETF MIB Doctors.
- When an object is deprecated, add a line to the Description indicating why (IETF convention).

C.3 MIB Detail

```
-- **********************************************************************
-- * IEEE 802.11 MIB
-- **********************************************************************

IEEE802dot11-MIB DEFINITIONS ::= BEGIN

IMPORTS
   MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
   Integer32, Counter32, Counter64, Unsigned32, TimeTicks, Gauge32
   FROM SNMPv2-SMI

   DisplayString , MacAddress, RowStatus, TruthValue,
   TEXTUAL-CONVENTION FROM SNMPv2-TC

   MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP FROM SNMPv2-CONF
   ifIndex, InterfaceIndex FROM IF-MIB;

-- **********************************************************************
-- *  MODULE IDENTITY
-- **********************************************************************

ieee802dot11 MODULE-IDENTITY
   LAST-UPDATED "201203300000Z"
   ORGANIZATION "IEEE 802.11"
   CONTACT-INFO
      "Chair: Bruce Kraemer
      Postal: 517 La Costa Court
      Melbourne, FL 32940, USA
      Tel: +1 321 427 4098
      Fax: +1 321 751 3988
      E-mail: bkraemer@ieee.org

      Editor: Adrian P Stephens
      E-mail: adrian.p.stephens@ieee.org"
   DESCRIPTION
      "The MIB module for IEEE 802.11 entities.
      iso(1).member-body(2).us(840).ieee802dot11(10036)"

   REVISION    "201203300000Z"
   DESCRIPTION "Draft Revision MB of the 802.11 Standard.

      Note that not all objects within this MIB are referenced by a group,  and
      not all groups are referenced by a MODULE-COMPLIANCE statement.   Some
      existing groups and the dot11Compliance MODULE-COMPLIANCE have been modi-
      fied since the previous revision of this standard.

      Implementations should not claim compliance to dot11Compliance."
   ::= { us 10036 }

-- **********************************************************************
-- *  Tree Definition
-- **********************************************************************

   member-body     OBJECT IDENTIFIER ::= { iso 2 }
   us              OBJECT IDENTIFIER ::= { member-body 840 }

-- **********************************************************************
-- *  Major sections
-- **********************************************************************

--  Station ManagemenT (SMT) Attributes
    --  DEFINED AS "The SMT object class provides the necessary support
    --  at the station to manage the processes in the station such that
    --  the station may work cooperatively as a part of an IEEE 802.11
    --  network."

   dot11smt OBJECT IDENTIFIER ::= { ieee802dot11 1 }
   --  dot11smt GROUPS
   --  dot11StationConfigTable                     ::= { dot11smt 1 }
   --  dot11AuthenticationAlgorithmsTable          ::= { dot11smt 2 }
   --  dot11WEPDefaultKeysTable                    ::= { dot11smt 3 }
   --  dot11WEPKeyMappingsTable                    ::= { dot11smt 4 }
   --  dot11PrivacyTable                           ::= { dot11smt 5 }
   --  dot11SMTnotification                        ::= { dot11smt 6 }
   --  dot11MultiDomainCapabilityTable             ::= { dot11smt 7 }
   --  dot11SpectrumManagementTable                ::= { dot11smt 8 }
   --  dot11RSNAConfigTable                        ::= { dot11smt 9 }
   --  dot11RSNAConfigPairwiseCiphersTable         ::= { dot11smt 10 }
   --  dot11RSNAConfigAuthenticationSuitesTable      ::= { dot11smt 11 }
   --  dot11RSNAStatsTable                           ::= { dot11smt 12 }
   --  dot11OperatingClassesTable                    ::= { dot11smt 13 }
   --  dot11RadioMeasurement                         ::= { dot11smt 14 }
   --  dot11FastBSSTransitionConfigTable             ::= { dot11smt 15 }
   --  dot11LCIDSETable                              ::= { dot11smt 16 }
   --  dot11HTStationConfigTable                     ::= { dot11smt 17 }
   --  dot11WirelessMgmtOptionsTable                 ::= { dot11smt 18}
   --  dot11LocationServicesNextIndex                ::= { dot11smt 19}
   --  dot11LocationServicesTable                    ::= { dot11smt 20}
   --  dot11WirelessMGTEventTable                    ::= { dot11smt 21}
   --  dot11WirelessNetworkManagement                ::= { dot11smt 22}
   --  dot11MeshSTAConfigTable                       ::= { dot11smt 23 }
   --  dot11MeshHWMPConfigTable                      ::= { dot11smt 24 }
   --  dot11RSNAConfigPasswordValueTable             ::= { dot11smt 25 }
   --  dot11RSNAConfigDLCGroupTable                  ::= { dot11smt 26 }

--  MAC Attributes
    --  DEFINED AS "The MAC object class provides the necessary support
    --  for the access control, generation, and verification of frame
    --  check sequences (FCSs), and proper delivery of valid data to
    --  upper layers."

   dot11mac OBJECT IDENTIFIER ::= { ieee802dot11 2 }

--  MAC GROUPS
   --  dot11OperationTable                       ::= { dot11mac 1 }
   --  dot11CountersTable                        ::= { dot11mac 2 }
   --  dot11GroupAddressesTable                  ::= { dot11mac 3 }
   --  dot11EDCATable                            ::= { dot11mac 4 }
   --  dot11QAPEDCATable                         ::= { dot11mac 5 }
   --  dot11QosCountersTable                     ::= { dot11mac 6 }
   --  dot11ResourceInfoTable                    ::= { dot11mac 7 }

--  Resource Type ID
   dot11res          OBJECT IDENTIFIER ::= { ieee802dot11 3 }
   dot11resAttribute OBJECT IDENTIFIER ::= { dot11res 1 }

--  PHY Attributes
    --  DEFINED AS "The PHY object class provides the necessary support
    --  for required PHY operational information that may vary from PHY
    --  to PHY and from STA to STA to be communicated to upper layers."

   dot11phy OBJECT IDENTIFIER ::= { ieee802dot11 4 }

--  PHY GROUPS
   --  dot11PhyOperationTable                              ::= { dot11phy 1 }
   --  dot11PhyAntennaTable                        ::= { dot11phy 2 }
   --  dot11PhyTxPowerTable                        ::= { dot11phy 3 }
   --  dot11PhyFHSSTable                           ::= { dot11phy 4 }
   --  dot11PhyDSSSTable                           ::= { dot11phy 5 }
   --  dot11PhyIRTable                             ::= { dot11phy 6 }
   --  dot11RegDomainsSupportedTable               ::= { dot11phy 7 }
   --  dot11AntennasListTable                      ::= { dot11phy 8 }
   --  dot11SupportedDataRatesTxTable              ::= { dot11phy 9 }
   --  dot11SupportedDataRatesRxTable              ::= { dot11phy 10 }
   --  dot11PhyOFDMTable                           ::= { dot11phy 11 }
   --  dot11PhyHRDSSSTable                         ::= { dot11phy 12 }
   --  dot11HoppingPatternTable                    ::= { dot11phy 13 }
   --  dot11PhyERPTable                            ::= { dot11phy 14 }
   --  dot11PhyHTTable                             ::= { dot11phy 15 }
   --  dot11SupportedMCSTxTable                    ::= { dot11phy 16 }
   --  dot11SupportedMCSRxTable                    ::= { dot11phy 17 }
   --  dot11TransmitBeamformingConfigTable         ::= { dot11phy 18 }
-- **********************************************************************
-- *  Textual conventions for 802 definitions
-- **********************************************************************

WEPKeytype ::= TEXTUAL-CONVENTION
   STATUS      current
   DESCRIPTION "Represents the type of WEP key."
   SYNTAX      OCTET STRING (SIZE (5))

TSFType ::= TEXTUAL-CONVENTION
   STATUS      current
   DESCRIPTION "Represents the type of TSF."
   SYNTAX      OCTET STRING (SIZE (8))

-- **********************************************************************
-- *  MIB attribute OBJECT-TYPE definitions follow
-- **********************************************************************

-- **********************************************************************
-- *  dot11StationConfig  TABLE
-- **********************************************************************

dot11StationConfigTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11StationConfigEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "Station Configuration attributes. In tabular form to allow for multiple
      instances on an agent."
   ::= { dot11smt 1 }

dot11StationConfigEntry OBJECT-TYPE
   SYNTAX Dot11StationConfigEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "An entry in the dot11StationConfigTable. It is possible for there to be
      multiple IEEE 802.11 interfaces on one agent, each with its unique MAC
      address. The relationship between an IEEE 802.11 interface and an inter-
      face in the context of the Internet-standard MIB is one-to-one. As such,
      the value of an ifIndex object instance can be directly used to identify
      corresponding instances of the objects defined herein.

      ifIndex - Each IEEE 802.11 interface is represented by an ifEntry. Inter-
      face tables in this MIB module are indexed by ifIndex."
   INDEX { ifIndex }
   ::= { dot11StationConfigTable 1 }

Dot11StationConfigEntry ::= SEQUENCE
   {
      dot11StationID                                      MacAddress,
      dot11MediumOccupancyLimit                           Unsigned32,
      dot11CFPollable                                     TruthValue,
      dot11CFPPeriod                                      Unsigned32,
      dot11CFPMaxDuration                                 Unsigned32,
      dot11AuthenticationResponseTimeOut                  Unsigned32,
      dot11PrivacyOptionImplemented                       TruthValue,
      dot11PowerManagementMode                            INTEGER,
      dot11DesiredSSID                                    OCTET STRING,
      dot11DesiredBSSType                                 INTEGER,
      dot11OperationalRateSet                             OCTET STRING,
      dot11BeaconPeriod                                   Unsigned32,
      dot11DTIMPeriod                                     Unsigned32,
      dot11AssociationResponseTimeOut                        Unsigned32,
      dot11DisassociateReason                                Unsigned32,
      dot11DisassociateStation                               MacAddress,
      dot11DeauthenticateReason                              Unsigned32,
      dot11DeauthenticateStation                             MacAddress,
      dot11AuthenticateFailStatus                            Unsigned32,
      dot11AuthenticateFailStation                           MacAddress,
      dot11MultiDomainCapabilityImplemented                  TruthValue,
      dot11MultiDomainCapabilityActivated                    TruthValue,
      dot11CountryString                                     OCTET STRING,
      dot11SpectrumManagementImplemented                     TruthValue,
      dot11SpectrumManagementRequired                        TruthValue,
      dot11RSNAOptionImplemented                             TruthValue,
      dot11RSNAPreauthenticationImplemented                  TruthValue,
      dot11OperatingClassesImplemented                       TruthValue,
      dot11OperatingClassesRequired                          TruthValue,
      dot11QosOptionImplemented                              TruthValue,
      dot11ImmediateBlockAckOptionImplemented                TruthValue,
      dot11DelayedBlockAckOptionImplemented                  TruthValue,
      dot11DirectOptionImplemented                           TruthValue,
      dot11APSDOptionImplemented                             TruthValue,
      dot11QAckOptionImplemented                             TruthValue,
      dot11QBSSLoadImplemented                               TruthValue,
      dot11QueueRequestOptionImplemented                     TruthValue,
      dot11TXOPRequestOptionImplemented                      TruthValue,
      dot11MoreDataAckOptionImplemented                      TruthValue,
      dot11AssociateInNQBSS                                  TruthValue,
      dot11DLSAllowedInQBSS                                  TruthValue,
      dot11DLSAllowed                                        TruthValue,
      dot11AssociateStation                                  MacAddress,
      dot11AssociateID                                       Unsigned32,
      dot11AssociateFailStation                              MacAddress,
      dot11AssociateFailStatus                               Unsigned32,
      dot11ReassociateStation                                MacAddress,
      dot11ReassociateID                                     Unsigned32,
      dot11ReassociateFailStation                            MacAddress,
      dot11ReassociateFailStatus                             Unsigned32,
      dot11RadioMeasurementImplemented                       TruthValue,
      dot11RadioMeasurementActivated                         TruthValue,
      dot11RMMeasurementProbeDelay                           Unsigned32,
      dot11RMMeasurementPilotPeriod                          Unsigned32,
      dot11RMLinkMeasurementActivated                        TruthValue,
      dot11RMNeighborReportActivated                         TruthValue,
      dot11RMParallelMeasurementsActivated                   TruthValue,
      dot11RMRepeatedMeasurementsActivated                   TruthValue,
      dot11RMBeaconPassiveMeasurementActivated               TruthValue,
      dot11RMBeaconActiveMeasurementActivated                TruthValue,
      dot11RMBeaconTableMeasurementActivated                 TruthValue,
      dot11RMBeaconMeasurementReportingConditionsActivated   TruthValue,
      dot11RMFrameMeasurementActivated                       TruthValue,
      dot11RMChannelLoadMeasurementActivated                 TruthValue,
      dot11RMNoiseHistogramMeasurementActivated              TruthValue,
      dot11RMStatisticsMeasurementActivated                  TruthValue,
      dot11RMLCIMeasurementActivated                         TruthValue,
      dot11RMLCIAzimuthActivated                             TruthValue,
      dot11RMTransmitStreamCategoryMeasurementActivated      TruthValue,
      dot11RMTriggeredTransmitStreamCategoryMeasurementActivated
      TruthValue,
      dot11RMAPChannelReportActivated                        TruthValue,
      dot11RMMIBActivated                                    TruthValue,
      dot11RMMaxMeasurementDuration                          Unsigned32,
      dot11RMNonOperatingChannelMaxMeasurementDuration       Unsigned32,
      dot11RMMeasurementPilotTransmissionInformationActivated
      TruthValue,
      dot11RMMeasurementPilotActivated                       Unsigned32,
      dot11RMNeighborReportTSFOffsetActivated                TruthValue,
      dot11RMRCPIMeasurementActivated                        TruthValue,
      dot11RMRSNIMeasurementActivated                        TruthValue,
      dot11RMBSSAverageAccessDelayActivated                  TruthValue,
      dot11RMBSSAvailableAdmissionCapacityActivated          TruthValue,
      dot11RMAntennaInformationActivated                     TruthValue,
      dot11FastBSSTransitionImplemented                      TruthValue,
      dot11LCIDSEImplemented                                 TruthValue,
      dot11LCIDSERequired                                    TruthValue,
      dot11DSERequired                                       TruthValue,
      dot11ExtendedChannelSwitchActivated                    TruthValue,
      dot11RSNAProtectedManagementFramesActivated
      TruthValue,
      dot11RSNAUnprotectedManagementFramesAllowed            TruthValue,
      dot11AssociationSAQueryMaximumTimeout                  Unsigned32,
      dot11AssociationSAQueryRetryTimeout                    Unsigned32,
      dot11HighThroughputOptionImplemented                   TruthValue,
      dot11RSNAPBACRequired                                  TruthValue,
      dot11PSMPOptionImplemented                             TruthValue,
      dot11TunneledDirectLinkSetupImplemented                TruthValue,
      dot11TDLSPeerUAPSDBufferSTAActivated                   TruthValue,
      dot11TDLSPeerPSMActivated                              TruthValue,
      dot11TDLSPeerUAPSDIndicationWindow                     Unsigned32,
      dot11TDLSChannelSwitchingActivated                     TruthValue,
      dot11TDLSPeerSTAMissingAckRetryLimit                   Unsigned32,
      dot11TDLSResponseTimeout                               Unsigned32,
      dot11OCBActivated                                      TruthValue,
      dot11TDLSProbeDelay                                    Unsigned32,
      dot11TDLSDiscoveryRequestWindow                        Unsigned32,
      dot11TDLSACDeterminationInterval                       Unsigned32,
      dot11WirelessManagementImplemented                     TruthValue,
      dot11BssMaxIdlePeriod                                  Unsigned32,
      dot11BssMaxIdlePeriodOptions                           OCTET STRING,
      dot11TIMBroadcastInterval                              Unsigned32,
      dot11TIMBroadcastOffset                                Integer32,
      dot11TIMBroadcastHighRateTIMRate                       Unsigned32,
      dot11TIMBroadcastLowRateTIMRate                        Unsigned32,
      dot11StatsMinTriggerTimeout                            Unsigned32,
      dot11RMCivicMeasurementActivated                       TruthValue,
      dot11RMIdentifierMeasurementActivated                  TruthValue,
      dot11TimeAdvertisementDTIMInterval                     Unsigned32,
      dot11TimeAdvertisementTimeError                        OCTET STRING,
      dot11TimeAdvertisementTimeValue                        OCTET STRING,
      dot11RM3rdPartyMeasurementActivated                    TruthValue,
      dot11InterworkingServiceImplemented                    TruthValue,
      dot11InterworkingServiceActivated                      TruthValue,
      dot11QosMapImplemented                                 TruthValue,
      dot11QosMapActivated                                   TruthValue,
      dot11EBRImplemented                                    TruthValue,
      dot11EBRActivated                                      TruthValue,
      dot11ESNetwork                                         TruthValue,
      dot11SSPNInterfaceImplemented                          TruthValue,
      dot11SSPNInterfaceActivated                            TruthValue,
      dot11HESSID                                            MacAddress,
      dot11EASImplemented                                    TruthValue,
      dot11EASActivated                                      TruthValue,
      dot11MSGCFImplemented                                  TruthValue,
      dot11MSGCFActivated                                    TruthValue,
      dot11MeshActivated                                     TruthValue,
      dot11RejectUnadmittedTraffic                           TruthValue,
      dot11BSSBroadcastNullCount                             Unsigned32
   }
```

```
dot11StationID OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by the SME or external management entity.
        Changes take effect for the next MLME-START.request primitive.

        The purpose of dot11StationID is to allow a manager to identify a station
        for its own purposes. This attribute provides for that eventuality while
        keeping the true MAC address independent. Its syntax is MAC address, and
        the default value is the station's assigned, unique MAC address."
    ::= { dot11StationConfigEntry 1 }

dot11MediumOccupancyLimit OBJECT-TYPE
    SYNTAX Unsigned32 (0..1000)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute indicates the maximum amount of time, in TU, that a point
        coordinator (PC) may control the usage of the wireless medium (WM) without
        relinquishing control for long enough to allow at least one instance of
        DCF access to the medium. The maximum value of this variable is 1000."
    DEFVAL { 100 }
    ::= { dot11StationConfigEntry 2 }

dot11CFPollable OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        When this attribute is true, it indicates that the STA is able to respond
        to a CF-Poll with a data frame within a SIFS time. This attribute is false
        if the STA is not able to respond to a CF-Poll with a data frame within a
        SIFS time."
    ::= { dot11StationConfigEntry 3 }

dot11CFPPeriod OBJECT-TYPE
    SYNTAX Unsigned32 (0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        In an AP, it is written by the MAC when it receives an MLME-START.request
        primitive.
        In a non-AP STA, it is written by the MAC when it receives an updated CF
        Parameter Set in a Beacon frame.

        The attribute describes the number of DTIM intervals between the start of
        CFPs. It is modified by MLME-START.request primitive."
    ::= { dot11StationConfigEntry 4 }

dot11CFPMaxDuration OBJECT-TYPE
    SYNTAX Unsigned32 (0..65535)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        In an AP, it is written by the MAC when it receives an MLME-START.request
        primitive.
        In a non-AP STA, it is written by the MAC when it receives an updated CF
        Parameter Set in a Beacon frame.

        The attribute describes the maximum duration of the CFP in TU that may be
        generated by the PCF. It is modified by MLME-START.request primitive."
    ::= { dot11StationConfigEntry 5 }

dot11AuthenticationResponseTimeOut OBJECT-TYPE
    SYNTAX Unsigned32 (1..4294967295)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect for the next Authentication frame.

        This attribute specifies the number of time units (TUs) that a responding
        STA should wait for the next frame in the authentication sequence."
    ::= { dot11StationConfigEntry 6 }

dot11PrivacyOptionImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute, when true, indicates that the IEEE             802.11 WEP
        option is implemented."
    DEFVAL { false }
    ::= { dot11StationConfigEntry 7 }

dot11PowerManagementMode OBJECT-TYPE
    SYNTAX INTEGER { active(1), powersave(2) }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when an MLME-POWERMGT.request primitive is
        received.

        This attribute specifies the power management mode of the STA. When equal
        to active, it indicates that the station is not in power save (PS) mode.
        When equal to powersave, it indicates that the station is in power save
        mode. The power management mode is transmitted in all frames according to
        the rules in  8.2.4.1.8."
    ::= { dot11StationConfigEntry 8 }

dot11DesiredSSID OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(0..32))
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect for the next MLME-SCAN.request primitive.

        This attribute reflects the Service Set ID (SSID) used in the SSID param-
        eter of the most recent MLME-SCAN.request primitive. This value may be
```

```
      modified by an external management entity and used by the local SME to
      make decisions about the Scanning process."
   ::= { dot11StationConfigEntry 9 }

dot11DesiredBSSType OBJECT-TYPE
   SYNTAX INTEGER { infrastructure(1), independent(2), any(3) }
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect for the next MLME-SCAN.request primitive.

      This attribute specifies the type of BSS the station uses when scanning
      for a BSS with which to synchronize. This value is used to filter Probe
      Response frames and Beacon frames. When equal to infrastructure, the sta-
      tion synchronizes only with a BSS in which the Capability Information
      field has the ESS subfield equal to 1. When equal to independent, the sta-
      tion synchronizes only with a BSS whose Capability Information field has
      the IBSS subfield equal to 1. When equal to any, the station may synchro-
      nize to either type of BSS."
   ::= { dot11StationConfigEntry 10 }

dot11OperationalRateSet OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(1..126))
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when joining or establishing a BSS.

      This attribute specifies the set of non-HT data rates at which the station
      may transmit data. The attribute that specifies the set of HT data rates
      is dot11HTOperationalMCSSet.  Each octet contains a value representing a
      rate. Each rate is within the range from 2 to 127, corresponding to data
      rates in increments of 500 kbit/s from 1 Mb/s to 63.5 Mb/s, and is sup-
      ported (as indicated in the supported rates table) for receiving data.
      This value is reported in transmitted Beacon, Probe Request, Probe
      Response, Association Request, Association Response, Reassociation
      Request, and Reassociation Response frames, and is used to determine
      whether a BSS with which the station desires to synchronize is suitable.
      It is also used when starting a BSS, as specified in  6.3.4."
   ::= { dot11StationConfigEntry 11 }

dot11BeaconPeriod OBJECT-TYPE
   SYNTAX Unsigned32 (1..65535)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect for the next MLME-START.request primitive.

      This attribute specifies the number of TUs that a station uses for sched-
      uling Beacon transmissions. This value is transmitted in Beacon and Probe
      Response frames."
   ::= { dot11StationConfigEntry 12 }

dot11DTIMPeriod OBJECT-TYPE
   SYNTAX Unsigned32(1..255)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
```

```
       It is written by an external management entity.
       Changes take effect for the next MLME-START.request primitive.

       This attribute specifies the number of beacon intervals that elapse
       between transmission of Beacon frames containing a TIM element whose DTIM
       Count field is 0. This value is transmitted in the DTIM Period field of
       Beacon frames."
    ::= { dot11StationConfigEntry 13 }

dot11AssociationResponseTimeOut OBJECT-TYPE
    SYNTAX Unsigned32 (1..4294967295)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
       "This is a control variable.
       It is written by an external management entity.
       Changes take effect as soon as practical in the implementation.

       This attribute specifies the number of TU that a requesting STA should
       wait for a response to a transmitted association-request MMPDU."
    ::= { dot11StationConfigEntry 14 }

dot11DisassociateReason OBJECT-TYPE
    SYNTAX Unsigned32(0..65535)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
       "This is a status variable.
       It is written by the MAC when a STA disassociates.

       This attribute holds the most recently transmitted Reason Code in a Disas-
       sociation frame. If no Disassociation frame has been transmitted, the
       value of this attribute is 0."
    REFERENCE "IEEE Std 802.11-2012, 8.4.1.7"
    ::= { dot11StationConfigEntry 15 }

dot11DisassociateStation OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
       "This is a status variable.
       It is written by the MAC when a STA disassociates.

       This attribute holds the MAC address from the Address 1 field of the most
       recently transmitted Disassociation frame. If no Disassociation frame has
       been transmitted, the value of this attribute is 0."
    ::= { dot11StationConfigEntry 16 }

dot11DeauthenticateReason OBJECT-TYPE
    SYNTAX Unsigned32(0..65535)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
       "This is a status variable.
       It is written by the MAC when a STA deauthenticates.

       This attribute holds the most recently transmitted Reason Code in a Deau-
       thentication frame. If no Deauthentication frame has been transmitted, the
       value of this attribute is 0."
    REFERENCE "IEEE Std 802.11-2012, 8.4.1.7"
    ::= { dot11StationConfigEntry 17 }

dot11DeauthenticateStation OBJECT-TYPE
```

```
    SYNTAX MacAddress
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when a STA deauthenticates.

        This attribute holds the MAC address from the Address 1 field of the most
        recently transmitted Deauthentication frame. If no Deauthentication frame
        has been transmitted, the value of this attribute is 0."
    ::= { dot11StationConfigEntry 18 }

dot11AuthenticateFailStatus OBJECT-TYPE
    SYNTAX Unsigned32(0..65535)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when a STA fails to authenticate.

        This attribute holds the most recently transmitted Status Code in a failed
        Authentication frame. If no failed Authentication frame has been transmit-
        ted, the value of this attribute is 0."
    REFERENCE "IEEE Std 802.11-2012, 8.4.1.9"
    ::= { dot11StationConfigEntry 19 }

dot11AuthenticateFailStation OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when a STA fails to authenticate.

        This attribute holds the MAC address from the Address 1 field of the most
        recently transmitted failed Authentication frame. If no failed Authentica-
        tion frame has been transmitted, the value of this attribute is 0."
    ::= { dot11StationConfigEntry 20 }

dot11MultiDomainCapabilityImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute, when true, indicates that the station implementation is
        capable of supporting multiple regulatory domains. The capability is dis-
        abled, otherwise."
    DEFVAL { false }
    ::= { dot11StationConfigEntry 21 }

dot11MultiDomainCapabilityActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by the SME or external management entity.
        Changes take effect for the next MLME-START.request primitive.

        This attribute, when true, indicates that the capability of the station to
        operate in multiple regulatory domains is enabled."
    DEFVAL { false }
    ::= { dot11StationConfigEntry 22 }

dot11CountryString OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(3))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by the SME.
        Changes take effect for the next MLME-START.request primitive.

        This attribute identifies the country or noncountry entity in which the
        station is operating. If it is a country, the first two octets of this
        string is the two character country code as described in document ISO/IEC
        3166-1. The third octet is one of the following:

        1. an ASCII space character, if the regulations under which the station is
        operating encompass all environments for the current frequency band in the
        country,

        2. an ASCII 'O' character, if the regulations under which the station is
        operating are for an Outdoor environment only, or

        3. an ASCII 'I' character, if the regulations under which the station is
        operating are for an Indoor environment only.

        4. an ASCII 'X' character, if the station is operating under a noncountry
        entity. The first two octets of the noncountry entity is two ASCII 'XX'
        characters.

        5. the binary representation of the Operating Class table number currently
        in use, from the set of tables defined in Annex E, e.g. Table E-1 is rep-
        resented as x'01'."
    ::= { dot11StationConfigEntry 23 }

dot11SpectrumManagementImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute, when true, indicates that the station implementation is
        capable of supporting spectrum management. The capability is disabled oth-
        erwise."
    DEFVAL { false }
    ::= { dot11StationConfigEntry 24 }

dot11SpectrumManagementRequired OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by the SME or external management entity.
        Changes take effect for the next MLME-START.request primitive.

        A STA uses the defined TPC and DFS procedures if this attribute is true;
        otherwise it does not use the defined TPC and DFS procedures."
    DEFVAL { false }
    ::= { dot11StationConfigEntry 25 }
```

```
dot11RSNAOptionImplemented OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      This variable indicates whether the entity is RSNA-capable."
   ::= { dot11StationConfigEntry 26 }

dot11RSNAPreauthenticationImplemented OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      This variable indicates whether the entity supports RSNA preauthentica-
      tion. This cannot be true unless dot11RSNAOptionImplemented is true."
   ::= { dot11StationConfigEntry 27 }

dot11OperatingClassesImplemented OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      This attribute, when true, indicates that the station implementation is
      capable of supporting operating classes. The capability is disabled other-
      wise."
   DEFVAL { false }
   ::= { dot11StationConfigEntry 28 }

dot11OperatingClassesRequired OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by the SME or external management entity.
      Changes take effect for the next MLME-START.request primitive.

      A STA uses the defined operating classes procedures if this attribute is
      true."
   DEFVAL { false }
   ::= { dot11StationConfigEntry 29 }

dot11QosOptionImplemented OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      This attribute, when true, indicates that the station implementation is
      capable of supporting QoS. The capability is disabled, otherwise."
   DEFVAL { false }
   ::= { dot11StationConfigEntry 30}
```

```
dot11ImmediateBlockAckOptionImplemented OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      This attribute, when true, indicates that the station implementation is
      capable of supporting Immediate Block Ack. The capability is disabled,
      otherwise."
   DEFVAL { false }
   ::= { dot11StationConfigEntry 31}

dot11DelayedBlockAckOptionImplemented OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      This attribute, when true, indicates that the station implementation is
      capable of supporting Delayed Block Ack. The capability is disabled, oth-
      erwise."
   DEFVAL { false }
   ::= { dot11StationConfigEntry 32 }

dot11DirectOptionImplemented OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      This attribute, when true, indicates that the station implementation is
      capable of sending and receiving frames from a non-AP STA in the BSS. The
      capability is disabled, otherwise."
   DEFVAL { false }
   ::= { dot11StationConfigEntry 33 }

dot11APSDOptionImplemented OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      This attribute is available only at an AP. When true, this attribute indi-
      cates that the AP implementation is capable of delivering data and polls
      to stations using APSD."
   ::= { dot11StationConfigEntry 34 }

dot11QAckOptionImplemented OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      This attribute, when true, indicates that the station implementation is
```

```
        capable of interpreting the CF-Ack bit in a received frame of type data
        even if the frame is not directed to the QoS station. The capability is
        disabled, otherwise. A STA is capable of interpreting the CF-Ack bit in a
        received data frame if that station is the recipient of the data frame,
        regardless of the value of this MIB variable."
    DEFVAL { false }
    ::= { dot11StationConfigEntry 35 }

dot11QBSSLoadImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute is available only at an AP. This attribute, when true,
        indicates that the AP implementation is capable of generating and trans-
        mitting the BSS load element in the Beacon and Probe Response frames. The
        capability is disabled, otherwise."
    DEFVAL { false }
    ::= { dot11StationConfigEntry 36 }

dot11QueueRequestOptionImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute is available only at an AP. This attribute, when true,
        indicates that the AP is capable of processing the Queue Size field in QoS
        Control field of QoS Data type frames. The capability is disabled, other-
        wise."
    DEFVAL { false }
    ::= { dot11StationConfigEntry 37 }

dot11TXOPRequestOptionImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute is available only at an AP. This attribute, when true,
        indicates that the AP is capable of processing the TXOP Duration requested
        field in QoS Control field of QoS Data type frames. The capability is dis-
        abled, otherwise."
    DEFVAL { false }
    ::= { dot11StationConfigEntry 38 }

dot11MoreDataAckOptionImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute, when true, indicates that the station implementation is
        capable of interpreting the More Data bit in the ACK frames. The capabil-
        ity is disabled, otherwise."
    DEFVAL { false }
    ::= { dot11StationConfigEntry 39 }

dot11AssociateInNQBSS OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute, when true, indicates that the station may associate in a
        non-QoS BSS. When false, this attribute indicates that the station does
        not associate in a non-QoS BSS."
    DEFVAL { true }
    ::= { dot11StationConfigEntry 40 }

dot11DLSAllowedInQBSS OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute available at the AP, when true, indicates that STAs may set
        up DLS and communicate using DLS with other STAs in the BSS. When false,
        this attribute indicates that the stations do not set up DLS nor communi-
        cate with other STAs using DLS in the BSS."
    DEFVAL { true }
    ::= { dot11StationConfigEntry 41 }

dot11DLSAllowed OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute available at non-AP STAs, when true, indicates that the STA
        may set up DLS or accept DLS Requests, and communicate with other STAs in
        the BSS using DLS. When false, this attribute indicates that the STA does
        not set up DLS nor accept DLS, nor communicate with other STAs in the BSS
        using DLS."
    DEFVAL { true }
    ::= { dot11StationConfigEntry 42}

dot11AssociateStation OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when a STA associates.

        This attribute indicates the MAC address from the Address 1 field of the
        most recently transmitted association response frame. If no association
        response frame has been transmitted, the value of this attribute is 0."
    ::= { dot11StationConfigEntry 43 }
```

```
dot11AssociateID OBJECT-TYPE
   SYNTAX Unsigned32(0..2007)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the MAC when a STA associates.

      This attribute indicates the Association ID from the most recently trans-
      mitted association response frame. If no association response frame has
      been transmitted, the value of this attribute is 0."
   ::= { dot11StationConfigEntry 44 }

dot11AssociateFailStation OBJECT-TYPE
   SYNTAX MacAddress
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the MAC when a STA fails to associate.

      This attribute indicates the MAC address from the Address 1 field of the
      most recently transmitted failed association response frame. If no failed
      association response frame has been transmitted, the value of this attri-
      bute is 0."
   ::= { dot11StationConfigEntry 45 }

dot11AssociateFailStatus OBJECT-TYPE
   SYNTAX Unsigned32(0..65535)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the MAC when a STA fails to associate.

      This attribute indicates the most recently transmitted Status Code in a
      failed association response frame. If no failed association response frame
      has been transmitted, the value of this attribute is 0."
   ::= { dot11StationConfigEntry 46 }

dot11ReassociateStation OBJECT-TYPE
   SYNTAX MacAddress
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the MAC when a STA reassociates.

      This attribute indicates the MAC address from the Address 1 field of the
      most recently transmitted reassociation response frame. If no reassocia-
      tion response frame has been transmitted, the value of this attribute is
      0."
   ::= { dot11StationConfigEntry 47 }

dot11ReassociateID OBJECT-TYPE
   SYNTAX Unsigned32(0..2007)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the MAC when a STA reassociates.

      This attribute indicates the Association ID from the most recently trans-
      mitted reassociation response frame. If no reassociation response frame
      has been transmitted, the value of this attribute is 0."
   ::= { dot11StationConfigEntry 48 }

dot11ReassociateFailStation OBJECT-TYPE
   SYNTAX MacAddress
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the MAC when a STA fails to reassociate.

      This attribute indicates the MAC address from the Address 1 field of the
      most recently transmitted failed reassociation response frame. If no
      failed reassociation response frame has been transmitted, the value of
      this attribute is 0."
   ::= { dot11StationConfigEntry 49 }

dot11ReassociateFailStatus OBJECT-TYPE
   SYNTAX Unsigned32(0..65535)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the MAC when a STA fails to reassociate.

      This attribute indicates the most recently transmitted Status Code in a
      failed reassociation response frame. If no failed reassociation response
      frame has been transmitted, the value of this attribute is 0."
   ::= { dot11StationConfigEntry 50 }

dot11RadioMeasurementImplemented OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      This attribute, when true, indicates that the station implementation is
      capable of supporting Radio Measurement. Otherwise it is not capable of
      performing Radio Measurement."
   DEFVAL { false }
   ::= { dot11StationConfigEntry 51 }

dot11RadioMeasurementActivated OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity when any of the MIB vari-
      ables listed in 8.4.2.47 is equal to true.
      Changes take effect with the next MLME-START.request primitive or MLME-
      JOIN.request primitive.

      This attribute, when true, indicates that one or more of the Radio Mea-
      surement Activated Capabilities MIB attributes, listed in 8.4.2.47, are
      equal to true. A STA may use the defined Radio Measurement procedures if
      this attribute is true."
   DEFVAL { false }
   ::= { dot11StationConfigEntry 52 }

dot11RMMeasurementProbeDelay OBJECT-TYPE
   SYNTAX Unsigned32
```

```
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        The value of ProbeDelay to be used when making a beacon type measurement
        with measurement mode active when dot11RMActiveBeaconMeasurementActivated
        is true."
    ::= { dot11StationConfigEntry 53 }

dot11RMMeasurementPilotPeriod OBJECT-TYPE
    SYNTAX Unsigned32 (1..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute specifies the number of TUs that a station uses for sched-
        uling Measurement Pilot transmissions. This value is transmitted in Mea-
        surement Pilot frames. The default period is 25% of dot11BeaconPeriod."
    ::= { dot11StationConfigEntry 54 }

dot11RMLinkMeasurementActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect for the next MLME-START.request primitive or MLME-
        JOIN.request primitive.

        This attribute, when true, indicates that dot11RadioMeasurementActivated
        is true and that the station capability for Link Measurement is enabled. A
        value of false indicates the station has no Link Measurement capability or
        that the capability is present but is disabled."
    DEFVAL { false }
    ::= { dot11StationConfigEntry 55 }

dot11RMNeighborReportActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute, when true, indicates that dot11RadioMeasurementActivated
        is true and that the station capability for neighbor report is enabled.
        False indicates the station has no neighbor report capability or that the
        capability is present but is disabled."
    DEFVAL { false }
    ::= { dot11StationConfigEntry 56 }

dot11RMParallelMeasurementsActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
```

```
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect for the next MLME-START.request primitive or MLME-
        JOIN.request primitive.

        This attribute, when true, indicates that dot11RadioMeasurementActivated
        is true and that the station capability for Parallel Measurements is
        enabled. False indicates the station has no Parallel Measurements capabil-
        ity or that the capability is present but is disabled."
    DEFVAL { false }
    ::= { dot11StationConfigEntry 57 }

dot11RMRepeatedMeasurementsActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect for the next MLME-START.request primitive or MLME-
        JOIN.request primitive.

        This attribute, when true, indicates that dot11RadioMeasurementActivated
        is true and that the station capability for Repeated Measurements is
        enabled. False indicates the station has no Repeated Measurements capabil-
        ity or that the capability is present but is disabled."
    DEFVAL { false }
    ::= { dot11StationConfigEntry 58 }

dot11RMBeaconPassiveMeasurementActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect for the next MLME-START.request primitive or MLME-
        JOIN.request primitive.

        This attribute, when true, indicates that dot11RadioMeasurementActivated
        is true and that the station capability for Beacon Passive Measurement is
        enabled. False indicates the station has no Beacon Passive Measurement
        capability or that the capability is present but is disabled."
    DEFVAL { false }
    ::= { dot11StationConfigEntry 59 }

dot11RMBeaconActiveMeasurementActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect for the next MLME-START.request primitive or MLME-
        JOIN.request primitive.

        This attribute, when true, indicates that dot11RadioMeasurementActivated
        is true and that the station capability for Beacon Active Measurement is
        enabled. False indicates the station has no Beacon Active Measurement
        capability or that the capability is present but is disabled."
    DEFVAL { false }
    ::= { dot11StationConfigEntry 60 }

dot11RMBeaconTableMeasurementActivated OBJECT-TYPE
```

```
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect for the next MLME-START.request primitive or MLME-
        JOIN.request primitive.

        This attribute, when true, indicates that dot11RadioMeasurementActivated
        is true and that the station capability for Beacon Table Measurement is
        enabled. False indicates the station has no Beacon Table Measurement capa-
        bility or that the capability is present but is disabled."
    DEFVAL { false }
    ::= { dot11StationConfigEntry 61 }

dot11RMBeaconMeasurementReportingConditionsActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect for the next MLME-START.request primitive or MLME-
        JOIN.request primitive.

        This attribute, when true, indicates that dot11RadioMeasurementActivated
        is true and that the station capability for Beacon Measurement Reporting
        Conditions is enabled. False indicates the station has no Beacon Measure-
        ment Reporting Conditions capability or that the capability is present but
        is disabled."
    DEFVAL { false }
    ::= { dot11StationConfigEntry 62 }

dot11RMFrameMeasurementActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect for the next MLME-START.request primitive or MLME-
        JOIN.request primitive.

        This attribute, when true, indicates that dot11RadioMeasurementActivated
        is true and that the station capability for Frame Measurement is enabled.
        False indicates the station has no Frame Measurement capability or that
        the capability is present but is disabled."
    DEFVAL { false }
    ::= { dot11StationConfigEntry 63 }

dot11RMChannelLoadMeasurementActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect for the next MLME-START.request primitive or MLME-
        JOIN.request primitive.

        This attribute, when true, indicates that dot11RadioMeasurementActivated
        is true and that the station capability for Channel Load Measurement is
        enabled. False indicates the station has no Channel Load Measurement capa-
```

```
      bility or that the capability is present but is disabled."
   DEFVAL { false }
   ::= { dot11StationConfigEntry 64 }

dot11RMNoiseHistogramMeasurementActivated OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect for the next MLME-START.request primitive or MLME-
      JOIN.request primitive.

      This attribute, when true, indicates that dot11RadioMeasurementActivated
      is true and that the station capability for Noise Histogram Measurement is
      enabled. False indicates the station has no Noise Histogram Measurement
      capability or that the capability is present but is disabled."
   DEFVAL { false }
   ::= { dot11StationConfigEntry 65 }

dot11RMStatisticsMeasurementActivated OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute, when true, indicates that dot11RadioMeasurementActivated
      is true and that the station capability for Statistics Measurement is
      enabled. False indicates the station has no Statistics Measurement capa-
      bility or that the capability is present but is disabled."
   DEFVAL { false }
   ::= { dot11StationConfigEntry 66 }

dot11RMLCIMeasurementActivated OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect for the next MLME-START.request primitive or MLME-
      JOIN.request primitive.

      This attribute, when true, indicates that dot11RadioMeasurementActivated
      is true and that the station capability for LCI Measurement is enabled.
      False indicates the station has no LCI Measurement capability or that the
      capability is present but is disabled."
   DEFVAL { false }
   ::= { dot11StationConfigEntry 67 }

dot11RMLCIAzimuthActivated OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect for the next MLME-START.request primitive or MLME-
      JOIN.request primitive.
```

```
        This attribute, when true, indicates that dot11RadioMeasurementActivated
        is true and that the station capability for LCI Azimuth Measurement is
        enabled. False indicates the station has no LCI Azimuth Measurement capa-
        bility or that the capability is present but is disabled."
    DEFVAL { false }
    ::= { dot11StationConfigEntry 68 }

dot11RMTransmitStreamCategoryMeasurementActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute, when true, indicates that dot11RadioMeasurementActivated
        is true and that the station capability for Transmit Stream/Category Mea-
        surement is enabled. False indicates the station has no Transmit Stream/
        Category Measurement capability or that the capability is present but is
        disabled."
    DEFVAL { false }
    ::= { dot11StationConfigEntry 69 }

dot11RMTriggeredTransmitStreamCategoryMeasurementActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute, when true, indicates that dot11RadioMeasurementActivated
        is true and that the station capability for Triggered Transmit Stream/Cat-
        egory Measurement is enabled. False indicates the station has no Triggered
        Transmit Stream/Category Measurement capability or that the capability is
        present but is disabled."
    DEFVAL { false }
    ::= { dot11StationConfigEntry 70 }

dot11RMAPChannelReportActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect for the next MLME-START.request primitive or MLME-
        JOIN.request primitive.

        This attribute, when true, indicates that dot11RadioMeasurementActivated
        is true and that the station capability for AP Channel Report is enabled.
        False indicates the station has no AP Channel Report capability or that
        the capability is present but is disabled."
    DEFVAL { false }
    ::= { dot11StationConfigEntry 71 }

dot11RMMIBActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute, when true, indicates that dot11RadioMeasurementActivated
        is true and that the station capability for RM MIB is enabled. False indi-
        cates the station has no RM MIB capability or that the capability is pres-
        ent but is disabled. See RM MIB details."
    DEFVAL { false }
    ::= { dot11StationConfigEntry 72 }

dot11RMMaxMeasurementDuration OBJECT-TYPE
    SYNTAX Unsigned32(0 .. 7)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute indicates the maximum measurement duration for operating
        channel measurements, where
        Max Measurement Duration in TUs =
          (2 ** (dot11RMMaxMeasurementDuration - 4)) * BeaconInterval

        Further details are provided in 10.11.4"
    ::= { dot11StationConfigEntry 73 }

dot11RMNonOperatingChannelMaxMeasurementDuration OBJECT-TYPE
    SYNTAX Unsigned32(0 .. 7)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect for the next MLME-START.request primitive or MLME-
        JOIN.request primitive.

        This attribute indicates the maximum measurement duration for nonoperating
        channel measurements, where

        Non-OpMax Measurement Duration in TUs =
          (2 ** (dot11RMNonOperatingChannelMaxMeasurementDuration - 4))
          * BeaconInterval

        Further details are provided in 10.11.4"
    ::= { dot11StationConfigEntry 74 }

dot11RMMeasurementPilotTransmissionInformationActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute, when true, indicates that dot11RadioMeasurementActivated
        is true and that the station capability for Measurement Pilot Transmission
        information is enabled. False indicates the station has no Measurement
        Pilot Transmission information capability or that the capability is pres-
        ent but is disabled."
    DEFVAL { false }
    ::= { dot11StationConfigEntry 75 }
```

```
dot11RMMeasurementPilotActivated OBJECT-TYPE
    SYNTAX Unsigned32(0 .. 7)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute indicates the station capability for Measurement Pilot. The
        value 0 indicates the station has no Measurement Pilot capability or that
        the capability is present but is disabled. Activated values 1-7 are
        defined in Table 10-7."
    DEFVAL { 0 }
    ::= { dot11StationConfigEntry 76 }

dot11RMNeighborReportTSFOffsetActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute, when true, indicates that dot11RadioMeasurementActivated
        is true and that the station capability for Neighbor Report TSF Offset is
        enabled. False indicates the station has no Neighbor Report TSF Offset
        capability or that the capability is present but is disabled."
    DEFVAL { false }
    ::= { dot11StationConfigEntry 77 }

dot11RMRCPIMeasurementActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect for the next MLME-START.request primitive or MLME-
        JOIN.request primitive.

        This attribute, when true, indicates that dot11RadioMeasurementActivated
        is true and that the station capability for RCPI Measurement is enabled.
        False indicates the station has no RCPI Measurement capability or that the
        capability is present but is disabled."
    DEFVAL { false }
    ::= { dot11StationConfigEntry 78 }

dot11RMRSNIMeasurementActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute, when true, indicates that dot11RadioMeasurementActivated
        is true and that the station capability for RSNI Measurement is enabled.
        False indicates the station has no RSNI Measurement capability or that the
        capability is present but is disabled."
    DEFVAL { false }
    ::= { dot11StationConfigEntry 79 }
```

```
dot11RMBSSAverageAccessDelayActivated OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect for the next MLME-START.request primitive or MLME-
      JOIN.request primitive.

      This attribute, when true, indicates that dot11RadioMeasurementActivated
      is true and that the station capability for BSS Average Access Delay is
      enabled. False indicates the station has no BSS Average Access Delay capa-
      bility or that the capability is present but is disabled."
   DEFVAL { false }
   ::= { dot11StationConfigEntry 80 }

dot11RMBSSAvailableAdmissionCapacityActivated OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect for the next MLME-START.request primitive or MLME-
      JOIN.request primitive.

      This attribute, when true, indicates that dot11RadioMeasurementActivated
      is true and that the station capability for BSS Available Admission Capac-
      ity is enabled. False indicates the station has no BSS Available Admission
      Capacity capability or that the capability is present but is disabled."
   DEFVAL { false }
   ::= { dot11StationConfigEntry 81 }

dot11RMAntennaInformationActivated OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect for the next MLME-START.request primitive or MLME-
      JOIN.request primitive.

      This attribute, when true, indicates that dot11RadioMeasurementActivated
      is true and that the station capability for Antenna information is
      enabled. False indicates the station has no Antenna information capability
      or that the capability is present but is disabled."
   DEFVAL { false }
   ::= { dot11StationConfigEntry 82 }

dot11FastBSSTransitionImplemented OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      This object indicates if the entity is fast BSS transition capable."
   ::= { dot11StationConfigEntry 83 }

dot11LCIDSEImplemented OBJECT-TYPE
```

```
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute, when true, indicates that the station implementation is
        capable of supporting LCI DSE. The capability is disabled, otherwise."
    DEFVAL { false }
    ::= { dot11StationConfigEntry 84 }

dot11LCIDSERequired OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by the SME.

        A STA uses the Dependent Station Enablement procedures if this attribute
        is true."
    DEFVAL { false }
    ::= { dot11StationConfigEntry 85 }

dot11DSERequired OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by the SME.

        This attribute, when true, indicates that the station operation is depen-
        dent on enablement from enabling STAs. This attribute, when false, indi-
        cates that the station operation does not depend on enablement from STAs."
    DEFVAL { false }
    ::= { dot11StationConfigEntry 86 }

dot11ExtendedChannelSwitchActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by the SME when the device is initialized for operation in a
        band defined by an Operating Class.

        This attribute, when true, indicates that the station implementation is
        capable of supporting Extended Channel Switch Announcement. The capability
        is disabled, otherwise."
    DEFVAL { false }
    ::= { dot11StationConfigEntry 87 }

--************************************************************
--* Management frame protection MIBs
--************************************************************
dot11RSNAProtectedManagementFramesActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
```

```
      Changes take effect as soon as practical in the implementation.

      This variable indicates whether this STA enables management frame protec-
      tion."
   DEFVAL { false }
   ::= { dot11StationConfigEntry 88}

dot11RSNAUnprotectedManagementFramesAllowed OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This variable indicates whether this STA supports RSNA STAs which do not
      provide robust management frames protection."
   DEFVAL { true }
   ::= { dot11StationConfigEntry 89}

--**********************************************************
--* SA Query Procedure MIBs
--**********************************************************
dot11AssociationSAQueryMaximumTimeout OBJECT-TYPE
   SYNTAX Unsigned32 (1..4294967295)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute specifies the number of time units (TUs) that an AP can
      wait, from the scheduling of the first SA Query Request to allow associa-
      tion process to be started without starting additional SA Query procedure
      if a successful SA Query Response is not received."
   DEFVAL { 1000 }
   ::= { dot11StationConfigEntry 90}

dot11AssociationSAQueryRetryTimeout OBJECT-TYPE
   SYNTAX Unsigned32 (1..4294967295)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute specifies the number of time units (TUs) that an AP waits
      between issuing two subsequent MLME-SAQuery.request primitive primitives."
   DEFVAL { 201 }
   ::= { dot11StationConfigEntry 91}

dot11HighThroughputOptionImplemented OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      This attribute indicates whether the entity is HT Capable."
   ::= { dot11StationConfigEntry 92}
```

```
dot11RSNAPBACRequired OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This variable indicates whether this STA requires the Protection of Block
      Ack agreements."
   DEFVAL { false }
   ::= { dot11StationConfigEntry 93}

dot11PSMPOptionImplemented OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      This attribute, when true, indicates that the station implementation is
      capable of supporting PSMP."
   DEFVAL { false }
   ::= { dot11StationConfigEntry 94 }

dot11TunneledDirectLinkSetupImplemented OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This attribute, when true, indicates that the STA implementation is capa-
      ble of supporting Tunneled DirectLink Setup."
   DEFVAL { false }
   ::= { dot11StationConfigEntry 95 }

dot11TDLSPeerUAPSDBufferSTAActivated OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This attribute, when true, indicates that the STA implementation is capa-
      ble of supporting TDLS Peer U-APSD."
   DEFVAL { false }
   ::= { dot11StationConfigEntry 96 }

dot11TDLSPeerPSMActivated OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This attribute, when true, indicates that the STA implementation is capa-
      ble of supporting TDLS Peer PSM."
   DEFVAL { false }
   ::= { dot11StationConfigEntry 97 }

dot11TDLSPeerUAPSDIndicationWindow OBJECT-TYPE
   SYNTAX Unsigned32 (1..256)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This attribute indicates the minimum interval in Beacon Intervals between
```

```
        successive Peer Traffic Indication frames."
    DEFVAL { 1 }
    ::= { dot11StationConfigEntry 98 }

dot11TDLSChannelSwitchingActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This attribute, when true, indicates that the STA implementation is capa-
        ble of supporting TDLS Channel Switching."
    DEFVAL { false }
    ::= { dot11StationConfigEntry 99 }

dot11TDLSPeerSTAMissingAckRetryLimit OBJECT-TYPE
    SYNTAX Unsigned32 (1..100)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This attribute indicates the number of times the TDLS STA may retry a
        frame for which it does not receive an ACK from TDLS peer STA in power
        save mode after the TDLS peer STA does not receive an ACK to an individu-
        ally addressed MPDU sent with the EOSP equal to 1 or to an individually
        addressed MPDU that initiated a channel switch."
    DEFVAL { 3 }
    ::= { dot11StationConfigEntry 100 }

dot11TDLSResponseTimeout OBJECT-TYPE
    SYNTAX Unsigned32 (1..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This attribute indicates the amount of time in units of seconds the STA
        waits before timing out a TDLS setup request."
    DEFVAL { 5 }
    ::= { dot11StationConfigEntry 101 }

dot11OCBActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "A STA uses the defined outside the context of a BSS procedures if and
        only if this attribute is true. The default value of this attribute is
        false."
    DEFVAL { false }
    ::= { dot11StationConfigEntry 102 }

dot11TDLSProbeDelay OBJECT-TYPE
    SYNTAX Unsigned32 (1..65535)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This attribute indicates the amount of time in units of microseconds the
        STA waits before transmitting on a new channel, in the absence of traffic
        on the channel that causes a CCA state to be created."
    DEFVAL { 1000 }
    ::= { dot11StationConfigEntry 103 }

dot11TDLSDiscoveryRequestWindow OBJECT-TYPE
    SYNTAX Unsigned32 (1..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
```

```
      "This attribute indicates the minimum number of DTIM intervals between
      successive TDLS Discovery Request frames."
   DEFVAL { 2 }
   ::= { dot11StationConfigEntry 104 }

dot11TDLSACDeterminationInterval OBJECT-TYPE
   SYNTAX Unsigned32 (1..255)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This attribute indicates the number of seconds during which the lowest AC
      of transmitted traffic is determined."
   DEFVAL { 1 }
   ::= { dot11StationConfigEntry 105 }

dot11WirelessManagementImplemented OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      This attribute, when true, indicates that the station implementation is
      capable of supporting one or more wireless network management services."
   ::= { dot11StationConfigEntry 106}

dot11BssMaxIdlePeriod OBJECT-TYPE
   SYNTAX Unsigned32 (1..65535)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      This attribute indicates that the number of 1000 TUs that pass before an
      AP disassociates an inactive non-AP STA. This value is transmitted in the
      Association Response and Reassociation Response frames."
   ::= { dot11StationConfigEntry 107}

dot11BssMaxIdlePeriodOptions OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(1))
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      This attribute indicates the options associated with the BSS Max Idle
      capability. This value is transmitted in the Association Response and
      Reassociation Response frames."
   ::= { dot11StationConfigEntry 108}

dot11TIMBroadcastInterval OBJECT-TYPE
   SYNTAX Unsigned32 (0..255)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.
```

```
        This attribute indicates the number of beacon periods between TIM frame
        transmissions. A value of 0 disables TIM Broadcast for the requesting sta-
        tion."
    DEFVAL { 0 }
    ::= { dot11StationConfigEntry 109}

dot11TIMBroadcastOffset OBJECT-TYPE
    SYNTAX Integer32 (-214748364..214748363)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        This attribute indicates the offset in microseconds with a tolerance of +/
        - 4 microseconds relative to the TBTT for which a TIM frame is scheduled
        for transmission. The field contains a signed integer."
    DEFVAL { 0 }
    ::= { dot11StationConfigEntry 110}

dot11TIMBroadcastHighRateTIMRate OBJECT-TYPE
    SYNTAX Unsigned32 (0..65535)
    UNITS "0.5 Mb/s"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        This attribute indicates the rate used to transmit the high data rate
        TIM frame, in units of 0.5 Mb/s. A value of 0 indicates that the high rate
        TIM frame is not transmitted."
    DEFVAL { 0 }
    ::= { dot11StationConfigEntry 111}

dot11TIMBroadcastLowRateTIMRate OBJECT-TYPE
    SYNTAX Unsigned32 (0..65535)
    UNITS "0.5 Mb/s"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        This attribute indicates the rate used to transmit the low data rate
        TIM frame, in units of 0.5 Mb/s. A value of 0 indicates that the low rate
        TIM frame is not transmitted."
    DEFVAL { 0 }
    ::= { dot11StationConfigEntry 112}

dot11StatsMinTriggerTimeout OBJECT-TYPE
    SYNTAX Unsigned32 (10..7200)
    UNITS "seconds"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        This attribute indicates the minimum allowable value for Triggered Time-
```

```
      out. A Triggered STA Statistics report is generated by the STA after the
      timeout if none of the trigger conditions are satisfied."
   DEFVAL { 10 }
   ::= { dot11StationConfigEntry 113 }

dot11RMCivicMeasurementActivated OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      This attribute, when true, indicates that dot11RadioMeasurementActivated
      is true and that the station capability for Location Civic Measurement is
      enabled. False indicates the station has no Location Civic Measurement
      capability or that the capability is present but is disabled."
   DEFVAL { false }
   ::= { dot11StationConfigEntry 114 }

dot11RMIdentifierMeasurementActivated OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      This attribute, when true, indicates that dot11RadioMeasurementActivated
      is true and that the station capability for Location Identifier Measure-
      ment is enabled. False indicates the station has no Location Identifier
      Measurement capability or that the capability is present but is disabled."
   DEFVAL { false }
   ::= { dot11StationConfigEntry 115 }

dot11TimeAdvertisementDTIMInterval OBJECT-TYPE
   SYNTAX Unsigned32 (1..255)
   UNITS "dtims"
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      This attribute indicates the interval in number of DTIMS when the Time
      Advertisement element is included in beacon frames."
   DEFVAL { 1 }
   ::= { dot11StationConfigEntry 116 }

dot11InterworkingServiceImplemented OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.

      Its value is determined by device capabilities.

      This attribute when true, indicates the STA is capable of interworking
      with external networks. A STA setting this to true implements Interworking
      Service. When this is false, the STA does not implement Interworking Ser-
```

```
        vice."
    DEFVAL {false}
    ::= { dot11StationConfigEntry 117 }

dot11InterworkingServiceActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by an external management entity or the SME. Changes take
        effect as soon as practical in the implementation.

        This attribute when true, indicates the capability of the STA to interwork
        with external networks is enabled. The capability is disabled otherwise."
    DEFVAL {false}
    ::= { dot11StationConfigEntry 118 }

dot11QosMapImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.

        Its value is determined by device capabilities.

        This attribute available at STAs, when true, indicates the STA is capable
        of supporting the QoS Map procedures. When this is set to false, the STA
        does not implement QoS Map procedures."
    DEFVAL {false}
    ::= { dot11StationConfigEntry 119 }

dot11QosMapActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by an external management entity or the SME. Changes take
        effect as soon as practical in the implementation.

        This attribute, when true, indicates the capability of the STA to support
        QoS Map procedures is enabled. The capability is disabled otherwise."
    DEFVAL {false}
    ::= { dot11StationConfigEntry 120 }

dot11EBRImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.

        Its value is determined by device capabilities.

        This attribute available at STAs, when true, indicates the STA is capable
        of supporting Expedited Bandwidth Request procedures. When this is false,
        the STA does not implement Expedited Bandwidth Request procedures."
    DEFVAL {false}
    ::= { dot11StationConfigEntry 121 }
```

```
dot11EBRActivated OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.

      It is written by an external management entity or the SME. Changes take
      effect as soon as practical in the implementation.

      This attribute, when true, indicates the capability of the STA to support
      Expedited Bandwidth Request procedures is enabled. The capability is dis-
      abled otherwise."
   DEFVAL {false}
   ::= { dot11StationConfigEntry 122 }

dot11ESNetwork OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.

      It is written by an external management entity or the SME. Changes take
      effect as soon as practical in the implementation.

      The emergency services access network type set to true indicates that the
      BSS is used exclusively for the purposes of accessing emergency services.
      This object is not used by non-AP STAs."
   ::= { dot11StationConfigEntry 123 }

dot11SSPNInterfaceImplemented OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.

      Its value is determined by device capabilities.

      This attribute when true, indicates the AP is capable of SSPN Interface
      service. When this is false, the STA does not implement SSPN Interface
      Service. This object is not used by non-AP STAs. The default value of this
      attribute is false."
   DEFVAL {false}
   ::= { dot11StationConfigEntry 124 }

dot11SSPNInterfaceActivated OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.

      It is written by an external management entity or the SME. Changes take
      effect as soon as practical in the implementation.

      This attribute, when true, indicates the capability of the AP to provide
      SSPN Interface service is enabled. The capability is disabled, otherwise.
      The default value of this attribute is false."
   DEFVAL {false}
   ::= { dot11StationConfigEntry 125 }

dot11HESSID OBJECT-TYPE
```

```
    SYNTAX MacAddress
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by an external management entity or the SME. Changes take
        effect for the next MLME-START.request primitive.

        This attribute is used by an AP and is the 6-octet homogeneous ESS identi-
        fier field, whose value is set to one of the BSSIDs in the ESS. It is
        required that the same value of HESSID be used for all BSSs in the homoge-
        neous ESS."
    ::= { dot11StationConfigEntry 126 }

dot11EASImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.

        Its value is determined by device capabilities.

        This attribute when true, indicates the STA is capable of emergency alert
        system notification with external networks. A STA setting this to true
        implements emergency alert system notification. When this is false, the
        STA does not implement emergency alert system notification."
    DEFVAL {false}
    ::= { dot11StationConfigEntry 127 }

dot11EASActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by an external management entity or the SME. Changes take
        effect as soon as practical in the implementation.

        This attribute when true, indicates the STA is capable of supporting emer-
        gency alert system. The capability is disabled otherwise."
    DEFVAL {false}
    ::= { dot11StationConfigEntry 128 }

dot11MSGCFImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.

        Its value is determined by device capabilities.

        This attribute when true, indicates the non-AP STA is capable of support-
        ing the MSGCF procedures defined in 6.4. When false, the non-AP STA does
        not implement MSGCF procedures. This object is not used by APs. The
        default value of this attribute is false."
    DEFVAL {false}
    ::= { dot11StationConfigEntry 129 }

dot11MSGCFActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by an external management entity or the SME. Changes take
        effect as soon as practical in the implementation.

        This attribute, when true, indicates the capability of the non-AP STA to
        provide the MSGCF is enabled. The capability is disabled, otherwise. The
        default value of this attribute is false."
    DEFVAL {false}
    ::= { dot11StationConfigEntry 130 }

dot11TimeAdvertisementTimeError OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(5))
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        This attribute indicates the Time Error value as defined in the Time
        Advertisement element Time Error field when the Time Capabilities field is
        set to 2. This field is included in the Time Advertisement element in Bea-
        con and Probe Response frames."
    DEFVAL { ''H }
    ::= { dot11StationConfigEntry 131 }

dot11TimeAdvertisementTimeValue OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(10))
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        This attribute indicates the TimeAdvertisement Time Value as defined in
        the Time Advertisement element Time Value field when the Time Capabilities
        field is set to 2. The format is defined in Table 8-132 and is included in
        the Time Advertisement element in Beacon and Probe Response frames."
    ::= { dot11StationConfigEntry 132 }

dot11RM3rdPartyMeasurementActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        This attribute, when true, indicates that dot11RadioMeasurementActivated
        is true and that the station capability for Third Party Location Measure-
        ment is enabled. False indicates the station has no Third Party Location
        Measurement capability or that the capability is present but is disabled."
    DEFVAL { false }
    ::= { dot11StationConfigEntry 133 }

dot11RejectUnadmittedTraffic OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect at the next MLME-START.request primitive or  MLME-
        JOIN.request primitive.

        This attribute when true indicates the STA's MA-UNITDATA primitive rejects
        frames whose requested priority corresponds to an AC for which admission
        control is mandatory and for which there is not an admitted TSPEC."
    DEFVAL { false }
    ::= {dot11StationConfigEntry 134}

dot11BSSBroadcastNullCount OBJECT-TYPE
    SYNTAX Unsigned32 (1..64)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute specifies the number of group addressed Null data frames a
        STA may transmit before it changes power management modes."
    DEFVAL { 3 }
    ::= {dot11StationConfigEntry 135}

dot11MeshActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        When this object is true, this indicates that the STA is a mesh STA. Con-
        figuration variables for mesh operation are found in the
        dot11MeshSTAConfigTable."
    ::= { dot11StationConfigEntry 136 }

-- **********************************************************************
-- *    End of dot11StationConfig  TABLE
-- **********************************************************************

-- **********************************************************************
-- *    dot11AuthenticationAlgorithms  TABLE
-- **********************************************************************

dot11AuthenticationAlgorithmsTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11AuthenticationAlgorithmsEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "This (conceptual) table of attributes is a set of all the authentication
        algorithms supported by the stations. The following are the default values
        and the associated algorithm:
        Value = 1: Open System
        Value = 2: Shared Key
        Value = 3: Fast BSS Transition (FT)
        Value = 4: Simultaneous authentication of equals (SAE)"
    REFERENCE
        "IEEE Std 802.11-2012, 8.4.1.1"
    ::= { dot11smt 2 }
```

```
dot11AuthenticationAlgorithmsEntry OBJECT-TYPE
    SYNTAX Dot11AuthenticationAlgorithmsEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An Entry (conceptual row) in the Authentication Algorithms Table.

        ifIndex - Each IEEE 802.11 interface is represented by an ifEntry. Inter-
        face tables in this MIB module are indexed by ifIndex."
    INDEX { ifIndex,
    dot11AuthenticationAlgorithmsIndex }
    ::= { dot11AuthenticationAlgorithmsTable  1 }

Dot11AuthenticationAlgorithmsEntry ::=
    SEQUENCE {
        dot11AuthenticationAlgorithmsIndex                  Unsigned32,
        dot11AuthenticationAlgorithm                        INTEGER,
        dot11AuthenticationAlgorithmsActivated              TruthValue }

dot11AuthenticationAlgorithmsIndex OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "The auxiliary variable used to identify instances of the columnar objects
        in the Authentication Algorithms Table."
    ::= { dot11AuthenticationAlgorithmsEntry 1 }

dot11AuthenticationAlgorithm OBJECT-TYPE
    SYNTAX INTEGER {
        openSystem(1),
        sharedKey(2),
        fastBSSTransition(3),
        simultaneousAuthEquals(4) }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute is the authentication algorithm described by this entry in
        the table. The following values can be used here
        Value = 1: Open System
        Value = 2: Shared Key
        Value = 3: Fast BSS Transition (FT)
        Value = 4: Simultaneous authentication of equals (SAE)"
    ::= { dot11AuthenticationAlgorithmsEntry 2 }

dot11AuthenticationAlgorithmsActivated  OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute, when true at a station, enables the acceptance of the
        authentication algorithm described in the corresponding table entry in
        authentication frames received by the station that have odd authentication
        sequence numbers. The default value of this attribute is true when the
        value of dot11AuthenticationAlgorithm is Open System and false otherwise."
```

```
   ::= { dot11AuthenticationAlgorithmsEntry 3 }

-- **********************************************************************
-- *    End of dot11AuthenticationAlgorithms  TABLE
-- **********************************************************************

-- **********************************************************************
-- *    dot11WEPDefaultKeys  TABLE
-- **********************************************************************

dot11WEPDefaultKeysTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11WEPDefaultKeysEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "Conceptual table for WEP default keys. This table contains the four WEP
      default secret key values corresponding to the four possible KeyID values.
      The WEP default secret keys are logically WRITE-ONLY. Attempts to read the
      entries in this table return unsuccessful status and values of null or 0.
      The default value of each WEP default key is null."
   REFERENCE "IEEE Std 802.11-2012, 11.2.2"
   ::= { dot11smt 3 }

dot11WEPDefaultKeysEntry OBJECT-TYPE
   SYNTAX Dot11WEPDefaultKeysEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "An Entry (conceptual row) in the WEP Default Keys Table.

      ifIndex - Each IEEE 802.11 interface is represented by an ifEntry. Inter-
      face tables in this MIB module are indexed by ifIndex."
   INDEX { ifIndex, dot11WEPDefaultKeyIndex }
   ::= { dot11WEPDefaultKeysTable  1 }

Dot11WEPDefaultKeysEntry ::=
   SEQUENCE {
      dot11WEPDefaultKeyIndex                               Unsigned32,
      dot11WEPDefaultKeyValue                               WEPKeytype }

dot11WEPDefaultKeyIndex OBJECT-TYPE
   SYNTAX Unsigned32 (1..4)
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "The auxiliary variable used to identify instances of the columnar objects
      in the WEP Default Keys Table. The value of this variable is equal to the
      WEPDefaultKeyID + 1"
   ::= { dot11WEPDefaultKeysEntry 1 }

dot11WEPDefaultKeyValue OBJECT-TYPE
   SYNTAX WEPKeytype
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      A WEP default secret key value."
   ::= { dot11WEPDefaultKeysEntry 2 }

-- **********************************************************************
-- *    End of dot11WEPDefaultKeys  TABLE
-- **********************************************************************

-- **********************************************************************
-- *    dot11WEPKeyMappings  TABLE
-- **********************************************************************

dot11WEPKeyMappingsTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11WEPKeyMappingsEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "Conceptual table for WEP Key Mappings. The MIB supports the ability to
      share a separate WEP key for each RA/TA pair. The Key Mappings Table con-
      tains zero or one entry for each MAC address and contains two fields for
      each entry: WEPOn and the corresponding WEP key. The WEP key mappings are
      logically WRITE-ONLY. Attempts to read the entries in this table return
      unsuccessful status and values of null or 0. The default value for all
      WEPOn fields is false."
   REFERENCE "IEEE Std 802.11-2012, 11.2.2"
   ::= { dot11smt 4 }

dot11WEPKeyMappingsEntry OBJECT-TYPE
   SYNTAX Dot11WEPKeyMappingsEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "An Entry (conceptual row) in the WEP Key Mappings Table.

      ifIndex - Each IEEE 802.11 interface is represented by an ifEntry. Inter-
      face tables in this MIB module are indexed by ifIndex."
   INDEX { ifIndex, dot11WEPKeyMappingIndex }
   ::= { dot11WEPKeyMappingsTable  1 }

Dot11WEPKeyMappingsEntry ::=
   SEQUENCE {
      dot11WEPKeyMappingIndex                              Unsigned32,
      dot11WEPKeyMappingAddress                            MacAddress,
      dot11WEPKeyMappingWEPOn                              TruthValue,
      dot11WEPKeyMappingValue                              WEPKeytype,
      dot11WEPKeyMappingStatus                             RowStatus }

dot11WEPKeyMappingIndex OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "The auxiliary variable used to identify instances of the columnar objects
      in the WEP Key Mappings Table."
   ::= { dot11WEPKeyMappingsEntry 1 }

dot11WEPKeyMappingAddress OBJECT-TYPE
   SYNTAX MacAddress
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "The MAC address of the STA for which the values from this key mapping
      entry are to be used."
   ::= { dot11WEPKeyMappingsEntry 2 }

dot11WEPKeyMappingWEPOn OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
```

```
      "Boolean as to whether WEP is to be used when communicating with the
      dot11WEPKeyMappingAddress STA."
   ::= { dot11WEPKeyMappingsEntry 3 }

dot11WEPKeyMappingValue OBJECT-TYPE
   SYNTAX WEPKeytype
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "A WEP secret key value."
   ::= { dot11WEPKeyMappingsEntry 4 }

dot11WEPKeyMappingStatus OBJECT-TYPE
   SYNTAX RowStatus
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "The status column used for creating, modifying, and deleting instances of
      the columnar objects in the WEP key mapping Table."
   DEFVAL { active }
   ::= { dot11WEPKeyMappingsEntry 5 }

-- **********************************************************************
-- *    End of dot11WEPKeyMappings  TABLE
-- **********************************************************************

-- **********************************************************************
-- *    dot11Privacy TABLE
-- **********************************************************************

dot11PrivacyTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11PrivacyEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "Group containing attributes concerned with IEEE 802.11 Privacy. Created
      as a table to allow multiple instantiations on an agent."
   ::= { dot11smt 5 }

dot11PrivacyEntry OBJECT-TYPE
   SYNTAX Dot11PrivacyEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "An entry in the dot11PrivacyTable Table.

      ifIndex - Each IEEE 802.11 interface is represented by an ifEntry. Inter-
      face tables in this MIB module are indexed by ifIndex."
   INDEX { ifIndex }
   ::= { dot11PrivacyTable 1 }

Dot11PrivacyEntry ::=
   SEQUENCE {
      dot11PrivacyInvoked                             TruthValue,
      dot11WEPDefaultKeyID                            Unsigned32,
      dot11WEPKeyMappingLengthImplemented             Unsigned32,
      dot11ExcludeUnencrypted                         TruthValue,
      dot11WEPICVErrorCount                           Counter32,
      dot11WEPExcludedCount                           Counter32,
      dot11RSNAActivated                              TruthValue,
      dot11RSNAPreauthenticationActivated             TruthValue }

dot11PrivacyInvoked OBJECT-TYPE
   SYNTAX TruthValue
```

```
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        When this attribute is true, it indicates that some level of security is
        invoked for transmitting data frames. For WEP-only clients, the security
        mechanism used is WEP.

        For RSNA-capable clients, an additional variable dot11RSNAActivated indi-
        cates whether RSNA is enabled. If dot11RSNAActivated is false or the MIB
        variable does not exist, the security mechanism invoked is WEP; if
        dot11RSNAActivated is true, RSNA security mechanisms invoked are config-
        ured in the dot11RSNAConfigTable."
    DEFVAL { false }
    ::= { dot11PrivacyEntry 1 }

dot11WEPDefaultKeyID  OBJECT-TYPE
    SYNTAX Unsigned32 (0..3)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute indicates the use of the first, second, third, or fourth
        element of the WEPDefaultKeys array when equal to values of zero, one,
        two, or three."
    REFERENCE "IEEE Std 802.11-2012, 11.2.2"
    DEFVAL { 0 }
    ::= { dot11PrivacyEntry 2 }

dot11WEPKeyMappingLengthImplemented  OBJECT-TYPE
    SYNTAX Unsigned32 (10..4294967295)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        The maximum number of tuples that dot11WEPKeyMappings can hold."
    REFERENCE "IEEE Std 802.11-2012, 11.2.2"
    ::= { dot11PrivacyEntry 3 }

dot11ExcludeUnencrypted  OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        When this attribute is true, the STA does not indicate at the MAC service
        interface received MSDUs that have the Protected Frame subfield of the
        Frame Control field equal to 0. When this attribute is false, the STA may
        accept MSDUs that have the Protected Frame subfield of the Frame Control
        field equal to 0."
    DEFVAL { false }
    ::= { dot11PrivacyEntry 4 }
```

```
dot11WEPICVErrorCount  OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the MAC when an ICV error is detected.

      This counter increments when a frame is received with the Protected Frame
      subfield of the Frame Control field equal to one and the value of the ICV
      as received in the frame does not match the ICV value that is calculated
      for the contents of the received frame. ICV errors for TKIP are not
      counted in this variable but in dot11RSNAStatsTKIPICVErrors."
   ::= { dot11PrivacyEntry 5 }

dot11WEPExcludedCount  OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the MAC when a bad frame is received.

      This counter increments when a frame is received with the Protected Frame
      subfield of the Frame Control field equal to 0 and the value of
      dot11ExcludeUnencrypted causes that frame to be discarded."
   ::= { dot11PrivacyEntry 6 }

dot11RSNAActivated OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect for the next MLME-START.request primitive or MLME-
      JOIN.request primitive.

      When this object is true, this indicates that RSNA is enabled on this
      entity. The entity advertises the RSN  element in its Beacon and Probe
      Response frames. Configuration variables for RSNA operation are found in
      the dot11RSNAConfigTable.

      This object requires that dot11PrivacyInvoked also be equal to true."
   ::= { dot11PrivacyEntry 7 }

dot11RSNAPreauthenticationActivated OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      When this object is true, this indicates that RSNA preauthentication is
      enabled on this entity.

      This object requires that dot11RSNAActivated also be equal to true."
   ::= { dot11PrivacyEntry 8 }

-- **********************************************************************
-- *    End of dot11Privacy  TABLE
-- **********************************************************************
-- **********************************************************************
-- *    dot11SMTnotification Objects
-- **********************************************************************

dot11SMTnotification OBJECT IDENTIFIER ::= { dot11smt 6 }

dot11Disassociate NOTIFICATION-TYPE
   OBJECTS { ifIndex, dot11DisassociateReason, dot11DisassociateStation }
   STATUS current
   DESCRIPTION
      "The disassociate notification is sent when the STA sends a Disassociation
      frame. The value of the notification includes the MAC address of the MAC
      to which the Disassociation frame was sent and the reason for the disasso-
      ciation.

      ifIndex - Each IEEE 802.11 interface is represented by an ifEntry. Inter-
      face tables in this MIB module are indexed by ifIndex."
   ::= { dot11SMTnotification 0 1 }

dot11Deauthenticate NOTIFICATION-TYPE
   OBJECTS { ifIndex, dot11DeauthenticateReason, dot11DeauthenticateStation }
   STATUS current
   DESCRIPTION
      "The deauthenticate notification is sent when the STA sends a Deauthenti-
      cation frame. The value of the notification includes the MAC address of
      the MAC to which the Deauthentication frame was sent and the reason for
      the deauthentication.

      ifIndex - Each IEEE 802.11 interface is represented by an ifEntry. Inter-
      face tables in this MIB module are indexed by ifIndex."
   ::= { dot11SMTnotification 0 2 }

dot11AuthenticateFail NOTIFICATION-TYPE
   OBJECTS {
      ifIndex, dot11AuthenticateFailStatus, dot11AuthenticateFailStation }
   STATUS current
   DESCRIPTION
      "The authenticate failure notification is sent when the STA sends an
      Authentication frame with a status code other than 'successful'. The value
      of the notification includes the MAC address of the MAC to which the
      Authentication frame was sent and the reason for the authentication fail-
      ure.

      ifIndex - Each IEEE 802.11 interface is represented by an ifEntry. Inter-
      face tables in this MIB module are indexed by ifIndex."
   ::= { dot11SMTnotification 0 3 }

dot11Associate NOTIFICATION-TYPE
   OBJECTS { ifIndex, dot11AssociateStation, dot11AssociateID}
   STATUS current
   DESCRIPTION
      "The associate notification is sent when the STA sends an Association
      Response frame with a status code equal to 'successful'. The value of the
      notification includes the MAC address of the MAC to which the Association
      Response frame was sent and the Association ID. ifIndex - Each 802.11
      interface is represented by an ifEntry. Interface tables in this MIB mod-
      ule are indexed by ifIndex."
   ::= { dot11SMTnotification 0 4 }

dot11AssociateFailed NOTIFICATION-TYPE
   OBJECTS { ifIndex, dot11AssociateFailStatus, dot11AssociateFailStation }
   STATUS current
   DESCRIPTION
```

```
        "The associate failed notification is sent when the STA sends an Associa-
        tion Response frame with a status code other than 'successful'. The value
        of the notification includes the MAC address of the MAC to which the Asso-
        ciation Response frame was sent and the reason for the association fail-
        ure. ifIndex - Each 802.11 interface is represented by an ifEntry.
        Interface tables in this MIB module are indexed by ifIndex."
    ::= { dot11SMTnotification 0 5 }

dot11Reassociate NOTIFICATION-TYPE
    OBJECTS { ifIndex, dot11ReassociateStation, dot11ReassociateID}
    STATUS current
    DESCRIPTION
        "The reassociate notification is sent when the STA sends an Reassociation
        Response frame with a status code equal to 'successful'. The value of the
        notification includes the MAC address of the MAC to which the Reassocia-
        tion Response frame was sent and the Reassociation ID. ifIndex - Each
        802.11 interface is represented by an ifEntry. Interface tables in this
        MIB module are indexed by ifIndex."
    ::= { dot11SMTnotification 0 6 }

dot11ReassociateFailed NOTIFICATION-TYPE
    OBJECTS { ifIndex, dot11ReassociateFailStatus, dot11ReassociateStation }
    STATUS current
    DESCRIPTION
        "The reassociate failed notification is sent when the STA sends an Reasso-
        ciation Response frame with a status code other than 'successful'. The
        value of the notification includes the MAC address of the MAC to which the
        Reassociation Response frame was sent and the reason for the reassociation
        failure. ifIndex - Each 802.11 interface is represented by an ifEntry.
        Interface tables in this MIB module are indexed by ifIndex."
    ::= { dot11SMTnotification 0 7 }
-- **********************************************************************
-- *    End of dot11SMTnotification Objects
-- **********************************************************************

-- **********************************************************************
-- * dot11MultiDomainCapability TABLE
-- **********************************************************************

dot11MultiDomainCapabilityTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11MultiDomainCapabilityEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "This (conceptual) table of attributes for cross-domain mobility."
    ::= { dot11smt 7 }

dot11MultiDomainCapabilityEntry OBJECT-TYPE
    SYNTAX Dot11MultiDomainCapabilityEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry (conceptual row) in the Multiple Domain Capability Table.

        IfIndex - Each IEEE 802.11 interface is represented by an ifEntry. Inter-
        face tables in this MIB are indexed by ifIndex."
    INDEX { ifIndex, dot11MultiDomainCapabilityIndex }
    ::= { dot11MultiDomainCapabilityTable 1 }

Dot11MultiDomainCapabilityEntry ::=
    SEQUENCE {
        dot11MultiDomainCapabilityIndex                      Unsigned32,
        dot11FirstChannelNumber                              Unsigned32,
        dot11NumberofChannels                                Unsigned32,
       dot11MaximumTransmitPowerLevel                        Integer32 }

dot11MultiDomainCapabilityIndex OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
       "The auxiliary variable used to identify instances of the columnar objects
       in the Multi Domain Capability Table."
    ::= { dot11MultiDomainCapabilityEntry 1 }

dot11FirstChannelNumber OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
       "This is a control variable.
       It is written by an external management entity.
       Changes take effect as soon as practical in the implementation.

       This attribute indicates the value of the lowest channel number in the
       subband for the associated domain country string."
    DEFVAL { 0 }
    ::= { dot11MultiDomainCapabilityEntry 2 }

dot11NumberofChannels OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
       "This is a control variable.
       It is written by an external management entity.
       Changes take effect as soon as practical in the implementation.

       This attribute indicates the value of the total number of channels allowed
       in the subband for the associated domain country string."
    DEFVAL { 0 }
    ::= { dot11MultiDomainCapabilityEntry 3 }

dot11MaximumTransmitPowerLevel OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
       "This is a control variable.
       It is written by an external management entity.
       Changes take effect as soon as practical in the implementation.

       This attribute indicates the maximum transmit power, in dBm, allowed in
       the subband for the associated domain country string."
    DEFVAL { 0 }
    ::= { dot11MultiDomainCapabilityEntry 4 }

-- ********************************************************************
-- * End of dot11MultiDomainCapability TABLE
-- ********************************************************************
-- ********************************************************************
-- * dot11SpectrumManagement TABLE
-- ********************************************************************
dot11SpectrumManagementTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11SpectrumManagementEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
```

```
        "(Conceptual) table of attributes for spectrum management"
    ::= {dot11smt 8}

dot11SpectrumManagementEntry OBJECT-TYPE
    SYNTAX Dot11SpectrumManagementEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry (conceptual row) in the Spectrum Management Table.

        IfIndex - Each 802.11 interface is represented by an ifEntry. Interface
        tables in this MIB are indexed by ifIndex."
    INDEX {ifIndex, dot11SpectrumManagementIndex}
    ::= { dot11SpectrumManagementTable 1 }

Dot11SpectrumManagementEntry ::=
    SEQUENCE {
        dot11SpectrumManagementIndex                        Unsigned32,
        dot11MitigationRequirement                          Integer32,
        dot11ChannelSwitchTime                              Unsigned32,
        dot11PowerCapabilityMaxImplemented                  Integer32,
        dot11PowerCapabilityMinImplemented                  Integer32 }

dot11SpectrumManagementIndex OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "The auxiliary variable used to identify instances of the columnar objects
        in the Spectrum Management Table."
    ::= { dot11SpectrumManagementEntry 1 }

dot11MitigationRequirement OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute indicates the mitigation requirement in dB required."
    DEFVAL { 3 }
    ::= { dot11SpectrumManagementEntry 2 }

dot11ChannelSwitchTime OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute indicates assumed channel switch time, measured in TUs. The
        unit of this attribute is TUs. The minimum value of this attribute is 1
        TU."
    DEFVAL { 2 }
    ::= { dot11SpectrumManagementEntry 3 }

dot11PowerCapabilityMaxImplemented OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
```

```
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      This attribute indicates the maximum transmit Power Capability of this
      station. The unit of this attribute is dBm."
   DEFVAL { 0 }
   ::= { dot11SpectrumManagementEntry 4 }

dot11PowerCapabilityMinImplemented OBJECT-TYPE
   SYNTAX Integer32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      This attribute indicates the minimum transmit Power Capability of this
      station. The unit of this attribute is dBm."
   DEFVAL { -100 }
   ::= { dot11SpectrumManagementEntry 5 }
-- ******************************************************************
-- * End of dot11SpectrumManagement TABLE
-- ******************************************************************


-- ********************************************************************
-- * dot11RSNAConfig TABLE (RSNA and TSN)
-- ********************************************************************

dot11RSNAConfigTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11RSNAConfigEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "The table containing RSNA configuration objects."
   ::= { dot11smt 9 }

dot11RSNAConfigEntry OBJECT-TYPE
   SYNTAX Dot11RSNAConfigEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "An entry in the dot11RSNAConfigTable."
   INDEX { ifIndex }
   ::= { dot11RSNAConfigTable 1 }

Dot11RSNAConfigEntry ::=
   SEQUENCE {
      dot11RSNAConfigVersion                                 Unsigned32,
      dot11RSNAConfigPairwiseKeysImplemented                 Unsigned32,
      dot11RSNAConfigGroupCipher                             OCTET STRING,
      dot11RSNAConfigGroupRekeyMethod                        INTEGER,
      dot11RSNAConfigGroupRekeyTime                          Unsigned32,
      dot11RSNAConfigGroupRekeyPackets                       Unsigned32,
      dot11RSNAConfigGroupRekeyStrict                        TruthValue,
      dot11RSNAConfigPSKValue                                OCTET STRING,
      dot11RSNAConfigPSKPassPhrase                           DisplayString,
      dot11RSNAConfigGroupUpdateCount                        Unsigned32,
      dot11RSNAConfigPairwiseUpdateCount                     Unsigned32,
      dot11RSNAConfigGroupCipherSize                         Unsigned32,
      dot11RSNAConfigPMKLifetime                             Unsigned32,
      dot11RSNAConfigPMKReauthThreshold                      Unsigned32,
      dot11RSNAConfigNumberOfPTKSAReplayCountersImplemented
```

```
                                                            Unsigned32,
       dot11RSNAConfigSATimeout                             Unsigned32,
       dot11RSNAAuthenticationSuiteSelected                 OCTET STRING,
       dot11RSNAPairwiseCipherSelected                      OCTET STRING,
       dot11RSNAGroupCipherSelected                         OCTET STRING,
       dot11RSNAPMKIDUsed                                   OCTET STRING,
       dot11RSNAAuthenticationSuiteRequested                OCTET STRING,
       dot11RSNAPairwiseCipherRequested                     OCTET STRING,
       dot11RSNAGroupCipherRequested                        OCTET STRING,
       dot11RSNATKIPCounterMeasuresInvoked                  Unsigned32,
       dot11RSNA4WayHandshakeFailures                       Unsigned32,
       dot11RSNAConfigNumberOfGTKSAReplayCountersImplemented
                                                            Unsigned32,
       dot11RSNAConfigSTKKeysImplemented                    Unsigned32,
       dot11RSNAConfigSTKCipher                             OCTET STRING,
       dot11RSNAConfigSTKRekeyTime                          Unsigned32,
       dot11RSNAConfigSMKUpdateCount                        Unsigned32,
       dot11RSNAConfigSTKCipherSize                         Unsigned32,
       dot11RSNAConfigSMKLifetime                           Unsigned32,
       dot11RSNAConfigSMKReauthThreshold                    Unsigned32,
       dot11RSNAConfigNumberOfSTKSAReplayCountersImplemented
                                                            Unsigned32,
       dot11RSNAPairwiseSTKSelected                         OCTET STRING,
       dot11RSNASMKHandshakeFailures                        Unsigned32,
       dot11RSNASAERetransPeriod                            Unsigned32,
       dot11RSNASAEAntiCloggingThreshold                    Unsigned32,
       dot11RSNASAESync                                     Unsigned32 }

-- dot11RSNAConfigEntry 1 has been deprecated.

dot11RSNAConfigVersion OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      The highest RSNA version this entity supports. See 8.4.2.10."
   ::= { dot11RSNAConfigEntry 2 }

dot11RSNAConfigPairwiseKeysImplemented OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      This object indicates how many pairwise keys the entity supports for
      RSNA."
   ::= { dot11RSNAConfigEntry 3 }

dot11RSNAConfigGroupCipher OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(4))
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This object indicates the group cipher suite selector the entity must use.
      The group cipher suite in the RSN  element takes its value from this vari-
```

```
        able. It consists of an OUI (the first 3 octets) and a cipher suite iden-
        tifier (the last octet)."
    ::= { dot11RSNAConfigEntry 4 }

dot11RSNAConfigGroupRekeyMethod OBJECT-TYPE
SYNTAX INTEGER { disabled(1), timeBased(2), packetBased(3),
    timepacketBased(4) }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This object selects a mechanism for rekeying the RSNA GTK. The default is
        time-based, once per day. Rekeying the GTK is only applicable to an entity
        acting in the Authenticator role (an AP in an ESS)."
    DEFVAL { timeBased }
    ::= { dot11RSNAConfigEntry 5 }

dot11RSNAConfigGroupRekeyTime OBJECT-TYPE
    SYNTAX Unsigned32 (1..4294967295)
    UNITS "seconds"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        The time in seconds after which the RSNA GTK is refreshed. The timer
        starts at the moment the GTK was set using the MLME-SETKEYS.request prim-
        itive."
    DEFVAL { 86400 } -- once per day
    ::= { dot11RSNAConfigEntry 6 }

dot11RSNAConfigGroupRekeyPackets OBJECT-TYPE
    SYNTAX Unsigned32 (1..4294967295)
    UNITS "1000 packets"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        A packet count (in 1000s of packets) after which the RSNA GTK is
        refreshed. The packet counter starts at the moment the GTK was set using
        the MLME-SETKEYS.request primitive and it counts all packets encrypted
        using the current GTK."
    ::= { dot11RSNAConfigEntry 7 }

dot11RSNAConfigGroupRekeyStrict OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This object signals that the GTK be refreshed whenever a STA leaves the
        BSS that possesses the GTK."
    ::= { dot11RSNAConfigEntry 8 }
```

```
dot11RSNAConfigPSKValue OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(32))
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      The PSK for when RSNA in PSK mode is the selected AKM suite. In that case,
      the PMK obtains its value from this object.

      This object is logically write-only. Reading this variable returns unsuc-
      cessful status or null or 0."
   ::= { dot11RSNAConfigEntry 9 }

dot11RSNAConfigPSKPassPhrase OBJECT-TYPE
   SYNTAX DisplayString
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      The PSK, for when RSNA in PSK mode is the selected AKM suite, is config-
      ured by dot11RSNAConfigPSKValue.

      An alternative manner of setting the PSK uses the password-to-key algo-
      rithm defined in M.4. This variable provides a means to enter a pass-
      phrase. When this object is written, the RSNA entity uses the password-to-
      key algorithm specified in M.4 to derive a preshared and populate
      dot11RSNAConfigPSKValue with this key.
      This object is logically write-only. Reading this variable returns unsuc-
      cessful status or null or 0."
   ::= { dot11RSNAConfigEntry 10 }

-- dot11RSNAConfigEntry 11 and dot11RSNAConfigEntry 12 have been
-- deprecated.

dot11RSNAConfigGroupUpdateCount OBJECT-TYPE
   SYNTAX Unsigned32 (1..4294967295)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      The number of times Message 1 in the RSNA Group Key Handshake is retried
      per GTK Handshake attempt."
   DEFVAL { 3 }
   ::= { dot11RSNAConfigEntry 13 }

dot11RSNAConfigPairwiseUpdateCount OBJECT-TYPE
   SYNTAX Unsigned32 (1..4294967295)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.
```

```
        The number of times Message 1 and Message 3 in the RSNA 4-Way Handshake is
        retried per 4-Way Handshake attempt."
    DEFVAL { 3 }
    ::= { dot11RSNAConfigEntry 14 }

dot11RSNAConfigGroupCipherSize OBJECT-TYPE
    SYNTAX Unsigned32 (0..4294967295)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This object indicates the length in bits of the group cipher key."
    ::= { dot11RSNAConfigEntry 15 }

dot11RSNAConfigPMKLifetime OBJECT-TYPE
    SYNTAX Unsigned32 (1..4294967295)
    UNITS "seconds"
    MAX-ACCESSread-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        The maximum lifetime of a PMK in the PMK cache."
    DEFVAL { 43200 }
    ::= { dot11RSNAConfigEntry 16 }

dot11RSNAConfigPMKReauthThreshold OBJECT-TYPE
    SYNTAX Unsigned32 (1..100)
    UNITS "percentage"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        The percentage of the PMK lifetime that should expire before an IEEE
        802.1X reauthentication occurs."
    DEFVAL { 70 }
    ::= { dot11RSNAConfigEntry 17 }

dot11RSNAConfigNumberOfPTKSAReplayCountersImplemented OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        Specifies the number of PTKSA replay counters per association:
        0 -> 1 replay counter,
        1 -> 2 replay counters,
        2 -> 4 replay counters,
        3 -> 16 replay counters"
    DEFVAL { 3 }
    ::= { dot11RSNAConfigEntry 18 }

dot11RSNAConfigSATimeout OBJECT-TYPE
    SYNTAX Unsigned32 (1..4294967295)
    UNITS "seconds"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        The maximum time a security association takes to set up."
    DEFVAL { 60 }
    ::= { dot11RSNAConfigEntry 19 }

dot11RSNAAuthenticationSuiteSelected OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(4))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the RSNA Key Management in the SME when a security asso-
        ciation is established.

        The selector of the last AKM suite negotiated."
    ::= { dot11RSNAConfigEntry 20 }

dot11RSNAPairwiseCipherSelected OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(4))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the RSNA Key Management in the SME when a security asso-
        ciation is established.

        The selector of the last pairwise cipher negotiated."
    ::= { dot11RSNAConfigEntry 21 }

dot11RSNAGroupCipherSelected OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(4))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the RSNA Key Management in the SME when a security asso-
        ciation is established.

        The selector of the last group cipher negotiated."
    ::= { dot11RSNAConfigEntry 22 }

dot11RSNAPMKIDUsed OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(16))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the RSNA Key Management in the SME when a security asso-
        ciation is established.

        The selector of the last PMKID used in the last 4-Way Handshake."
    ::= { dot11RSNAConfigEntry 23 }

dot11RSNAAuthenticationSuiteRequested OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(4))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the RSNA Key Management in the SME when a security asso-
        ciation is established.

        The selector of the last AKM suite requested."
    ::= { dot11RSNAConfigEntry 24 }

dot11RSNAPairwiseCipherRequested OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(4))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the RSNA Key Management in the SME when a security asso-
        ciation is established.

        The selector of the last pairwise cipher requested."
    ::= { dot11RSNAConfigEntry 25 }

dot11RSNAGroupCipherRequested OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(4))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the RSNA Key Management in the SME when a security asso-
        ciation is established.

        The selector of the last group cipher requested."
    ::= { dot11RSNAConfigEntry 26 }

dot11RSNATKIPCounterMeasuresInvoked OBJECT-TYPE
    SYNTAX Unsigned32 (0..4294967295)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when the condition described below occurs.

        Counts the number of times that a TKIP MIC failure occurred two times
        within 60 s and TKIP countermeasures were invoked. This attribute counts
        both local and remote MIC failure events reported to this STA. It incre-
        ments every time TKIP countermeasures are invoked"
    ::= { dot11RSNAConfigEntry 27 }

dot11RSNA4WayHandshakeFailures OBJECT-TYPE
    SYNTAX Unsigned32 (0..4294967295)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the RSNA Key Management in the SME when the condition
        described below occurs.

        Counts the number of 4-Way Handshake failures."
    ::= { dot11RSNAConfigEntry 28 }

dot11RSNAConfigNumberOfGTKSAReplayCountersImplemented OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        Specifies the number of GTKSA replay counters per association:
        0 -> 1 replay counter,
        1 -> 2 replay counters,
        2 -> 4 replay counters,
        3 -> 16 replay counters"
    DEFVAL { 3 }
    ::= { dot11RSNAConfigEntry 29 }

dot11RSNAConfigSTKKeysImplemented OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This object indicates how many STK keys the entity supports for RSNA."
    ::= { dot11RSNAConfigEntry 30 }

dot11RSNAConfigSTKCipher OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(4))
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This object specifies the ciphersuite used by the STK for a DLS link."
    ::= { dot11RSNAConfigEntry 31}

dot11RSNAConfigSTKRekeyTime OBJECT-TYPE
    SYNTAX Unsigned32 (1..4294967295)
    UNITS "seconds"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        The time in seconds after which an RSNA STK is refreshed. The timer starts
        at the moment the STK was set using the MLME-SETKEYS.request primitive."
    DEFVAL { 86400 } -- once per day
    ::= { dot11RSNAConfigEntry 32 }

dot11RSNAConfigSMKUpdateCount OBJECT-TYPE
    SYNTAX Unsigned32 (1..4294967295)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        The number of times Message 1 in the RSNA SMK Handshake is retried per SMK
        Handshake attempt."
    DEFVAL { 3 }
    ::= { dot11RSNAConfigEntry 33 }

dot11RSNAConfigSTKCipherSize OBJECT-TYPE
    SYNTAX Unsigned32 (0..4294967295)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This object indicates the length in bits of the STK cipher key."
    ::= { dot11RSNAConfigEntry 34 }

dot11RSNAConfigSMKLifetime OBJECT-TYPE
    SYNTAX Unsigned32 (1..4294967295)
    UNITS "seconds"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        The maximum lifetime of an SMK in the SMK cache."
    DEFVAL { 43200 }
    ::= { dot11RSNAConfigEntry 35 }

dot11RSNAConfigSMKReauthThreshold OBJECT-TYPE
    SYNTAX Unsigned32 (0..4294967295)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute indicates the number of seconds for which an SMK authenti-
        cation is valid. A new SMK authentication must be completed successfully
        before the number of seconds indicated by this attribute elapse, from the
        prior authentication, before the STAs become unauthenticated."
    ::= { dot11RSNAConfigEntry 36 }

dot11RSNAConfigNumberOfSTKSAReplayCountersImplemented OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        Specifies the number of STKSA replay counters per association:
        0 -> 1 replay counter,
        1 -> 2 replay counters,
        2 -> 4 replay counters,
        3 -> 16 replay counters"
    DEFVAL { 3 }
    ::= { dot11RSNAConfigEntry 37 }

dot11RSNAPairwiseSTKSelected OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(4))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the RSNA Key Management in the SME when a security asso-
        ciation is established.
```

```
      The selector of the last STK cipher negotiated."
   ::= { dot11RSNAConfigEntry 38 }

dot11RSNASMKHandshakeFailures OBJECT-TYPE
   SYNTAX Unsigned32 (0..4294967295)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the RSNA Key Management in the SME when the condition
      described below occurs.

      Counts the number of SMK Handshake failures."
   ::= { dot11RSNAConfigEntry 39 }

dot11RSNASAERetransPeriod OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by the SME when establishing or becoming a member of a BSS.
      Changes take effect for the next MLME-START.request.

      This object specifies the initial retry timeout, in millisecond units,
      used by the SAE authentication and key establishment protocol."
   DEFVAL { 40 }
   ::= { dot11RSNAConfigEntry 40 }

dot11RSNASAEAntiCloggingThreshold OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      This object specifies the maximum number of SAE protocol instances allowed
      to simultaneously be in either Commit or Confirmed state."
   DEFVAL { 5 }
   ::= { dot11RSNAConfigEntry 41 }

dot11RSNASAESync OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      This object specifies the maximum number of synchronization errors that
      are allowed to happen prior to disassociation of the offending SAE peer."
   DEFVAL { 5 }
   ::= { dot11RSNAConfigEntry 42 }

-- **********************************************************************
-- * End of dot11RSNAConfig TABLE
-- **********************************************************************

-- **********************************************************************
-- * dot11RSNAConfigPairwiseCiphers TABLE
-- **********************************************************************

dot11RSNAConfigPairwiseCiphersTable OBJECT-TYPE
```

```
    SYNTAX SEQUENCE OF Dot11RSNAConfigPairwiseCiphersEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "This table lists the pairwise ciphers supported by this entity. It allows
        enabling and disabling of each pairwise cipher by network management. The
        pairwise cipher suite list in the RSNE is formed using the information in
        this table."
    ::= { dot11smt 10 }

dot11RSNAConfigPairwiseCiphersEntry OBJECT-TYPE
    SYNTAX Dot11RSNAConfigPairwiseCiphersEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "The table entry, indexed by the interface index (or all interfaces) and
        the pairwise cipher."
    INDEX { ifIndex, dot11RSNAConfigPairwiseCipherIndex }
    ::= { dot11RSNAConfigPairwiseCiphersTable 1 }

Dot11RSNAConfigPairwiseCiphersEntry ::=
    SEQUENCE {
        dot11RSNAConfigPairwiseCipherIndex                      Unsigned32,
        dot11RSNAConfigPairwiseCipherImplemented                OCTET STRING,
        dot11RSNAConfigPairwiseCipherActivated                  TruthValue,
        dot11RSNAConfigPairwiseCipherSizeImplemented            Unsigned32  }

dot11RSNAConfigPairwiseCipherIndex OBJECT-TYPE
    SYNTAX Unsigned32 (1..4294967295)
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "The auxiliary index into the dot11RSNAConfigPairwiseCiphersTable."
    ::= { dot11RSNAConfigPairwiseCiphersEntry 1 }

dot11RSNAConfigPairwiseCipherImplemented OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(4))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        The selector of a supported pairwise cipher. It consists of an OUI (the
        first 3 octets) and a cipher suite identifier (the last octet)."
    ::= { dot11RSNAConfigPairwiseCiphersEntry 2 }

dot11RSNAConfigPairwiseCipherActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This object enables or disables the pairwise cipher."
    ::= { dot11RSNAConfigPairwiseCiphersEntry 3 }

dot11RSNAConfigPairwiseCipherSizeImplemented OBJECT-TYPE
    SYNTAX Unsigned32 (0..4294967295)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
```

```
      "This is a capability variable.
      Its value is determined by device capabilities.

      This object indicates the length in bits of the pairwise cipher key. This
      should be 256 for TKIP and 128 for CCMP."
   ::= { dot11RSNAConfigPairwiseCiphersEntry 4 }

-- **********************************************************************
-- * End of dot11RSNAConfigPairwiseCiphers TABLE
-- **********************************************************************

-- **********************************************************************
-- * dot11RSNAConfigAuthenticationSuites TABLE
-- **********************************************************************

dot11RSNAConfigAuthenticationSuitesTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11RSNAConfigAuthenticationSuitesEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "This table lists the AKM suites supported by this entity. Each AKM suite
      can be individually enabled and disabled. The AKM suite list in the RSNE
      is formed using the information in this table."
   ::= { dot11smt 11 }

dot11RSNAConfigAuthenticationSuitesEntry OBJECT-TYPE
   SYNTAX Dot11RSNAConfigAuthenticationSuitesEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "An entry (row) in the dot11RSNAConfigAuthenticationSuitesTable."
   INDEX { dot11RSNAConfigAuthenticationSuiteIndex }
   ::= { dot11RSNAConfigAuthenticationSuitesTable 1 }

Dot11RSNAConfigAuthenticationSuitesEntry ::=
   SEQUENCE {
      dot11RSNAConfigAuthenticationSuiteIndex               Unsigned32,
      dot11RSNAConfigAuthenticationSuiteImplemented         OCTET STRING,
      dot11RSNAConfigAuthenticationSuiteActivated           TruthValue }

dot11RSNAConfigAuthenticationSuiteIndex OBJECT-TYPE
   SYNTAX Unsigned32 (1..4294967295)
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "The auxiliary variable used as an index into the
      dot11RSNAConfigAuthenticationSuitesTable."
   ::= { dot11RSNAConfigAuthenticationSuitesEntry 1 }

dot11RSNAConfigAuthenticationSuiteImplemented OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(4))
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      The selector of an AKM suite. It consists of an OUI (the first 3 octets)
      and a cipher suite identifier (the last octet)."
   ::= { dot11RSNAConfigAuthenticationSuitesEntry 2 }

dot11RSNAConfigAuthenticationSuiteActivated OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This variable indicates whether the corresponding AKM suite is enabled/
      disabled."
   ::= { dot11RSNAConfigAuthenticationSuitesEntry 3 }

-- ********************************************************************
-- * End of dot11RSNAConfigAuthenticationSuites TABLE
-- ********************************************************************

-- ********************************************************************
-- * dot11RSNAStats TABLE
-- ********************************************************************

dot11RSNAStatsTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11RSNAStatsEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "This table maintains per-STA statistics in an RSN. The entry with
      dot11RSNAStatsSTAAddress equal to FF-FF-FF-FF-FF-FF contains statistics
      for group addressed traffic."
   ::= { dot11smt 12 }

dot11RSNAStatsEntry OBJECT-TYPE
   SYNTAX Dot11RSNAStatsEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "An entry in the dot11RSNAStatsTable."
   INDEX { ifIndex, dot11RSNAStatsIndex }
   ::= { dot11RSNAStatsTable 1 }

Dot11RSNAStatsEntry ::=
   SEQUENCE {
      dot11RSNAStatsIndex                                    Unsigned32,
      dot11RSNAStatsSTAAddress                               MacAddress,
      dot11RSNAStatsVersion                                  Unsigned32,
      dot11RSNAStatsSelectedPairwiseCipher                   OCTET STRING,
      dot11RSNAStatsTKIPICVErrors                            Counter32,
      dot11RSNAStatsTKIPLocalMICFailures                     Counter32,
      dot11RSNAStatsTKIPRemoteMICFailures                    Counter32,
      dot11RSNAStatsCCMPReplays                              Counter32,
      dot11RSNAStatsCCMPDecryptErrors                        Counter32,
      dot11RSNAStatsTKIPReplays                              Counter32,
      dot11RSNAStatsCMACICVErrors                            Counter32,
      dot11RSNAStatsCMACReplays                              Counter32,
      dot11RSNAStatsRobustMgmtCCMPReplays                    Counter32,
      dot11RSNABIPMICErrors                                  Counter32  }

dot11RSNAStatsIndex OBJECT-TYPE
   SYNTAX Unsigned32 (1..4294967295)
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "An auxiliary index into the dot11RSNAStatsTable."
   ::= { dot11RSNAStatsEntry 1 }

dot11RSNAStatsSTAAddress OBJECT-TYPE
   SYNTAX MacAddress
```

```
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the RSNA Key Management in the SME.

        The MAC address of the STA to which the statistics in this conceptual row
        belong."
    ::= { dot11RSNAStatsEntry 2 }

dot11RSNAStatsVersion OBJECT-TYPE
    SYNTAX Unsigned32 (1..4294967295)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the RSNA Key Management in the SME.

        The RSNA version with which the STA associated."
    ::= { dot11RSNAStatsEntry 3 }

dot11RSNAStatsSelectedPairwiseCipher OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(4))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the RSNA Key Management in the SME.

        The pairwise cipher suite Selector (as defined in 8.4.2.27.2) used during
        association, in transmission order."
    ::= { dot11RSNAStatsEntry 4 }

dot11RSNAStatsTKIPICVErrors OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when the condition described below occurs.

        Counts the number of TKIP ICV errors encountered when decrypting packets
        for the STA."
    ::= { dot11RSNAStatsEntry 5 }

dot11RSNAStatsTKIPLocalMICFailures OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when the condition described below occurs.

        Counts the number of MIC failures encountered when checking the integrity
        of packets received from the STA at this entity."
    ::= { dot11RSNAStatsEntry 6 }

dot11RSNAStatsTKIPRemoteMICFailures OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when the condition described below occurs.
        Counts the number of MIC failures encountered by the STA identified by
        dot11StatsSTAAddress and reported back to this entity."
    ::= { dot11RSNAStatsEntry 7 }

dot11RSNAStatsCCMPReplays OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when the condition described below occurs.

        The number of received CCMP MPDUs discarded by the replay mechanism."
    ::= { dot11RSNAStatsEntry 8 }

dot11RSNAStatsCCMPDecryptErrors OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when the condition described below occurs.

        The number of received MPDUs discarded by the CCMP decryption algorithm."
    ::= { dot11RSNAStatsEntry 9 }

dot11RSNAStatsTKIPReplays OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when the condition described below occurs.

        Counts the number of TKIP replay errors detected."
    ::= { dot11RSNAStatsEntry 10 }

dot11RSNAStatsCMACICVErrors OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when the condition described below occurs.

        The number of received MPDUs discarded by the CMAC integrity check algo-
        rithm."
    ::= { dot11RSNAStatsEntry 11 }

dot11RSNAStatsCMACReplays OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when the condition described below occurs.

        The number of received MPDUs discarded by the CMAC replay errors."
    ::= { dot11RSNAStatsEntry 12 }

dot11RSNAStatsRobustMgmtCCMPReplays OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when the condition described below occurs.

        The number of received robust management frame MPDUs discarded due to CCMP
        replay errors"
    ::= {dot11RSNAStatsEntry 13}

dot11RSNABIPMICErrors OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when the condition described below occurs.

        The number of received MMPDUs discarded due to BIP MIC errors"
    ::= {dot11RSNAStatsEntry 14}

-- ********************************************************************
-- * End of dot11RSNAStats TABLE
-- ********************************************************************

-- **********************************************************************
-- * dot11OperatingClasses TABLE
-- **********************************************************************

dot11OperatingClassesTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11OperatingClassesEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "(Conceptual) table of attributes for operating classes"
    ::= {dot11smt 13}

dot11OperatingClassesEntry OBJECT-TYPE
    SYNTAX Dot11OperatingClassesEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry (conceptual row) in the Operating Classes Table.

        IfIndex - Each 802.11 interface is represented by an ifEntry. Interface
        tables in this MIB are indexed by ifIndex."
    INDEX {ifIndex, dot11OperatingClassesIndex}
    ::= { dot11OperatingClassesTable 1 }

Dot11OperatingClassesEntry ::=
    SEQUENCE {
        dot11OperatingClassesIndex                          Unsigned32,
        dot11OperatingClass                                 Unsigned32,
        dot11CoverageClass                                  Unsigned32 }

dot11OperatingClassesIndex OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "The auxiliary variable used to identify instances of the columnar objects
        in the Operating Classes Table."
    ::= { dot11OperatingClassesEntry 1 }

dot11OperatingClass OBJECT-TYPE
```

```
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by the SME or external management entity when the device is
        initialized.

        This attribute indicates the operating class to be used."
    DEFVAL { 0 }
    ::= { dot11OperatingClassesEntry 2 }

dot11CoverageClass OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by the SME or external management entity when the device is
        initialized.

        This attribute indicates the coverage class to be used."
    DEFVAL { 0 }
    ::= { dot11OperatingClassesEntry 3 }

-- **********************************************************************
-- * End of dot11OperatingClasses TABLE
-- **********************************************************************
-- **********************************************************************
-- * dot11FastBSSTransitionConfig TABLE
-- **********************************************************************
dot11FastBSSTransitionConfigTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11FastBSSTransitionConfigEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "The table containing fast BSS transition configuration objects."
   ::= { dot11smt 15 }

dot11FastBSSTransitionConfigEntry OBJECT-TYPE
   SYNTAX Dot11FastBSSTransitionConfigEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "An entry in the dot11FastBSSTransitionConfigTable."
   INDEX { ifIndex }
   ::= { dot11FastBSSTransitionConfigTable 1 }

Dot11FastBSSTransitionConfigEntry ::=
   SEQUENCE {
      dot11FastBSSTransitionActivated                 TruthValue,
      dot11FTMobilityDomainID                         OCTET STRING,
      dot11FTOverDSActivated                          TruthValue,
      dot11FTResourceRequestSupported                 TruthValue,
      dot11FTR0KeyHolderID                            OCTET STRING,
      dot11FTR0KeyLifetime                            Unsigned32,
      dot11FTR1KeyHolderID                            OCTET STRING,
      dot11FTReassociationDeadline                    Unsigned32 }

dot11FastBSSTransitionActivated OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      When this object is true, this indicates that fast BSS transition (FT) is
      enabled on this entity. The entity  advertises the FT-related elements in
      its Beacon and Probe Response frames. This object requires that
      dot11FastBSSTransitionImplemented also be equal to true."
   ::= { dot11FastBSSTransitionConfigEntry 1 }

dot11FTMobilityDomainID OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(2))
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute specifies the Mobility Domain identifier (MDID) of this
      entity.

      The MDID is used to indicate a group of APs, within an ESS, between which
      a STA can use fast BSS transition services. Fast BSS transitions are
      allowed only between APs that have the same MDID and are within the same
      ESS. They are not allowed between APs with different MDIDs or in different
      ESSs.
```

```
        Since fast BSS transition services are defined only within the scope of an
        ESS, there is no requirement that MDIDs be unique across ESSs."
    ::= { dot11FastBSSTransitionConfigEntry 2 }

dot11FTOverDSActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        When this object is true, this indicates that fast BSS transition via the
        over-the-DS protocol as described in Clause 12 is enabled on this AP
        entity."
    ::= { dot11FastBSSTransitionConfigEntry 3 }

dot11FTResourceRequestSupported OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        When this object is true, this indicates that the fast BSS transition (FT)
        resource request procedures of 12.10 are supported on this AP entity."
    ::= { dot11FastBSSTransitionConfigEntry 4 }

dot11FTR0KeyHolderID OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(1..48))
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute specifies the PMK-R0 key holder identifier (R0KH-ID) of the
        Authenticator of this AP.

        NOTE: Backend protocol may allow longer NAS Client identifiers (e.g.,
        RADIUS allows up to 253-octet NAS-Identifier), but when used with fast BSS
        transition, the maximum length is limited to 48 octets. The same value
        must be used for the NAS Client identifier and dot11FTR0KeyHolderID."
    ::= { dot11FastBSSTransitionConfigEntry 5 }

dot11FTR0KeyLifetime OBJECT-TYPE
    SYNTAX Unsigned32 (60..4294967295)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute specifies the default lifetime of the PMK-R0, in seconds,
        when a Session-Timeout attribute is not provided during the EAP authenti-
        cation. This attribute also applies when the PMK-R0 is derived from a
        PSK."
    DEFVAL { 1209600 }
    ::= { dot11FastBSSTransitionConfigEntry 6 }
```

```
dot11FTR1KeyHolderID OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(6))
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute specifies the PMK-R1 key holder identifier (R1KH-ID) of the
      Authenticator of this AP. It is equal to a MAC address of the entity hold-
      ing the PMK-R1 in the Authenticator."
   ::= { dot11FastBSSTransitionConfigEntry 7 }

dot11FTReassociationDeadline OBJECT-TYPE
   SYNTAX Unsigned32 (1000..4294967295)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute specifies the number of time units (TUs) that this target
      AP entity retains a PTKSA and reserves any specified resources for a STA
      while waiting for a reassociation from that STA. It is assumed that this
      value is administered consistently across the mobility domain."
   DEFVAL { 1000 }
   ::= { dot11FastBSSTransitionConfigEntry 8 }

-- **********************************************************************
-- * End of dot11FastBSSTransitionConfig TABLE
-- **********************************************************************

-- *********************************************************************
-- * dot11LCIDSE TABLE
-- *********************************************************************

dot11LCIDSETable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11LCIDSEEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "Group contains conceptual table of attributes for Dependent Station
      Enablement."
   ::= { dot11smt 16 }

dot11LCIDSEEntry OBJECT-TYPE
   SYNTAX Dot11LCIDSEEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "An entry in the dot11LCIDSETable Indexed by dot11LCIDSEIndex."
   INDEX { dot11LCIDSEIndex }
   ::= { dot11LCIDSETable 1 }

Dot11LCIDSEEntry ::=
   SEQUENCE {
      dot11LCIDSEIndex                              Unsigned32,
      dot11LCIDSEIfIndex                            InterfaceIndex,
      dot11LCIDSECurrentOperatingClass              Unsigned32,
      dot11LCIDSELatitudeResolution                 Unsigned32,
      dot11LCIDSELatitudeInteger                    Integer32,
      dot11LCIDSELatitudeFraction                        Integer32,
      dot11LCIDSELongitudeResolution                     Unsigned32,
      dot11LCIDSELongitudeInteger                        Integer32,
      dot11LCIDSELongitudeFraction                       Integer32,
      dot11LCIDSEAltitudeType                            INTEGER,
      dot11LCIDSEAltitudeResolution                      Unsigned32,
      dot11LCIDSEAltitudeInteger                         Integer32,
      dot11LCIDSEAltitudeFraction                        Integer32,
      dot11LCIDSEDatum                                   Unsigned32,
      dot11RegLocAgreement                               TruthValue,
      dot11RegLocDSE                                     TruthValue,
      dot11DependentSTA                                  TruthValue,
      dot11DependentEnablementIdentifier                 Unsigned32,
      dot11DSEEnablementTimeLimit                        Unsigned32,
      dot11DSEEnablementFailHoldTime                     Unsigned32,
      dot11DSERenewalTime                                Unsigned32,
      dot11DSETransmitDivisor                            Unsigned32 }

dot11LCIDSEIndex OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "Index for LCI DSE elements in dot11LCIDSETable, greater than 0."
   ::= { dot11LCIDSEEntry 1 }

dot11LCIDSEIfIndex OBJECT-TYPE
   SYNTAX InterfaceIndex
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "Each IEEE 802.11 interface is represented by an ifEntry. Interface Tables
      in this MIB are indexed by ifIndex."
   ::= { dot11LCIDSEEntry 2 }

dot11LCIDSECurrentOperatingClass OBJECT-TYPE
   SYNTAX Unsigned32 (0..255)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by the SME when the device is initialized.

      Current Operating Class is 8 bits indicating the particular Operating
      Class in use by the radio."
   ::= { dot11LCIDSEEntry 3 }

dot11LCIDSELatitudeResolution OBJECT-TYPE
   SYNTAX Unsigned32 (0..63)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by the SME when the device is initialized.

      Latitude resolution is 6 bits indicating the number of valid bits in the
      fixed-point value of Latitude. This field is derived from IETF RFC 3825."
   ::= { dot11LCIDSEEntry 4 }

dot11LCIDSELatitudeInteger OBJECT-TYPE
   SYNTAX Integer32 (-359..359)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
```

```
      "This is a control variable.
      It is written by the SME when the device is initialized.

      Latitude is a twos-complement 34-bit fixed point value consisting of 9
      bits of integer and 25 bits of fraction. This field contains the 9 bits of
      integer portion of Latitude. This field is derived from IETF RFC 3825."
   ::= { dot11LCIDSEEntry 5 }

dot11LCIDSELatitudeFraction OBJECT-TYPE
   SYNTAX Integer32 (-16777215..16777215)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by the SME when the device is initialized.

      Latitude is a twos-complement 34-bit fixed point value consisting of 9
      bits of integer and 25 bits of fraction. This field contains the 25 bits
      of fraction portion of Latitude. This field is derived from IETF RFC
      3825."
   ::= { dot11LCIDSEEntry 6 }

dot11LCIDSELongitudeResolution OBJECT-TYPE
   SYNTAX Unsigned32 (0..63)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by the SME when the device is initialized.

      Longitude resolution is 6 bits indicating the number of valid bits in the
      fixed-point value of Longitude. This field is derived from IETF RFC 3825."
   ::= { dot11LCIDSEEntry 7 }

dot11LCIDSELongitudeInteger OBJECT-TYPE
   SYNTAX Integer32 (-359..359)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by the SME when the device is initialized.

      Longitude is a twos-complement 34-bit fixed point value consisting of 9
      bits of integer and 25 bits of fraction. This field contains the 9 bits of
      integer portion of Longitude. This field is derived from IETF RFC 3825."
   ::= { dot11LCIDSEEntry 8 }

dot11LCIDSELongitudeFraction OBJECT-TYPE
   SYNTAX Integer32 (-16777215..16777215)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by the SME when the device is initialized.

      Longitude is a twos-complement 34-bit fixed point value consisting of 9
      bits of integer and 25 bits of fraction. This field contains the 25 bits
      of fraction portion of Longitude. This field is derived from IETF RFC
      3825."
   ::= { dot11LCIDSEEntry 9 }

dot11LCIDSEAltitudeType OBJECT-TYPE
   SYNTAX INTEGER { meters(1), floors(2), hagm(3) }
   MAX-ACCESS read-only
```

```
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by the SME when the device is initialized.

        Altitude Type is 4 bits encoding the type of altitude.
        Codes defined are:
        meters : in 2s-complement fixed-point 22-bit integer part with 8-bit frac-
        tion
        floors : in 2s-complement fixed-point 22-bit integer part with 8-bit frac-
        tion
        hagm : Height Above Ground in meters, in 2s-complement fixed-point 22-bit
        integer part with 8-bit fraction.

        This field is derived from IETF RFC 3825."
    DEFVAL { 3 }
    ::= { dot11LCIDSEEntry 10 }

dot11LCIDSEAltitudeResolution OBJECT-TYPE
    SYNTAX Unsigned32 (0..63)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by the SME when the device is initialized.

        Altitude resolution is 6 bits indicating the number of valid bits in the
        altitude. This field is derived from IETF RFC 3825."
    ::= { dot11LCIDSEEntry 11 }

dot11LCIDSEAltitudeInteger OBJECT-TYPE
    SYNTAX Integer32 (-2097151..2097151)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by the SME when the device is initialized.

        Altitude is a 30-bit value defined by the Altitude type field. The field
        is encoded as a 2s-complement fixed-point 22-bit integer Part with 8-bit
        fraction. This field contains the fixed-point Part of Altitude. This field
        is derived from IETF RFC 3825."
    ::= { dot11LCIDSEEntry 12 }

dot11LCIDSEAltitudeFraction OBJECT-TYPE
    SYNTAX Integer32 (-127..127)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by the SME when the device is initialized.

        Altitude is a 30-bit value defined by the Altitude type field. The field
        is encoded as a 2s-complement fixed-point 22-bit integer Part with 8-bit
        fraction. This field contains the fraction part of Altitude. This field is
        derived from IETF RFC 3825."
    ::= { dot11LCIDSEEntry 13 }

dot11LCIDSEDatum OBJECT-TYPE
    SYNTAX Unsigned32 (1..3)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by the SME when the device is initialized.

        Datum is an 8-bit value encoding the horizontal and vertical references
        used for the coordinates given in this LCI. IETF RFC 3825 defines the val-
        ues of Datum. Type 1 is WGS-84, the coordinate system used by GPS. Type 2
        is NAD83 with NAVD88 vertical reference. Type 3 is NAD83 with Mean Lower
        Low Water vertical datum. All other types are reserved. This field is
        derived from IETF RFC 3825."
    DEFVAL { 1 }
    ::= { dot11LCIDSEEntry 14 }

dot11RegLocAgreement OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when a DSE Enablement frame is received.

        RegLocAgreement reports the Enabling STA's Agreement status. False indi-
        cates it is operating away from national borders and outside national pol-
        icy zones. True indicates it is operating by agreement near national
        borders or inside national policy zones."
    DEFVAL { false }
    ::= { dot11LCIDSEEntry 15 }

dot11RegLocDSE OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when a DSE Enablement frame is received.

        RegLocDSE reports the Enabling STA's DSE status. False indicates Dependent
        STAs are not enabled. True indicates Dependent STA operation is enabled."
    DEFVAL { false }
    ::= { dot11LCIDSEEntry 16 }

dot11DependentSTA OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when a DSE Enablement frame is received.

        This attribute reports the Dependent STA status of the STA that sent the
        beacon or Probe Response with this information. False indicates that STA
        is not operating as a Dependent STA. True indicates that STA is operating
        as a Dependent STA."
    DEFVAL { true }
    ::= { dot11LCIDSEEntry 17 }

dot11DependentEnablementIdentifier OBJECT-TYPE
    SYNTAX Unsigned32 (0..65535)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when a DSE Enablement frame is received.

        This attribute reports the Dependent STA identifier assigned by the
        enabling STA to the dependent station."
    DEFVAL { 0 }
    ::= { dot11LCIDSEEntry 18 }

dot11DSEEnablementTimeLimit OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by the SME when the device is initialized.

        dot11DSEAssociateTimeLimit indicates the maximum number of seconds that a
        dependent STA may transmit in a DSE frequency band while attaining enable-
        ment with an enabling STA. Unless another value is mandated by regulatory
        authorities, the value is 32 seconds."
    DEFVAL { 32 }
    ::= { dot11LCIDSEEntry 19 }

dot11DSEEnablementFailHoldTime OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by the SME when the device is initialized.

        dot11DSEAssociateFailHoldTime indicates the number of seconds that a
        dependent STA must not transmit in a DSE frequency band when it fails to
        attain enablement with an enabling STA within dot11DSEEnablementTimeLimit
        seconds. Unless another value is mandated by regulatory authorities, the
        value is 512 seconds."
    DEFVAL { 512 }
    ::= { dot11LCIDSEEntry 20}

dot11DSERenewalTime OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by the SME when the device is initialized.

        dot11DSERenewalTime indicates the maximum number of seconds that a depen-
        dent STA may operate in a DSE frequency band without receiving and decod-
        ing an enabling signal from its enabling STA. Unless another value is
        mandated by regulatory authorities, the value is 60 seconds."
    DEFVAL { 60 }
    ::= { dot11LCIDSEEntry 21}

dot11DSETransmitDivisor OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by the SME when the device is initialized.

        dot11DSETransmitDivisor indicates the value used by a dependent STA when
        operating in a DSE frequency band and transmitting. The dependent STA
        sends an Action frame when the sum of dot11TransmittedFragmentCount,
        dot11GroupTransmittedFrameCount and dot11ReceivedFragmentCount modulo
        dot11DSETransmitDivisor equals 0. Unless another value is mandated by reg-
        ulatory authorities, the default value is 256."
    DEFVAL { 256 }
    ::= { dot11LCIDSEEntry 22}

-- **********************************************************************
-- * End of dot11LCIDSE TABLE
-- **********************************************************************

-- ************************************************************************
-- * dot11HTStationConfig TABLE
-- ************************************************************************
dot11HTStationConfigTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11HTStationConfigEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Station Configuration attributes. In tabular form to allow for multiple
        instances on an agent."
    ::= { dot11smt 17 }

dot11HTStationConfigEntry OBJECT-TYPE
    SYNTAX Dot11HTStationConfigEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry (conceptual row) in the dot11HTStationConfig Table.

        ifIndex - Each IEEE 802.11 interface is represented by an ifEntry. Inter-
        face tables in this MIB module are indexed by ifIndex."
    INDEX { ifIndex }
    ::= { dot11HTStationConfigTable 1 }

Dot11HTStationConfigEntry ::=
    SEQUENCE {
        dot11HTOperationalMCSSet                        OCTET STRING,
        dot11MIMOPowerSave                              INTEGER,
        dot11NDelayedBlockAckOptionImplemented          TruthValue,
        dot11MaxAMSDULength                             INTEGER,
        dot11STBCControlFrameOptionImplemented          TruthValue,
        dot11LsigTxopProtectionOptionImplemented        TruthValue,
        dot11MaxRxAMPDUFactor                           Unsigned32,
        dot11MinimumMPDUStartSpacing                    Unsigned32,
        dot11PCOOptionImplemented                       TruthValue,
        dot11TransitionTime                             Unsigned32,
        dot11MCSFeedbackOptionImplemented               INTEGER,
        dot11HTControlFieldSupported                    TruthValue,
        dot11RDResponderOptionImplemented               TruthValue,
        dot11SPPAMSDUCapable                            TruthValue,
        dot11SPPAMSDURequired                           TruthValue,
        dot11FortyMHzOptionImplemented                  TruthValue }

dot11HTOperationalMCSSet OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(1..127))
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect for the next MLME-START.request primitive or MLME-
        JOIN.request primitive.

        This attribute shall specify the set of MCS at which the station may
        transmit data. Each octet contains a value representing a rate. Each MCS
        shall be within the range from 1 to 127, and shall be supported for
        receiving data. This value is reported in transmitted Beacon, Probe
        Request, Probe Response, Association Request, Association Response, Reas-
```

```
      sociation Request, and Reassociation Response frames, and is used to
      determine whether a BSS with which the station desires to synchronize is
      suitable. It is also used when starting a BSS, as specified in 10.3."
   ::= { dot11HTStationConfigEntry 1 }

dot11MIMOPowerSave OBJECT-TYPE
   SYNTAX INTEGER { static(1), dynamic(2), mimo(3) }
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the PHY when the power save condition changes.

      This is an 8-bit integer value that identifies the configured power save
      state of MIMO."
   ::= { dot11HTStationConfigEntry 2 }

dot11NDelayedBlockAckOptionImplemented OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      This attribute, when true, indicates that the station implementation is
      capable of supporting the No Ack option of the Delayed Block Ack."
   DEFVAL { false }
   ::= { dot11HTStationConfigEntry 3 }

dot11MaxAMSDULength OBJECT-TYPE
   SYNTAX INTEGER { short(3839), long(7935) }
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      This attribute indicates the supported maximum size of A-MSDU."
   DEFVAL { short }
   ::= { dot11HTStationConfigEntry 4 }

dot11STBCControlFrameOptionImplemented OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      This attribute, when true, indicates that the station implementation is
      capable of processing the received control frames that are STBC frames."
   DEFVAL { false }
   ::= { dot11HTStationConfigEntry 5 }

dot11LsigTxopProtectionOptionImplemented OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      This attribute, when true, indicates that the station implementation is
```

```
        capable of supporting L-SIG TXOP protection option."
    DEFVAL { false }
    ::= { dot11HTStationConfigEntry 6 }

dot11MaxRxAMPDUFactor OBJECT-TYPE
    SYNTAX Unsigned32 (0..3)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute indicates the maximum length of A-MPDU that the STA can
        receive. The Maximum Rx A-MPDU defined by this field is equal to 2 **
        (13+dot11MaxRxAMPDUFactor) -1 octets."
    DEFVAL { 0 }
    ::= { dot11HTStationConfigEntry 7 }

dot11MinimumMPDUStartSpacing OBJECT-TYPE
    SYNTAX Unsigned32 (0..7)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute indicates the minimum time between the start of adjacent
        MPDUs within an A-MPDU. This time is measured at the PHY-SAP; the number
        of octets between the start of two consecutive MPDUs in A-MPDU shall be
        equal or greater than (dot11MinimumMPDUStartSpacing*PHY-bit-rate)/8. The
        encoding of the minimum time to this attribute is:
        0 - no restriction
        1 - 1/4 microsecond
        2 - 1/2 microsecond
        3 - 1 microsecond
        4 - 2 microseconds
        5 - 4 microseconds
        6 - 8 microseconds
        7 - 16 microseconds"
    DEFVAL { 0 }
    ::= { dot11HTStationConfigEntry 8 }

dot11PCOOptionImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute, when true, indicates that the station implementation is
        capable of supporting PCO."
    DEFVAL { false }
    ::= { dot11HTStationConfigEntry 9 }

dot11TransitionTime OBJECT-TYPE
    SYNTAX Unsigned32 (0..3)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute indicates that the maximum transition time within which the
        STA can switch between 20 MHz channel width and 40 MHz channel width with
        a high probability. The encoding of the transition time to this attribute
        is:
        0 - no transition
        1 - 400 microsecondss
        2 - 1500 microseconds
        3 - 5000 microseconds"
    DEFVAL { 2 }
    ::= { dot11HTStationConfigEntry 10 }

dot11MCSFeedbackOptionImplemented OBJECT-TYPE
    SYNTAX INTEGER { none(0), unsolicited (2), both (3) }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute indicates the MCS feed back capability supported by the
        station implementation."
    DEFVAL { 0 }
    ::= { dot11HTStationConfigEntry 11 }

dot11HTControlFieldSupported OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute, when true, indicates that the station implementation is
        capable of receiving HT Control field."
    DEFVAL { false }
    ::= { dot11HTStationConfigEntry 12 }

dot11RDResponderOptionImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute, when true, indicates that the station implementation is
        capable operating as an RD responder."
    DEFVAL { false }
    ::= { dot11HTStationConfigEntry 13 }

dot11SPPAMSDUCapable OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute, when true, indicates that the STA implementation is capa-
        ble of protecting the A-MSDU bit (Bit 7) in the QoS Control field when
        dot11RSNAActivated is true."
    DEFVAL { false }
    ::= { dot11HTStationConfigEntry 14 }

dot11SPPAMSDURequired OBJECT-TYPE
```

```
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute, when true, indicates that the STA is configured to disal-
        low (i.e., not to send or receive) PP A-MSDUs when dot11RSNAActivated is
        true."
    DEFVAL { false }
    ::= { dot11HTStationConfigEntry 15 }

dot11FortyMHzOptionImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute, when true, indicates that the STA is capable of transmit-
        ting and receiving on a 40 MHz channel using a 40 MHz mask."
    DEFVAL { false }
    ::= { dot11HTStationConfigEntry 16 }

-- **********************************************************************
-- * End of dot11HTStationConfig TABLE
-- **********************************************************************

-- **********************************************************************
-- * dot11WirelessMgmtOptions TABLE
-- **********************************************************************
dot11WirelessMgmtOptionsTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11WirelessMgmtOptionsEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Wireless Management attributes. In tabular form to allow for multiple
        instances on an agent. This table only applies to the interface if
        dot11WirelessManagementImplemented is set to true in the
        dot11StationConfigTable. Otherwise this table should be ignored."
    ::= { dot11smt 18 }

dot11WirelessMgmtOptionsEntry OBJECT-TYPE
    SYNTAX Dot11WirelessMgmtOptionsEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry in the dot11WirelessMgmtOptionsTable. For all Wireless Manage-
        ment
        features, an Activated MIB variable is used to activate/enable or deacti-
        vate/disable the corresponding feature. An Implemented MIB variable is
        used for an optional feature to indicate whether the feature is imple-
        mented. A mandatory feature does not have a corresponding Implemented MIB
        variable. It is possible for there to be multiple IEEE 802.11 interfaces
        on one agent, each with its unique MAC address. The relationship between
        an IEEE 802.11 interface and an interface in the context of the Internet-
        standard MIB is one-to-one. As such, the value of an ifIndex object
        instance can be directly used to identify corresponding instances of the
        objects defined herein. ifIndex - Each IEEE 802.11 interface is repre-
        sented by an ifEntry. Interface tables in this MIB module are indexed by
        ifIndex."
    INDEX { ifIndex }
    ::= { dot11WirelessMgmtOptionsTable 1 }

Dot11WirelessMgmtOptionsEntry ::=
    SEQUENCE {
        dot11MgmtOptionLocationActivated                      TruthValue,
        dot11MgmtOptionFMSImplemented                         TruthValue,
        dot11MgmtOptionFMSActivated                           TruthValue,
        dot11MgmtOptionEventsActivated                        TruthValue,
        dot11MgmtOptionDiagnosticsActivated                   TruthValue,
        dot11MgmtOptionMultiBSSIDImplemented                  TruthValue,
        dot11MgmtOptionMultiBSSIDActivated                    TruthValue,
        dot11MgmtOptionTFSImplemented                         TruthValue,
        dot11MgmtOptionTFSActivated                           TruthValue,
        dot11MgmtOptionWNMSleepModeImplemented                TruthValue,
        dot11MgmtOptionWNMSleepModeActivated                  TruthValue,
        dot11MgmtOptionTIMBroadcastImplemented                TruthValue,
        dot11MgmtOptionTIMBroadcastActivated                  TruthValue,
        dot11MgmtOptionProxyARPImplemented                    TruthValue,
        dot11MgmtOptionProxyARPActivated                      TruthValue,
        dot11MgmtOptionBSSTransitionImplemented               TruthValue,
        dot11MgmtOptionBSSTransitionActivated                 TruthValue,
        dot11MgmtOptionQoSTrafficCapabilityImplemented        TruthValue,
        dot11MgmtOptionQoSTrafficCapabilityActivated          TruthValue,
        dot11MgmtOptionACStationCountImplemented              TruthValue,
        dot11MgmtOptionACStationCountActivated                TruthValue,
        dot11MgmtOptionCoLocIntfReportingImplemented          TruthValue,
        dot11MgmtOptionCoLocIntfReportingActivated            TruthValue,
        dot11MgmtOptionMotionDetectionImplemented             TruthValue,
        dot11MgmtOptionMotionDetectionActivated               TruthValue,
        dot11MgmtOptionTODImplemented                         TruthValue,
        dot11MgmtOptionTODActivated                           TruthValue,
        dot11MgmtOptionTimingMsmtImplemented                  TruthValue,
        dot11MgmtOptionTimingMsmtActivated                    TruthValue,
        dot11MgmtOptionChannelUsageImplemented                TruthValue,
        dot11MgmtOptionChannelUsageActivated                  TruthValue,
        dot11MgmtOptionTriggerSTAStatisticsActivated          TruthValue,
        dot11MgmtOptionSSIDListImplemented                    TruthValue,
        dot11MgmtOptionSSIDListActivated                      TruthValue,
        dot11MgmtOptionMulticastDiagnosticsActivated          TruthValue,
        dot11MgmtOptionLocationTrackingImplemented            TruthValue,
        dot11MgmtOptionLocationTrackingActivated              TruthValue,
        dot11MgmtOptionDMSImplemented                         TruthValue,
        dot11MgmtOptionDMSActivated                           TruthValue,
        dot11MgmtOptionUAPSDCoexistenceImplemented            TruthValue,
        dot11MgmtOptionUAPSDCoexistenceActivated              TruthValue,
        dot11MgmtOptionWNMNotificationImplemented             TruthValue,
        dot11MgmtOptionWNMNotificationActivated               TruthValue,
        dot11MgmtOptionUTCTSFOffsetImplemented                TruthValue,
        dot11MgmtOptionUTCTSFOffsetActivated                  TruthValue
    }

dot11MgmtOptionLocationActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        This attribute, when true, indicates that the capability of the station to
        provide location is enabled. The capability is disabled, otherwise."
    DEFVAL { false}
    ::= { dot11WirelessMgmtOptionsEntry 1 }

dot11MgmtOptionFMSImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute, when true, indicates that the station implementation is
        capable of supporting FMS when the dot11WirelessManagementImplemented is
        set to true."
    ::= { dot11WirelessMgmtOptionsEntry 2 }

dot11MgmtOptionFMSActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        This attribute, when true, indicates that the capability of the station to
        provide FMS is enabled. The capability is disabled, otherwise"
    DEFVAL { false}
    ::= { dot11WirelessMgmtOptionsEntry 3 }

dot11MgmtOptionEventsActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        This attribute, when true, indicates that the capability of the station to
        provide Event Reporting is enabled. The capability is disabled, otherwise"
    DEFVAL { false}
    ::= { dot11WirelessMgmtOptionsEntry 4 }

dot11MgmtOptionDiagnosticsActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        This attribute, when true, indicates that the capability of the station to
        provide Diagnostic Reporting is enabled. The capability is disabled, oth-
        erwise."
    DEFVAL { false}
    ::= { dot11WirelessMgmtOptionsEntry 5 }

dot11MgmtOptionMultiBSSIDImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute, when true, indicates that the station implementation is
        capable of supporting Multiple BSSID when the
        dot11WirelessManagementImplemented is set to true."
    ::= { dot11WirelessMgmtOptionsEntry 6 }

dot11MgmtOptionMultiBSSIDActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        This attribute, when true, indicates that the capability of the station to
        provide Multi BSSID is enabled. The capability is disabled, otherwise."
    DEFVAL { false}
    ::= { dot11WirelessMgmtOptionsEntry 7 }

dot11MgmtOptionTFSImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute, when true, indicates that the station implementation is
        capable of supporting TFS when the dot11WirelessManagementImplemented is
        set to true."
    ::= { dot11WirelessMgmtOptionsEntry 8 }

dot11MgmtOptionTFSActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        This attribute, when true, indicates that TFS is enabled. TFS is disabled
        otherwise."
    DEFVAL { false}
    ::= { dot11WirelessMgmtOptionsEntry 9 }

dot11MgmtOptionWNMSleepModeImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute, when true, indicates that the station implementation is
        capable of supporting WNMSleep Mode when the
        dot11WirelessManagementImplemented is set to true."
    ::= { dot11WirelessMgmtOptionsEntry 10 }

dot11MgmtOptionWNMSleepModeActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        This attribute, when true, indicates that WNMSleep Mode is enabled.
        WNMSleep Mode is disabled otherwise."
    DEFVAL { false}
    ::= { dot11WirelessMgmtOptionsEntry 11 }

dot11MgmtOptionTIMBroadcastImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute, when true, indicates that the station implementation is
        capable of supporting TIM Broadcast when the
        dot11WirelessManagementImplemented is set to true."
    ::= { dot11WirelessMgmtOptionsEntry 12}

dot11MgmtOptionTIMBroadcastActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        This attribute, when true, indicates that TIM broadcast is enabled. TIM
        broadcast is disabled otherwise."
    DEFVAL { false}
    ::= { dot11WirelessMgmtOptionsEntry 13}

dot11MgmtOptionProxyARPImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute, when true, indicates that the station implementation is
        capable of supporting the Proxy ARP service, when the
        dot11WirelessManagementImplemented is set to true."
    ::= { dot11WirelessMgmtOptionsEntry 14 }

dot11MgmtOptionProxyARPActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        This attribute, when true, indicates that the capability of the AP to pro-
        vide the Proxy ARP service is enabled. The capability is disabled, other-
        wise."
    DEFVAL { false}
    ::= { dot11WirelessMgmtOptionsEntry 15 }

dot11MgmtOptionBSSTransitionImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This is a capability variable. Its value is determined by device capabil-
        ities. This attribute, when true, indicates that the station implementa-
        tion is capable of supporting BSS Transition Management, when the
        dot11WirelessManagementImplemented is set to true."
    ::= { dot11WirelessMgmtOptionsEntry 16 }

dot11MgmtOptionBSSTransitionActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        This attribute, when true, indicates that the capability of the station to
        provide BSS Transition is enabled. The capability is disabled, otherwise.
        "
    DEFVAL { false}
    ::= { dot11WirelessMgmtOptionsEntry 17 }

dot11MgmtOptionQoSTrafficCapabilityImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute, when true, indicates that the station implementation is
        capable of supporting QoS Traffic Capability when the
        dot11WirelessManagementImplemented is set to true."
    ::= { dot11WirelessMgmtOptionsEntry 18 }

dot11MgmtOptionQoSTrafficCapabilityActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        This attribute, when true, indicates that the capability of the station to
        provide QoS Traffic Capability is enabled. QoS Traffic Capability is dis-
        abled otherwise."
    DEFVAL { false}
    ::= { dot11WirelessMgmtOptionsEntry 19 }

dot11MgmtOptionACStationCountImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
```

```
      "This is a capability variable.
      Its value is determined by device capabilities.

      This attribute, when true, indicates that the station implementation is
      capable of supporting AC Station Count when the
      dot11WirelessManagementImplemented is set to true."
   ::= { dot11WirelessMgmtOptionsEntry 20 }

dot11MgmtOptionACStationCountActivated OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      This attribute, when true, indicates that the capability of the station to
      provide AC Station Count is enabled. AC Station Count is disabled other-
      wise."
   DEFVAL { false}
   ::= { dot11WirelessMgmtOptionsEntry 21 }

dot11MgmtOptionCoLocIntfReportingImplemented OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      This attribute, when true, indicates that the station implementation is
      capable of supporting Colocated Interference Reporting. The capability is
      disabled, otherwise."
   ::= { dot11WirelessMgmtOptionsEntry 22 }

dot11MgmtOptionCoLocIntfReportingActivated OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      This attribute, when true, indicates that the capability of the station to
      support Colocated Interference Reporting is enabled. The capability is
      disabled, otherwise."
   DEFVAL { false}
   ::= { dot11WirelessMgmtOptionsEntry 23 }

dot11MgmtOptionMotionDetectionImplemented OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      This attribute, when true, indicates that the station implementation is
      capable of supporting motion detection when the
      dot11WirelessManagementImplemented is set to true. "
   ::= { dot11WirelessMgmtOptionsEntry 24 }
```

```
dot11MgmtOptionMotionDetectionActivated OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      This attribute, when true, indicates that the capability to support motion
      detection is enabled."
   DEFVAL { false}
   ::= { dot11WirelessMgmtOptionsEntry 25 }

dot11MgmtOptionTODImplemented OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      This attribute, when true, indicates that the station implementation is
      capable of supporting Time Of Departure for Clause 16 transmitted frames,
      Clause 18 transmitted frames, Clause 17 transmitted frames, Clause 19
      transmitted frames and Clause 20 transmitted frames when the
      dot11WirelessManagementImplemented is set to true."
   ::= { dot11WirelessMgmtOptionsEntry 26 }

dot11MgmtOptionTODActivated OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      This attribute, when true, indicates that the capability to support Time
      Of Departure frames for transmitted Clause 16, Clause 18, Clause 17,
      Clause 19 and Clause 20 frames is enabled."
   DEFVAL { false}
   ::= { dot11WirelessMgmtOptionsEntry 27 }

dot11MgmtOptionTimingMsmtImplemented OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      This attribute, when true, indicates that the station implementation is
      capable of supporting Timing Measurement capability when the
      dot11WirelessManagementImplemented is set to true."
   ::= { dot11WirelessMgmtOptionsEntry 28 }

dot11MgmtOptionTimingMsmtActivated OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity or the SME.
```

```
        Changes take effect as soon as practical in the implementation.

        This attribute, when true, indicates that the station capability for Timing Measurement is enabled. False indicates the station has no Timing Measurement capability or that the capability is present but is disabled."
    DEFVAL { false}
    ::= { dot11WirelessMgmtOptionsEntry 29 }

dot11MgmtOptionChannelUsageImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute, when true, indicates that the station implementation is
        capable of supporting Channel Usage when the
        dot11WirelessManagementImplemented is set to true."
    ::= { dot11WirelessMgmtOptionsEntry 30 }

dot11MgmtOptionChannelUsageActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        This attribute, when true, indicates that Channel Usage is enabled. Channel Usage is disabled otherwise."
    DEFVAL { false}
    ::= { dot11WirelessMgmtOptionsEntry 31 }

dot11MgmtOptionTriggerSTAStatisticsActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        This attribute, when true, indicates that the capability of the station to
        provide triggered STA statistics is enabled. The capability is disabled
        otherwise"
    DEFVAL { false}
    ::= { dot11WirelessMgmtOptionsEntry 32 }

dot11MgmtOptionSSIDListImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute, when true, indicates that the station implementation is
        capable of supporting the SSID List capability when the
        dot11WirelessManagementImplemented is true."
    ::= { dot11WirelessMgmtOptionsEntry 33 }

dot11MgmtOptionSSIDListActivated OBJECT-TYPE
```

```
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        This attribute, when true, indicates that the capability of the station to
        support the SSID List capability is enabled. The capability is disabled,
        otherwise"
    DEFVAL { false}
    ::= { dot11WirelessMgmtOptionsEntry 34 }

dot11MgmtOptionMulticastDiagnosticsActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        This attribute, when true, indicates that the capability of the station to
        provide Multicast Diagnostic Reporting is enabled. The capability is dis-
        abled, otherwise."
    DEFVAL { false}
    ::= { dot11WirelessMgmtOptionsEntry 35 }

dot11MgmtOptionLocationTrackingImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute, when true, indicates that the station implementation is
        capable of supporting Location Track when the
        dot11WirelessManagementImplemented is true."
    ::= { dot11WirelessMgmtOptionsEntry 36 }

dot11MgmtOptionLocationTrackingActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        This attribute, when true, indicates that the capability of the station to
        provide Location Track is enabled. The capability is disabled otherwise."
    DEFVAL { false}
    ::= { dot11WirelessMgmtOptionsEntry 37 }

dot11MgmtOptionDMSImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.
        This attribute, when true, indicates that the station implementation is
        capable of supporting DMS when the dot11WirelessManagementImplemented is
        true."
    ::= { dot11WirelessMgmtOptionsEntry 38 }

dot11MgmtOptionDMSActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        This attribute, when true, indicates that DMS is enabled. DMS is disabled
        otherwise."
    DEFVAL { false}
    ::= { dot11WirelessMgmtOptionsEntry 39 }

dot11MgmtOptionUAPSDCoexistenceImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute, when true, indicates that the Station implementation is
        capable of supporting U-APSD Coexistence when the
        dot11WirelessManagementImplemented is set to true."
    ::= { dot11WirelessMgmtOptionsEntry 40}

dot11MgmtOptionUAPSDCoexistenceActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        This attribute, when true, indicates that U-APSD Coexistence is enabled.
        U-APSD Coexistence is disabled, otherwise."
    DEFVAL { false}
    ::= { dot11WirelessMgmtOptionsEntry 41}

dot11MgmtOptionWNMNotificationImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute, when true, indicates that the station implementation is
        capable of supporting WNM-Notification when the
        dot11WirelessManagementImplemented is set to true."
    ::= { dot11WirelessMgmtOptionsEntry 42}

dot11MgmtOptionWNMNotificationActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
```

```
      "This is a control variable.
      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      This attribute, when true, indicates that the capability of the station to
      provide WNM-Notification is enabled. The capability is disabled, other-
      wise."
   DEFVAL { false}
   ::= { dot11WirelessMgmtOptionsEntry 43}

dot11MgmtOptionUTCTSFOffsetImplemented OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      This attribute, when true, indicates that the Station implementation is
      capable of supporting UTC TSF Offset advertisement when the
      dot11WirelessManagementImplemented is set to true."
   ::= { dot11WirelessMgmtOptionsEntry 44}

dot11MgmtOptionUTCTSFOffsetActivated OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      This attribute, when true, indicates that UTC TSF Offset advertisement is
      enabled at the station. The capability is disabled, otherwise."
   DEFVAL { false}
   ::= { dot11WirelessMgmtOptionsEntry 45}

-- ********************************************************************
 -- * dot11LocationServices TABLE
 -- ********************************************************************

dot11LocationServicesNextIndex OBJECT-TYPE
   SYNTAX Unsigned32 (0..4294967295)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "Identifies a hint for the next value of dot11LocationServicesIndex to be
      used in a row creation attempt for dot11LocationServicesTable. If no new
      rows can be created for some reason, such as memory, processing require-
      ments, etc, the SME shall set this attribute to 0. It shall update this
      attribute to a proper value other than 0 as soon as it is capable of
      receiving new measurement requests. The nextIndex is not necessarily
      sequential nor monotonically increasing."
   ::= { dot11smt 19 }

dot11LocationServicesTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11LocationServicesEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "Group contains conceptual table of attributes for
      WNM LocationServices."
   ::= { dot11smt 20 }
```

```
dot11LocationServicesEntry OBJECT-TYPE
   SYNTAX Dot11LocationServicesEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "An entry in the dot11LocationServicesTable
      Indexed by dot11LocationServicesIndex."
   INDEX { dot11LocationServicesIndex }
   ::= { dot11LocationServicesTable 1 }

Dot11LocationServicesEntry ::=
   SEQUENCE {
      dot11LocationServicesIndex                             Unsigned32,
      dot11LocationServicesMACAddress                        MacAddress,
      dot11LocationServicesLIPIndicationMulticastAddress     MacAddress,
      dot11LocationServicesLIPReportIntervalUnits            INTEGER,
      dot11LocationServicesLIPNormalReportInterval           Unsigned32,
      dot11LocationServicesLIPNormalFramesperChannel         Unsigned32,
      dot11LocationServicesLIPInMotionReportInterval         Unsigned32,
      dot11LocationServicesLIPInMotionFramesperChannel       Unsigned32,
      dot11LocationServicesLIPBurstInterframeInterval        Unsigned32,
      dot11LocationServicesLIPTrackingDuration               Unsigned32,
      dot11LocationServicesLIPEssDetectionInterval           Unsigned32,
      dot11LocationServicesLocationIndicationChannelList     OCTET STRING,
      dot11LocationServicesLocationIndicationBroadcastDataRate
                                                             Unsigned32,
      dot11LocationServicesLocationIndicationOptionsUsed     OCTET STRING,
      dot11LocationServicesLocationIndicationIndicationParameters
                                                             OCTET STRING,
      dot11LocationServicesLocationStatus                    Unsigned32
   }

dot11LocationServicesIndex OBJECT-TYPE
   SYNTAX Unsigned32 (1..4294967295)
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "This attribute contains an auxiliary index into the
      dot11LocationServicesTable."
   ::= { dot11LocationServicesEntry 1 }

dot11LocationServicesMACAddress OBJECT-TYPE
   SYNTAX MacAddress
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This attribute is the MAC address of the STA reporting location informa-
      tion."
   ::= { dot11LocationServicesEntry 2 }

dot11LocationServicesLIPIndicationMulticastAddress OBJECT-TYPE
   SYNTAX MacAddress
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This attribute is the destination address to which the Location Track
      Notification frames are sent in a non-IBSS network; see 8.4.2.73.2 and
      10.23.4.1."
   ::= { dot11LocationServicesEntry 3 }

dot11LocationServicesLIPReportIntervalUnits OBJECT-TYPE
   SYNTAX INTEGER {
      hours(0),
      minutes(1),
      seconds(2),
      milliseconds(3)
      }
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This attribute contains the Location Indication Parameters Report Inter-
      val Units value."
   ::= { dot11LocationServicesEntry 4 }

dot11LocationServicesLIPNormalReportInterval OBJECT-TYPE
   SYNTAX Unsigned32 (0..65535)
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This attribute contains the time interval, expressed in the units indi-
      cated in the Report Interval Units field, at which the reporting STA is
      expected to transmit one or more Location Track Notification frames if
      either dot11MgmtOptionMotionDetectionActivated is false or the STA is sta-
      tionary. The STA will not transmit Location Track Notification frames when
      the Normal Report Interval is 0."
   ::= { dot11LocationServicesEntry 5 }

dot11LocationServicesLIPNormalFramesperChannel OBJECT-TYPE
   SYNTAX Unsigned32 (0..255)
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This attribute contains the number of Location Track Notification frames
      per channel sent or expected to be sent by the STA at each Normal Report
      Interval."
   ::= { dot11LocationServicesEntry 6 }

dot11LocationServicesLIPInMotionReportInterval OBJECT-TYPE
   SYNTAX Unsigned32 (0..65535)
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This attribute contains the time interval, expressed in the units indi-
      cated in the Report Interval Units field, at which the STA reports its
      location by sending a Location Track Notification frame when the reporting
      STA is in motion. If dot11MgmtOptionMotionDetectionActivated is false,
      this field is set to 0."
   ::= { dot11LocationServicesEntry 7}

dot11LocationServicesLIPInMotionFramesperChannel OBJECT-TYPE
   SYNTAX Unsigned32 (0..255)
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This attribute contains the number of Location Track Notification frames
      per channel sent or expected to be sent by the STA at each In-Motion
      Report Interval. If dot11MgmtOptionMotionDetectionActivated is false, this
      field is set to 0."
   ::= { dot11LocationServicesEntry 8 }

dot11LocationServicesLIPBurstInterframeInterval OBJECT-TYPE
   SYNTAX Unsigned32 (0..255)
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This attribute contains the target time interval, expressed in millisec-
      onds, between the transmissions of each of the Normal or In-Motion frames
      on the same channel. The Burst Inter-frame interval value is set to 0 to
      indicate that frames will be transmitted with no target inter-frame
      delay."
   ::= { dot11LocationServicesEntry 9 }

dot11LocationServicesLIPTrackingDuration OBJECT-TYPE
   SYNTAX Unsigned32 (0..255)
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This attribute contains the amount of time, in minutes, that a STA sends
      the Location Track Notification frames. The duration starts as soon as the
      STA sends a Location Configuration Response frame with a Location Status
      value of Success. If the Tracking Duration value is a nonzero value the
      STA will send Location Track Notification Frames, based on the Normal and
      In-Motion Report Interval field values, until the duration ends. If the
      Tracking Duration is 0 the STA will continuously send Location Track Noti-
      fication frames as defined by Normal and In-Motion Report Interval field
      values until transmission is terminated based on 10.23.4.2 procedures."
   ::= { dot11LocationServicesEntry 10}

dot11LocationServicesLIPEssDetectionInterval OBJECT-TYPE
   SYNTAX Unsigned32 (0..255)
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This attribute contains the interval, in minutes, that a STA checks for
      beacons transmitted by one or more APs belonging to the same ESS that con-
      figured the STA. If no beacons from the ESS are received for this period,
      the STA terminates transmission of Location Track Notification frames as
      described in 10.23.4.2 procedures. The ESS Detection Interval field is not
      used when the ESS Detection Interval field value is set to 0."
   ::= { dot11LocationServicesEntry 11}

dot11LocationServicesLocationIndicationChannelList OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE (2..254))
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This attribute contains one or more Operating Class and Channel octet
      pairs."
   ::= { dot11LocationServicesEntry 12}

dot11LocationServicesLocationIndicationBroadcastDataRate OBJECT-TYPE
   SYNTAX Unsigned32 (0..4294967295)
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This attribute specifies the target data rate at which the STA transmits
      Location Track Notification Frames. The Broadcast Target Data Rate field
      format is specified by the Rate Identification field defined in 8.4.1.32.
      A value of 0 indicates the STA transmits Location Track Notification
      frames at a rate chosen by the STA transmitting the Location Track Notifi-
      cation frames."
   DEFVAL { 0 }
   ::= { dot11LocationServicesEntry 13}

dot11LocationServicesLocationIndicationOptionsUsed OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(1))
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This attribute indicates the location configuration options used for
      transmitting Location Track Notification Frames."
   ::= { dot11LocationServicesEntry 14}
```

```
dot11LocationServicesLocationIndicationIndicationParameters OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE (1..255))
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This attribute indicates the location Indication Parameters used for
      transmitting Location Track Notification Frames."
   ::= { dot11LocationServicesEntry 15}

dot11LocationServicesLocationStatus OBJECT-TYPE
   SYNTAX Unsigned32 (0..255)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This attribute contains the Location Status value as indicated in
      Table 8-137, Event Report Status."
   ::= { dot11LocationServicesEntry 16 }

-- ********************************************************************
 -- * End of dot11LocationServices TABLE
 -- ********************************************************************

 -- ********************************************************************
 -- * dot11WirelessMGTEvent TABLE
 -- ********************************************************************
dot11WirelessMGTEventTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11WirelessMGTEventEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "Group contains the current list of WIRELESS Management reports that have
      been received by the MLME. The report tables shall be maintained as FIFO
      to preserve freshness, thus the rows in this table can be deleted for mem-
      ory constraints or other implementation constraints determined by the ven-
      dor.
      New rows shall have different RprtIndex values than those deleted within
      the range limitation of the index. One easy way is to monotonically
      increase the EventIndex for new reports being written in the table."
   ::= { dot11smt 21 }

dot11WirelessMGTEventEntry OBJECT-TYPE
   SYNTAX Dot11WirelessMGTEventEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "An entry in the dot11WirelessMGTEventTable
      Indexed by dot11WirelessMGTEventIndex."
   INDEX { dot11WirelessMGTEventIndex }
   ::= { dot11WirelessMGTEventTable 1 }

Dot11WirelessMGTEventEntry ::=
   SEQUENCE {
      dot11WirelessMGTEventIndex                           Unsigned32,
      dot11WirelessMGTEventMACAddress                      MacAddress,
      dot11WirelessMGTEventType                            INTEGER,
      dot11WirelessMGTEventStatus                          INTEGER,
      dot11WirelessMGTEventTSF                             TSFType,
      dot11WirelessMGTEventUTCOffset                       OCTET STRING,
      dot11WirelessMGTEventTimeError                       OCTET STRING,
      dot11WirelessMGTEventTransitionSourceBSSID           MacAddress,
      dot11WirelessMGTEventTransitionTargetBSSID           MacAddress,
      dot11WirelessMGTEventTransitionTime                  Unsigned32,
      dot11WirelessMGTEventTransitionReason                INTEGER,
      dot11WirelessMGTEventTransitionResult                 Unsigned32,
      dot11WirelessMGTEventTransitionSourceRCPI             Unsigned32,
      dot11WirelessMGTEventTransitionSourceRSNI             Unsigned32,
      dot11WirelessMGTEventTransitionTargetRCPI             Unsigned32,
      dot11WirelessMGTEventTransitionTargetRSNI             Unsigned32,
      dot11WirelessMGTEventRSNATargetBSSID                  MacAddress,
      dot11WirelessMGTEventRSNAAuthenticationType           OCTET STRING,
      dot11WirelessMGTEventRSNAEAPMethod                    OCTET STRING,
      dot11WirelessMGTEventRSNAResult                       Unsigned32,
      dot11WirelessMGTEventRSNARSNElement                   OCTET STRING,
      dot11WirelessMGTEventPeerSTAAddress                   MacAddress,
      dot11WirelessMGTEventPeerOperatingClass               Unsigned32,
      dot11WirelessMGTEventPeerChannelNumber                Unsigned32,
      dot11WirelessMGTEventPeerSTATxPower                   Integer32,
      dot11WirelessMGTEventPeerConnectionTime               Unsigned32,
      dot11WirelessMGTEventPeerPeerStatus                   Unsigned32,
      dot11WirelessMGTEventWNMLog                           OCTET STRING
   }

dot11WirelessMGTEventIndex OBJECT-TYPE
   SYNTAX Unsigned32 (1..4294967295)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This attribute contains an auxiliary index into the
      dot11WirelessMGTEventTable."
   ::= { dot11WirelessMGTEventEntry 1 }

dot11WirelessMGTEventMACAddress OBJECT-TYPE
   SYNTAX MacAddress
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This attribute is the MAC address of the STA providing the Event Report."
   ::= { dot11WirelessMGTEventEntry 2 }

dot11WirelessMGTEventType OBJECT-TYPE
   SYNTAX INTEGER {
      transition(0),
      rsna(1),
      peerToPeer(2),
      wnmLog(3),
      vendorSpecific(221)
      }
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This attribute indicates the request type of this WNM Event request."
   ::= { dot11WirelessMGTEventEntry 3 }

dot11WirelessMGTEventStatus OBJECT-TYPE
   SYNTAX INTEGER {
      successful(0),
      requestFailed(1),
      requestRefused(2),
      requestIncapable(3),
      detectedFrequentTransition(4)
      }
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This attribute contains the status value included in the Event Report."
   ::= { dot11WirelessMGTEventEntry 4 }
```

```
dot11WirelessMGTEventTSF OBJECT-TYPE
    SYNTAX TSFType
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This attribute contains the value of the Event timestamp field."
    ::= { dot11WirelessMGTEventEntry 5 }

dot11WirelessMGTEventUTCOffset OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(10))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This attribute indicates the UTC Offset Time Value optionally included in
        the Event Report."
    ::= { dot11WirelessMGTEventEntry 6}

dot11WirelessMGTEventTimeError OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(5))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This attribute contains the value of the Event Time Error field option-
        ally included in the Event Report."
    ::= { dot11WirelessMGTEventEntry 7}

dot11WirelessMGTEventTransitionSourceBSSID OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This attribute contains the value of the Source BSSID field in a Transi-
        tion event report."
    ::= { dot11WirelessMGTEventEntry 8}

dot11WirelessMGTEventTransitionTargetBSSID OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This attribute contains the value of the Target BSSID field in a Transi-
        tion event report."
    ::= { dot11WirelessMGTEventEntry 9}

dot11WirelessMGTEventTransitionTime OBJECT-TYPE
    SYNTAX Unsigned32 (0..65535)
    UNITS "TUs"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This attribute indicates the transition time for the reported transition
        event in TUs. The Transition time is defined as the time difference
        between the starting time and the ending time of a transition between APs,
        even if the transition results in remaining on the same AP. Start and end
        times for a transition event are defined in 10.23.2.2"
    ::= { dot11WirelessMGTEventEntry 10}

dot11WirelessMGTEventTransitionReason OBJECT-TYPE
    SYNTAX INTEGER {
        unspecified(0),
        excessiveFrameLossRatesPoorConditions(1),
        excessiveDelayForCurrentTrafficStreams(2),
        insufficientQosCapacityForCurrentTrafficStreams(3),
        firstAssociationToEss(4),
        loadBalancing(5),
        betterApFound(6),
        deauthenticatedDisassociatedFromPreviousAp(7),
        certificateToken(8),
        apFailedIeee8021XEapAuthentication(9),
        apFailed4wayHandshake(10),
        excessiveDataMICFailures(11),
        exceededFrameTransmissionRetryLimit(12),
        ecessiveBroadcastDisassociations(13),
        excessiveBroadcastDeauthentications(14),
        previousTransitionFailed(15)
        }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This attribute indicates the reason for the reported BSS Transition
        event. The format for this list of reasons is further detailed in
        8.4.2.70.2."
    ::= { dot11WirelessMGTEventEntry 11}

dot11WirelessMGTEventTransitionResult OBJECT-TYPE
    SYNTAX Unsigned32 (0..65535)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This attribute indicates the result of the attempted transition and is
        set to one of the status codes specified in Table 8-37."
    ::= { dot11WirelessMGTEventEntry 12 }

dot11WirelessMGTEventTransitionSourceRCPI OBJECT-TYPE
    SYNTAX Unsigned32 (0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This attribute indicates the received channel power of the most recently
        measured frame from the Source BSSID before the STA reassociates to the
        Target BSSID. The Source RCPI is a logarithmic function of the received
        signal power, as defined in the RCPI measurement subclause for the PHY
        Type."
    ::= { dot11WirelessMGTEventEntry 13 }

dot11WirelessMGTEventTransitionSourceRSNI OBJECT-TYPE
    SYNTAX Unsigned32 (0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This attribute indicates the received signal-to-noise indication of the
        most recently measured frame from the Source BSSID before the STA reasso-
        ciates to the Target BSSID. The Source RSNI is a logarithmic function of
        the signal-to-noise ratio, as defined in 8.4.2.43."
    ::= { dot11WirelessMGTEventEntry 14 }

dot11WirelessMGTEventTransitionTargetRCPI OBJECT-TYPE
    SYNTAX Unsigned32 (0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This attribute indicates the received channel power of the first measured
        frame just after STA reassociates to the Target BSSID. If association with
        target BSSID failed, the Target RCPI field indicates the received channel
        power of the most recently measured frame from the Target BSSID. The Tar-
        get RCPI is a logarithmic function of the received signal power, as
        defined in the RCPI measurement subclause for the PHY Type."
    ::= { dot11WirelessMGTEventEntry 15 }
```

```
dot11WirelessMGTEventTransitionTargetRSNI OBJECT-TYPE
   SYNTAX Unsigned32 (0..255)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This attribute indicates the received signal-to-noise indication of the
      first measured frame just after STA reassociates to the Target BSSID. If
      association with target BSSID failed, the Target RCPI field indicates the
      received signal-to-noise indication of the most recently measured frame
      from the Target BSSID. The Target RSNI is a logarithmic function of the
      signal-to-noise ratio, as defined in 8.4.2.43."
   ::= { dot11WirelessMGTEventEntry 16 }

dot11WirelessMGTEventRSNATargetBSSID OBJECT-TYPE
   SYNTAX MacAddress
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This attribute contains the value of the Target BSSID field in an RSNA
      event report."
   ::= { dot11WirelessMGTEventEntry 17 }

dot11WirelessMGTEventRSNAAuthenticationType OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE (4))
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This attribute indicates the AKM suite, as defined in Table 8-101. The
      first three octets indicate the OUI. The last octet indicates the suite
      type."
   ::= { dot11WirelessMGTEventEntry 18 }

dot11WirelessMGTEventRSNAEAPMethod OBJECT-TYPE
   SYNTAX OCTET STRING(SIZE (1..8))
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This attribute indicates a value that identifies the EAP Method. When the
      Authentication Type field is set to the value of either 00-0F-AC:1
      (Authentication negotiated over IEEE 802.1X or using PMKSA caching as
      defined in 11.5.9.3) or 00-0F-AC:3 (AKM suite selector for Fast BSS Tran-
      sition as defined in 11.6.1.7), the EAP Method field contains the IANA
      assigned EAP type defined at http://www.iana.org/assignments/eap-numbers.
      The EAP type contains either the legacy type (1 octet) or the expanded
      type (1 octet type = 254, 3-octet Vendor ID, 4-octet Vendor-Type). The EAP
      Method field is set to 0 otherwise."
   ::= { dot11WirelessMGTEventEntry 19 }

dot11WirelessMGTEventRSNAResult OBJECT-TYPE
   SYNTAX Unsigned32 (0..65535)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This attribute indicates the result of the RSNA event and is set to one
      of the status codes specified in Table 8-37."
   ::= { dot11WirelessMGTEventEntry 20}

dot11WirelessMGTEventRSNARSNElement OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(0..257))
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This attribute contains the entire contents of the negotiated RSNE at the
      time of the authentication attempt. The format of the RSNE is defined in
      8.4.2.27."
   ::= { dot11WirelessMGTEventEntry 21}

dot11WirelessMGTEventPeerSTAAddress OBJECT-TYPE
   SYNTAX MacAddress
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This attribute indicates the MAC address of the peer STA or IBSS BSSID is
      equal to the indicated MAC address. If this event is for a Peer-to-Peer
      link in an infrastructure BSS, this field contains the MAC address of the
      peer STA. If this event is for a Peer-to-Peer link in an IBSS, this field
      contains the BSSID of the IBSS."
   ::= { dot11WirelessMGTEventEntry 22 }

dot11WirelessMGTEventPeerOperatingClass OBJECT-TYPE
   SYNTAX Unsigned32 (1..255)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This attribute indicates the channel set for this Peer-to-Peer Event
      report. Country, Operating Class, and Channel Number together specify the
      channel frequency and spacing for this measurement request. Valid values
      of Operating Class are shown in Annex E."
   ::= { dot11WirelessMGTEventEntry 23 }

dot11WirelessMGTEventPeerChannelNumber OBJECT-TYPE
   SYNTAX Unsigned32 (1..255)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This attribute indicates the current operating channel for this Peer-to-
      Peer Event report. The Channel Number is only defined within the indicated
      Operating Class as shown in Annex E."
   ::= { dot11WirelessMGTEventEntry 24 }

dot11WirelessMGTEventPeerSTATxPower OBJECT-TYPE
   SYNTAX Integer32 (-128..127)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This attribute indicates the STA transmit power used for the Peer-to-Peer
      link. The STA Tx Power field indicates the target transmit power at the
      antenna in dBm with a tolerance of +/-5 dB for the lowest basic rate of
      the reporting STA. A value of -128 indicates that the value is unknown."
   ::= { dot11WirelessMGTEventEntry 25 }

dot11WirelessMGTEventPeerConnectionTime OBJECT-TYPE
   SYNTAX Unsigned32 (0..16777215)
   UNITS "seconds"
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This attribute indicates a value representing the connection time for the
      reported Peer-to-Peer event. If the Peer Status is 0, this field indicates
      the duration of the Direct Link. If the Peer Status is 1, this field indi-
      cates the time difference from the time the Direct Link was established to
      the time at which the reporting STA generated the event report. If the
      Peer Status is 2, this field indicates the duration of the IBSS member-
      ship. If the Peer Status is 3, this field indicates the time difference
      from the time the STA joined the IBSS to the time at which the reporting
      STA generated the event report. See 10.23.2.4."
   ::= { dot11WirelessMGTEventEntry 26 }
```

```
dot11WirelessMGTEventPeerPeerStatus OBJECT-TYPE
   SYNTAX Unsigned32 (0..255)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This attribute indicates the Peer link connection status as indicated in
      Table 8-139. See 8.4.2.70.4."
   ::= { dot11WirelessMGTEventEntry 27 }

dot11WirelessMGTEventWNMLog OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(0..2284))
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This attribute contains the entire syslog message, consisting of the PRI,
      HEADER, and MSG portion of a WNM Log message as described in IETF RFC
      3164-2001. The TAG field of the MSG portion of the message is a 17 octet
      string containing the ASCII representation of the STA MAC address using
      hexadecimal notation with colons between octets. The octet containing the
      individual/group bit occurs last, and that bit is in the least significant
      position within that octet. See 10.23.2.5."
   ::= { dot11WirelessMGTEventEntry 28 }

 -- ******************************************************************
 -- * End of dot11WirelessMGTEvent TABLE
 -- ******************************************************************
-- *********************************************************************
-- * IEEE 802.11 RM and WNM Interface MIB
-- *********************************************************************

-- * The primary interface to the Radio Measurements and Wireless
-- * Network Management functions is meant to be real-time information
-- * obtained through the request/response mechanisms of RM and WNM.
-- * A secondary interface to the measurements and management functions
-- * is through retention of information in this MIB. Non-SNMP requests
-- * for information are obtained via object IDs (OIDs) through the NDIS
-- * or wireless interfaces in the operating systems. SNMP requests for
-- * information are obtained via SNMP SETs and GETs (see [B24]).

-- *  dot11RMRequest and dot11RMReport Usage
-- *
-- * The dot11RMRequest and dot11RMReport portions of the RM MIB
-- * provide access to the Radio Measurement service. By performing
-- * SET operations on the various dot11RMRequest MIB objects,
-- * radio measurements may be initiated directly on the local STA or
-- * on any peer station within the same BSS. Subsequently, by
-- * performing GET operations on the various dot11RMReport MIB
-- * objects the results of the requested measurements may be
-- * retrieved.
-- *
-- * In the diagram below, a radio measurement could be initiated
-- * for STA x by performing a MIB.set operation on the RM MIB of
-- * STA x and specifying the MAC address of STA x in
-- * dot11RMRqstTargetAdd. Additionally, it is possible to have STA x
-- * request a measurement from STA y by performing a MIB.set operation
-- * on the SME MIB of STA x and specifying the MAC address of STA y in
-- * dot11RMRqstTargetAdd. In both cases the result of the measurements
-- * can be retrieved by performing a MIB.get operation on the RM MIB
-- * of STA x upon completion of the measurement.
-- *
-- *
-- *          MIB.Set                                        MIB.Set
-- *             or                                             or
-- *          MIB.Get                                        MIB.Get
-- *    +========|=========+                           +========|=========+
-- *    | SME    |         |                           | SME    |         |
-- *    |       \ /        |                           |       \ /        |
-- *    |   +=========+    |                           |   +=========+    |
-- *    |   | RM and  |    |                           |   | RM and  |    |
-- *    |   | WNM MIB |    |                           |   | WNM MIB |    |
-- *    |   |         |    |                           |   |         |    |
-- *    |   |         |    |                           |   |         |    |
-- *    |   +=========+    |                           |   +=========+    |
-- *    |                  |                           |                  |
-- *    |   / \            |                           |   / \            |
-- *    |    |   MREQUEST  |                           |    |   MREQUEST  |
-- *    +====+=============+                           +====+=============+
-- *    |    |   MREPORT   |                           |    |   MREPORT   |
-- *    |   \ /  MEASURE   |     Action frames         |   \ /  MEASURE   |
-- *    |                  | <==Measurement Request==> |                  |
-- *    |                  | <==Measurement Report===> |                  |
-- *    | MLME             |                           | MLME             |
-- *    +==================+                           +==================+
-- *           STA x                                          STA y
-- *
-- * Each STA maintains a single dot11RMRequestTable in the SME MIB
-- * used to initiate RM Measurement Requests. Each dot11RMRequestEntry
-- * in the table represents an individual Measurement Request that
-- * makes up a complete Measurement Request frame.
-- * Multiple Measurement Requests may be concatenated into a single
-- * Measurement Request frame by setting the same
-- * dot11RMRqstToken value into multiple dot11RMRequestEntrys.
-- *
-- * Each row, dot11RMRequestEntry, of the dot11RMRequestTable
-- * provides read-create access for the initiation of a measurement
-- * request. The dot11RMRequestNextIndex object can be used to
-- * determine which is the next available row. Each row corresponding to
-- * one measurement in the sequence is created with a dot11RMRqstRowStatus
-- * equal to notInService. Once the dot11RMRequestEntry(s) have been
-- * created for a desired measurement sequence the corresponding
-- * dot11RMRqstRowStatus(s) objects are set to active to indicate that
-- * the SME can trigger the appropriate MLME primitives. Upon processing
-- * the request, the SME returns the corresponding dot11RMRqstRowStatus(s)
-- * object to notInService and are now available for additional
-- * measurement requests.
-- *
-- * After a radio measurement is complete the RM populates the RMReport
-- * objects with the results of the measurement. Each STA maintains a set
-- * of RMReport tables, one corresponding to each measurement type. The
-- * results of the entire measurement sequence are spread across the tables
-- * based on the type of measurements requested. Each xxxReportEntry
-- * within a xxxReportTable contains a xxxRprtRqstToken that corresponds
-- * to the original dot11RMRqstToken in the measurement request. So the
-- * results of the measurement can be collected by searching the appropriate
-- * xxxReportTables and retrieve any reports with the matching request
-- * token.
-- *
-- * Similar structures and mechanisms are used for WNM
-- * Request and Reports. The WNM MIB definitions follow the RM MIB definitions
-- * in this Annex.

-- **********************************************************************

-- ********************************************************************
-- * Radio Measurement Interface MIB
-- ********************************************************************

   dot11RadioMeasurement OBJECT IDENTIFIER ::= { dot11smt 14 }
-- ********************************************************************
-- * Radio Measurement Requests
-- ********************************************************************
dot11RMRequest OBJECT IDENTIFIER ::= { dot11RadioMeasurement 1 }

-- ********************************************************************
-- * dot11RMRequest TABLE
-- ********************************************************************
dot11RMRequestNextIndex OBJECT-TYPE
   SYNTAX Unsigned32(0..65535)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when able to accept a new request.

      Identifies a hint for the next value of dot11RMRqstIndex to be used in a
      row creation attempt for dot11RMRequestTable. If no new rows can be cre-
      ated for some reason, such as memory, processing requirements, etc., the
      SME sets this attribute to 0. It updates this attribute to a proper value
      other than 0 as soon as it is capable of receiving new measurement
      requests. The nextIndex is not necessarily sequential nor monotonically
      increasing."
   ::= { dot11RMRequest 1 }

dot11RMRequestTable OBJECT-TYPE
```

```
    SYNTAX SEQUENCE OF Dot11RMRequestEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "This group contains the current list of requests for RM reports to be
        issued and have been issued until removed. A network manager adds a RM
        request by creating a row with createAndWait row status and then filling
        in the request parameters/attributes. The request becomes active to be
        issued when the row status is set to Active. The columnar objects or
        attributes other than the rowStatus are not written if the rowStatus is
        Active. The request rows can be deleted, if commanded by a network manager
        via changing the value of dot11RMRqstRowStatus to Destroy. This may leave
        orphaned rows if a manager crashes and forgets which rows are being used
        by it. One recommended way to manage orphaned or finished rows is to
        delete rows if their dot11RMRqstRowStatus remains other than Active for
        longer than a period (recommend at least 5 minutes, IETF RFC 2579). Or
        another recommended way is to delete older rows as needed based on their
        dot11RMRqstTimeStamp values. This can be done by the agent as well as the
        manager. "
    ::= { dot11RMRequest 2 }

dot11RMRequestEntry OBJECT-TYPE
    SYNTAX Dot11RMRequestEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry in the dot11RMRequestTable Indexed by dot11RMRqstIndex."
    INDEX { dot11RMRqstIndex }
    ::= { dot11RMRequestTable 1 }

Dot11RMRequestEntry ::=
    SEQUENCE {
        dot11RMRqstIndex                                 Unsigned32,
        dot11RMRqstRowStatus                             RowStatus,
        dot11RMRqstToken                                 OCTET STRING,
        dot11RMRqstRepetitions                           Unsigned32,
        dot11RMRqstIfIndex                               InterfaceIndex,
        dot11RMRqstType                                  INTEGER,
        dot11RMRqstTargetAdd                             MacAddress,
        dot11RMRqstTimeStamp                             TimeTicks,
        dot11RMRqstChanNumber                            Unsigned32,
        dot11RMRqstOperatingClass                        Unsigned32,
        dot11RMRqstRndInterval                           Unsigned32,
        dot11RMRqstDuration                              Unsigned32,
        dot11RMRqstParallel                              TruthValue,
        dot11RMRqstEnable                                TruthValue,
        dot11RMRqstRequest                               TruthValue,
        dot11RMRqstReport                                TruthValue,
        dot11RMRqstDurationMandatory                     TruthValue,
        dot11RMRqstBeaconRqstMode                        INTEGER,
        dot11RMRqstBeaconRqstDetail                      INTEGER,
        dot11RMRqstFrameRqstType                         INTEGER,
        dot11RMRqstBssid                                 MacAddress,
        dot11RMRqstSSID                                  OCTET STRING,
        dot11RMRqstBeaconReportingCondition              INTEGER,
        dot11RMRqstBeaconThresholdOffset                 Integer32,
        dot11RMRqstSTAStatRqstGroupID                    INTEGER,
        dot11RMRqstLCIRqstSubject                        INTEGER,
        dot11RMRqstLCILatitudeResolution                 Unsigned32,
        dot11RMRqstLCILongitudeResolution                Unsigned32,
        dot11RMRqstLCIAltitudeResolution                 Unsigned32,
        dot11RMRqstLCIAzimuthType                        INTEGER,
        dot11RMRqstLCIAzimuthResolution                  Unsigned32,
        dot11RMRqstPauseTime                             Unsigned32,
        dot11RMRqstTransmitStreamPeerQSTAAddress        MacAddress,
        dot11RMRqstTransmitStreamTrafficIdentifier      Unsigned32,
        dot11RMRqstTransmitStreamBin0Range              Unsigned32,
        dot11RMRqstTrigdQoSAverageCondition             TruthValue,
        dot11RMRqstTrigdQoSConsecutiveCondition         TruthValue,
        dot11RMRqstTrigdQoSDelayCondition               TruthValue,
        dot11RMRqstTrigdQoSAverageThreshold             Unsigned32,
        dot11RMRqstTrigdQoSConsecutiveThreshold         Unsigned32,
        dot11RMRqstTrigdQoSDelayThresholdRange          Unsigned32,
        dot11RMRqstTrigdQoSDelayThreshold               Unsigned32,
        dot11RMRqstTrigdQoSMeasurementCount             Unsigned32,
        dot11RMRqstTrigdQoSTimeout                      Unsigned32,
        dot11RMRqstChannelLoadReportingCondition        INTEGER,
        dot11RMRqstChannelLoadReference                 Unsigned32,
        dot11RMRqstNoiseHistogramReportingCondition     INTEGER,
        dot11RMRqstAnpiReference                        Unsigned32,
        dot11RMRqstAPChannelReport                      OCTET STRING,
        dot11RMRqstSTAStatPeerSTAAddress                MacAddress,
        dot11RMRqstFrameTransmitterAddress              MacAddress,
        dot11RMRqstVendorSpecific                       OCTET STRING,
        dot11RMRqstSTAStatTrigMeasCount                 Unsigned32,
        dot11RMRqstSTAStatTrigTimeout                   Unsigned32,
        dot11RMRqstSTAStatTrigCondition                 OCTET STRING,
        dot11RMRqstSTAStatTrigSTAFailedCntThresh        Unsigned32,
        dot11RMRqstSTAStatTrigSTAFCSErrCntThresh        Unsigned32,
        dot11RMRqstSTAStatTrigSTAMultRetryCntThresh     Unsigned32,
        dot11RMRqstSTAStatTrigSTAFrameDupeCntThresh     Unsigned32,
        dot11RMRqstSTAStatTrigSTARTSFailCntThresh       Unsigned32,
        dot11RMRqstSTAStatTrigSTAAckFailCntThresh       Unsigned32,
        dot11RMRqstSTAStatTrigSTARetryCntThresh         Unsigned32,
        dot11RMRqstSTAStatTrigQoSTrigCondition          OCTET STRING,
        dot11RMRqstSTAStatTrigQoSFailedCntThresh        Unsigned32,
        dot11RMRqstSTAStatTrigQoSRetryCntThresh         Unsigned32,
        dot11RMRqstSTAStatTrigQoSMultRetryCntThresh     Unsigned32,
        dot11RMRqstSTAStatTrigQoSFrameDupeCntThresh     Unsigned32,
        dot11RMRqstSTAStatTrigQoSRTSFailCntThresh       Unsigned32,
        dot11RMRqstSTAStatTrigQoSAckFailCntThresh       Unsigned32,
        dot11RMRqstSTAStatTrigQoSDiscardCntThresh       Unsigned32,
        dot11RMRqstSTAStatTrigRsnaTrigCondition         OCTET STRING,
        dot11RMRqstSTAStatTrigRsnaCMACICVErrCntThresh   Unsigned32,
        dot11RMRqstSTAStatTrigRsnaCMACReplayCntThresh   Unsigned32,
        dot11RMRqstSTAStatTrigRsnaRobustCCMPReplayCntThresh
                                                         Unsigned32,
        dot11RMRqstSTAStatTrigRsnaTKIPICVErrCntThresh   Unsigned32,
        dot11RMRqstSTAStatTrigRsnaTKIPReplayCntThresh   Unsigned32,
        dot11RMRqstSTAStatTrigRsnaCCMPDecryptErrCntThreshUnsigned32,
        dot11RMRqstSTAStatTrigRsnaCCMPReplayCntThresh   Unsigned32
     }

  dot11RMRqstIndex OBJECT-TYPE
     SYNTAX Unsigned32
     MAX-ACCESS not-accessible
     STATUS current
     DESCRIPTION
        "Index for RM Request elements in dot11RMRequestTable, greater than 0."
     ::= { dot11RMRequestEntry 1 }

  dot11RMRqstRowStatus OBJECT-TYPE
     SYNTAX RowStatus
     MAX-ACCESS read-create
     STATUS current
     DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when requesting a measure-
```

```
        ment, and by the SME when accepting a measurement request.

        The Row Status column of the current row, used for tracking status of an
        individual request. When this attribute is set to Active, AND a measure-
        ment request can be unambiguously created based on the parameters in the
        row, then the MLME may proceed to issue the request to its intended tar-
        gets when appropriate. If not, this attribute may be set to Not-ready
        immediately to indicate parametric errors. However, it is the network man-
        agers
        responsibility to correct the error. If the request is successfully issued
        to the target STA, then the rowStatus is set to notInService."
    REFERENCE "8.4.2.23"
    ::= { dot11RMRequestEntry 2 }

dot11RMRqstToken OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when the table entry is
        created, i.e., when requesting a measurement..
        Changes take effect when dot11RMRqstRowStatus is set to Active.

        This attribute indicates a unique string to identify a group of rows to be
        issued as parallel or sequential measurements. To guarantee the uniqueness
        of this token across multiple network managers, it is recommended that
        this token be prefixed with the IP address of the network manager creating
        this row. This token is not necessarily equivalent to the measurement
        tokens in RM request frames. If this attribute is an empty string, then
        this row of request is independent from other requests."
    DEFVAL { "" }
    ::= { dot11RMRequestEntry 3 }

dot11RMRqstRepetitions OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when requesting a measure-
        ment.
        Changes take effect when dot11RMRqstRowStatus is set to Active.

        This attribute indicates the requested number of repetitions for all the
        measurement request elements in this frame. A value of 0 in the Number of
        Repetitions field indicates measurement request elements are executed once
        without repetition."
    ::= { dot11RMRequestEntry 4 }

dot11RMRqstIfIndex OBJECT-TYPE
    SYNTAX InterfaceIndex
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when requesting a measure-
        ment.
        Changes take effect when dot11RMRqstRowStatus is set to Active.

        The ifIndex on which this row of the RM Request is to be issued."
    ::= { dot11RMRequestEntry 5 }

dot11RMRqstType OBJECT-TYPE
```

```
    SYNTAX INTEGER {
        channelLoad(3),
        noiseHistogram(4),
        beacon(5),
        frame(6),
        staStatistics(7),
        lci(8),
        transmitStream(9),
        pause(255) }
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when requesting a measure-
        ment.
        Changes take effect when dot11RMRqstRowStatus is set to Active.

        This attribute indicates the measurement type of this RM request row."
    ::= { dot11RMRequestEntry 6 }

dot11RMRqstTargetAdd OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when requesting a measure-
        ment.
        Changes take effect when dot11RMRqstRowStatus is set to Active.

        The MAC address of STA for this row of RM Request is to be issued to. If
        this attribute matches the MAC address of the dot11RMRqstIfIndex, then
        measurement request is for this STA itself to carry out."
    ::= { dot11RMRequestEntry 7 }

dot11RMRqstTimeStamp OBJECT-TYPE
    SYNTAX TimeTicks
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by the SME when dot11RMRequestRowStatus is set to Active.

        This attribute indicates the sysUpTime Value the last time when the
        dot11RMRqstRowStatus is set to active or when this row is created the
        first time."
    ::= { dot11RMRequestEntry 8 }

dot11RMRqstChanNumber OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when requesting a measure-
        ment.
        Changes take effect when dot11RMRqstRowStatus is set to Active.

        The target STA channel number on which to perform the measurements indi-
        cated in this request. The Channel Number is only defined within the indi-
        cated Operating Class for this measurement request. This attribute is
        ignored if dot11RMRqstType = STA Statistics Request, LCI Request, Transmit
        Stream/Category Measurement, or measurement pause. However, even in that
        case, the manager should set this attribute to the current channel for
        this interface, so that the row can be set to active when ready with all
        attributes indicated."
    ::= { dot11RMRequestEntry 9 }

dot11RMRqstOperatingClass OBJECT-TYPE
    SYNTAX Unsigned32 (1..255)
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when requesting a measure-
        ment.
        Changes take effect when dot11RMRqstRowStatus is set to Active.

        This attribute indicates the channel set for this measurement request.
        Country, Operating Class and Channel Number together specify the channel
        frequency and spacing for this measurement request. Valid values of Oper-
        ating Class are shown in Annex E."
    REFERENCE "Annex E"
    ::= { dot11RMRequestEntry 10 }

dot11RMRqstRndInterval OBJECT-TYPE
    SYNTAX Unsigned32
    UNITS "TUs"
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when requesting a measure-
        ment.
        Changes take effect when dot11RMRqstRowStatus is set to Active.

        This attribute indicates the upper bound of the random delay to be used
        prior to making the measurement, expressed in units of TUs. See 10.11.3.
        This attribute is ignored if dot11RMRqstType = STA Statistics Request, LCI
        Request, Transmit Stream/Category Measurement or measurement pause."
    DEFVAL { 0 }
    ::= { dot11RMRequestEntry 11 }

dot11RMRqstDuration OBJECT-TYPE
    SYNTAX Unsigned32
    UNITS "TUs"
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when requesting a measure-
        ment.
        Changes take effect when dot11RMRqstRowStatus is set to Active.

        This attribute indicates the preferred or mandatory measurement duration
        for this Measurement Request. This attribute is ignored if dot11RMRqstType
        = LCI Request or measurement pause."
    DEFVAL { 0 }
    ::= { dot11RMRequestEntry 12 }

dot11RMRqstParallel OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when requesting a measure-
        ment.
        Changes take effect when dot11RMRqstRowStatus is set to Active.

        This attribute indicates the parallel bit for this Measurement Request
        element. This attribute, when false, indicates that the measurement is
        performed in sequence. This attribute, when true, indicates that this mea-
        surement should start at the same time as the measurement described by the
        next Measurement Request element in the next row if the next row indicates
        the same value for dot11RMRqstToken."
    DEFVAL { false }
    ::= { dot11RMRequestEntry 13 }

dot11RMRqstEnable OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when requesting a measure-
        ment.
        Changes take effect when dot11RMRqstRowStatus is set to Active.

        This attribute indicates the enable bit for this Measurement Request ele-
        ment."
    DEFVAL { false }
    ::= { dot11RMRequestEntry 14 }

dot11RMRqstRequest OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when requesting a measure-
        ment.
        Changes take effect when dot11RMRqstRowStatus is set to Active.

        This attribute indicates the request bit for this Measurement Request ele-
        ment. This attribute, when true, indicates that this STA accepts measure-
        ment requests from the target STA."
    DEFVAL { false }
    ::= { dot11RMRequestEntry 15 }

dot11RMRqstReport OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when requesting a measure-
        ment.
        Changes take effect when dot11RMRqstRowStatus is set to Active.

        This attribute indicates the report bit for this Measurement Request ele-
        ment. This attribute, when true, indicates that the target STA may enable
        autonomous measurement reports to the requesting STA."
    DEFVAL { false }
    ::= { dot11RMRequestEntry 16 }

dot11RMRqstDurationMandatory OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
```

```
      It is written by an external management entity when requesting a measure-
      ment.
      Changes take effect when dot11RMRqstRowStatus is set to Active.

      This attribute indicates the duration mandatory bit for this Measurement
      Request element. This attribute, when true, indicates that the indicated
      Measurement Duration is a mandatory duration for this measurement. This
      attribute, when false, indicates that the indicated Measurement Duration
      is a maximum duration for this measurement."
   DEFVAL { false }
   ::= { dot11RMRequestEntry 17 }

dot11RMRqstBeaconRqstMode OBJECT-TYPE
   SYNTAX INTEGER {
      passive(0),
      active(1),
      beaconTable(2) }
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity when requesting a measure-
      ment.
      Changes take effect when dot11RMRqstRowStatus is set to Active.

      This attribute indicates the Measurement Mode for this Beacon Request ele-
      ment. This attribute is only valid if the dot11RMRqstType is 5, indicating
      a beacon request, and is ignored otherwise."
   DEFVAL { 0 }
   ::= { dot11RMRequestEntry 18 }

dot11RMRqstBeaconRqstDetail OBJECT-TYPE
   SYNTAX INTEGER {
      noBody(0),
      fixedFieldsAndRequestedElements(1),
      allBody(2) }
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity when requesting a measure-
      ment.
      Changes take effect when dot11RMRqstRowStatus is set to Active.

      dot11RMRqstBeaconRqstDetail indicates the Reporting Detail for Beacon
      Request element. This attribute is only valid if the dot11RMRqstType is 5,
      indicating a beacon request, and is ignored otherwise."
   DEFVAL { 2 }
   ::= { dot11RMRequestEntry 19 }

dot11RMRqstFrameRqstType OBJECT-TYPE
   SYNTAX INTEGER { frameCountRep(1) }
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity when requesting a measure-
      ment.
      Changes take effect when dot11RMRqstRowStatus is set to Active.

      dot11RMRqstFrameRqstType indicates the Frame Request Type for Frame
      Request element. This attribute is only valid if the dot11RMRqstType is 6,
      indicating a frame request, and is ignored otherwise."
   DEFVAL { 1 }
```

```
    ::= { dot11RMRequestEntry 20 }

dot11RMRqstBssid OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when requesting a measure-
        ment.
        Changes take effect when dot11RMRqstRowStatus is set to Active.

        BSSID indicates the BSSID of the particular AP for which this measurement
        is requested. The BSSID is set to the wildcard BSSID when the measurement
        is to be performed on any AP(s) on the indicated channel. This attribute
        is only valid if the dot11RMRqstType is 5, indicating a beacon request,
        and is ignored otherwise."
    DEFVAL { 'FFFFFFFFFFFF'H }
    ::= { dot11RMRequestEntry 21 }

dot11RMRqstSSID OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(0..32))
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when requesting a measure-
        ment.
        Changes take effect when dot11RMRqstRowStatus is set to Active.

        This attribute indicates the SSID for the measurement. Zero length MIB
        element for SSID indicates the wildcard SSID. The SSID is set to the wild-
        card SSID when the measurement is to be performed on all ESSs/IBSSs on the
        indicated channel. This attribute is only valid if the dot11RMRqstType is
        5, indicating a beacon request, and is ignored otherwise."
    DEFVAL { ''H }
    ::= { dot11RMRequestEntry 22 }

dot11RMRqstBeaconReportingCondition OBJECT-TYPE
    SYNTAX INTEGER {
        afterEveryMeasurement(0),
        rcpiAboveAbsoluteThreshold(1),
        rcpiBelowAbsoluteThreshold(2),
        rsniAboveAbsoluteThreshold(3),
        rsniBelowAbsoluteThreshold(4),
        rcpiAboveOffsetThreshold(5),
        rcpiBelowOffsetThreshold(6),
        rsniAboveOffsetThreshold(7),
        rsniBelowOffsetThreshold(8),
        rcpiInBound(9),
        rsniInBound(10) }
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when requesting a measure-
        ment.
        Changes take effect when dot11RMRqstRowStatus is set to Active.

        This attribute indicates when the Beacon Measurement results are to be
        reported to the requesting STA. This attribute is only valid if the
        dot11RMRqstType is 5, indicating a beacon request, and is ignored other-
        wise."
    REFERENCE
    "IEEE 802.11, Table 8-66-Reporting Condition values for Beacon Request ele-
       ment"
    DEFVAL {0}
    ::= { dot11RMRequestEntry 23 }

dot11RMRqstBeaconThresholdOffset OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
       "This is a control variable.
       It is written by an external management entity when requesting a measure-
       ment.
       Changes take effect when dot11RMRqstRowStatus is set to Active.

       Threshold/Offset provides either the threshold value or the offset value
       to be used for conditional reporting. For indicated Reporting Conditions
       1-4, the integer range is (0..255). For indicated Reporting Conditions 5-
       10, the integer range is  (-127..+127). This attribute is only valid if
       the dot11RMRqstType is 5, indicating a beacon request, and is ignored oth-
       erwise."
    DEFVAL { 0 }
    ::= { dot11RMRequestEntry 24 }

dot11RMRqstSTAStatRqstGroupID OBJECT-TYPE
    SYNTAX INTEGER {
       dot11CountersTable(0),
       dot11MacStatistics(1),
       dot11QosCountersTableforUP0(2),
       dot11QosCountersTableforUP1(3),
       dot11QosCountersTableforUP2(4),
       dot11QosCountersTableforUP3(5),
       dot11QosCountersTableforUP4(6),
       dot11QosCountersTableforUP5(7),
       dot11QosCountersTableforUP6(8),
       dot11QosCountersTableforUP7(9),
       bSSAverageAccessDelays(10),
       dot11CountersGroup3Tablefor31(11),
       dot11CountersGroup3Tablefor32(12),
       dot11CountersGroup3Tablefor33(13),
       dot11CountersGroup3Tablefor34(14),
       dot11CountersGroup3Tablefor35(15),
       dot11RSNAStatsTable(16)}
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
       "This is a control variable.
       It is written by an external management entity when requesting a measure-
       ment.
       Changes take effect when dot11RMRqstRowStatus is set to Active.

       The attribute indicates the group identity for this Measurement Request
       element. This attribute is only valid if the dot11RMRqstType is 7, indi-
       cating a statistics request, and is ignored otherwise."
    DEFVAL { 0 }
    ::= { dot11RMRequestEntry 25 }

dot11RMRqstLCIRqstSubject OBJECT-TYPE
    SYNTAX INTEGER { local(0), remote(1) }
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
       "This is a control variable.
       It is written by an external management entity when requesting a measure-
```

```
      ment.
      Changes take effect when dot11RMRqstRowStatus is set to Active.

      The attribute indicates the subject of the LCI measurement request. This
      attribute is only valid if the dot11RMRqstType is 8, indicating an LCI
      request, and is ignored otherwise."
   DEFVAL { 0 }
   ::= { dot11RMRequestEntry 26 }

dot11RMRqstLCILatitudeResolution OBJECT-TYPE
   SYNTAX Unsigned32 (0..63)
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity when requesting a measure-
      ment.
      Changes take effect when dot11RMRqstRowStatus is set to Active.

      This attribute is 6 bits indicating the number of valid
      bits in the fixed-point value of Latitude of the LCI measurement request.
      This attribute is only valid if the dot11RMRqstType is 8, indicating an
      LCI request, and is ignored otherwise."
   ::= { dot11RMRequestEntry 27 }

dot11RMRqstLCILongitudeResolution OBJECT-TYPE
   SYNTAX Unsigned32 (0..63)
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity when requesting a measure-
      ment.
      Changes take effect when dot11RMRqstRowStatus is set to Active.

      This attribute is 6 bits indicating the number of valid bits in the fixed-
      point value of Longitude of the LCI measurement request. This attribute is
      only valid if the dot11RMRqstType is 8, indicating an LCI request, and is
      ignored otherwise."
   ::= { dot11RMRequestEntry 28 }

dot11RMRqstLCIAltitudeResolution OBJECT-TYPE
   SYNTAX Unsigned32 (0..63)
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity when requesting a measure-
      ment.
      Changes take effect when dot11RMRqstRowStatus is set to Active.

      This attribute is 6 bits indicating the number of valid
      bits in the fixed-point value of Altitude of the LCI measurement
      request. This attribute is only valid if the dot11RMRqstType is 8, indi-
      cating an LCI request, and is ignored otherwise."
   ::= { dot11RMRequestEntry 29 }

dot11RMRqstLCIAzimuthType OBJECT-TYPE
   SYNTAX INTEGER { frontSurfaceofSta(0), radioBeam(1) }
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity when requesting a measure-
      ment.
      Changes take effect when dot11RMRqstRowStatus is set to Active.

      This attribute indicates the azimuth reference for the LCI Azimuth mea-
      surement request. This attribute is only valid if the dot11RMRqstType is
      8, indicating an LCI request, and is ignored otherwise."
   DEFVAL{ 0 }
   ::= { dot11RMRequestEntry 30 }

dot11RMRqstLCIAzimuthResolution OBJECT-TYPE
   SYNTAX Unsigned32 (0..15)
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity when requesting a measure-
      ment.
      Changes take effect when dot11RMRqstRowStatus is set to Active.

      This attribute is 4 bits indicating the number of valid
      bits in the fixed-point value of Azimuth of the LCI Azimuth
      measurement request. This attribute is only valid if the dot11RMRqstType
      is 8, indicating an LCI request, and is ignored otherwise."
   ::= { dot11RMRequestEntry 31 }

dot11RMRqstPauseTime OBJECT-TYPE
   SYNTAX Unsigned32 (0..65535)
   UNITS "10 TUs"
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity when requesting a measure-
      ment.
      Changes take effect when dot11RMRqstRowStatus is set to Active.

      This attribute is a 16 bit unsigned integer number
      representing the time period for which measurements are
      suspended or paused. Measurement pause requests are used to
      provide time delays between the execution times of measurement
      request elements in a Measurement Request frame. This attribute is only
      valid if the dot11RMRqstType is 255, indicating an pause request, and is
      ignored otherwise."
   DEFVAL { 0 }
   ::= { dot11RMRequestEntry 32 }

dot11RMRqstTransmitStreamPeerQSTAAddress OBJECT-TYPE
   SYNTAX MacAddress
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity when requesting a measure-
      ment.
      Changes take effect when dot11RMRqstRowStatus is set to Active.

      This attribute indicates the peer STA address to be measured for a Trans-
      mit Stream/Category Measurement measurement. This attribute is only valid
      if the dot11RMRqstType is 9, indicating a transmit stream/category
      request, and is ignored otherwise."
   ::= { dot11RMRequestEntry 33 }

dot11RMRqstTransmitStreamTrafficIdentifier OBJECT-TYPE
   SYNTAX Unsigned32(0..16)
```

```
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when requesting a measure-
        ment.
        Changes take effect when dot11RMRqstRowStatus is set to Active.

        This attribute indicates the TC, or TS to be measured for a Transmit
        Stream/Category Measurement measurement. This attribute is only valid if
        the dot11RMRqstType is 9, indicating a transmit stream/category request,
        and is ignored otherwise."
    ::= { dot11RMRequestEntry 34 }

dot11RMRqstTransmitStreamBin0Range OBJECT-TYPE
    SYNTAX Unsigned32(1..255)
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when requesting a measure-
        ment.
        Changes take effect when dot11RMRqstRowStatus is set to Active.

        This attribute indicates the delay range for bin 0 of the transmit delay
        histogram. This attribute is only valid if the dot11RMRqstType is 9, indi-
        cating a transmit stream/category request, and is ignored otherwise."
    ::= { dot11RMRequestEntry 35 }

dot11RMRqstTrigdQoSAverageCondition OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when requesting a measure-
        ment.
        Changes take effect when dot11RMRqstRowStatus is set to Active.

        This attribute, when true, indicates a request for triggered reporting
        with trigger based on the number of discarded MSDUs reaching the
        dot11RMRqstTrigdQoSAverageThreshold when averaged over
        dot11RMRqstTrigdQoSMEasurementCount consecutive MSDUs. This attribute is
        only valid if the dot11RMRqstType is 9, indicating a transmit stream/cat-
        egory request, and is ignored otherwise."
    DEFVAL { false }
    ::= { dot11RMRequestEntry 36 }

dot11RMRqstTrigdQoSConsecutiveCondition OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when requesting a measure-
        ment.
        Changes take effect when dot11RMRqstRowStatus is set to Active.

        This attribute, when true, indicates a request for triggered reporting
        with trigger based on the consecutive number of MSDUs discarded reaching
        dot11RMRqstTrigdQoSConsecutiveThreshold. This attribute is only valid if
        the dot11RMRqstType is 9, indicating a transmit stream/category request,
        and is ignored otherwise."
    DEFVAL { false }
    ::= { dot11RMRequestEntry 37 }

dot11RMRqstTrigdQoSDelayCondition OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when requesting a measure-
        ment.
        Changes take effect when dot11RMRqstRowStatus is set to Active.

        This attribute, when true, indicates a request for triggered reporting
        with trigger based on the consecutive number of MSDUs that experience a
        transmit delay greater than dot11RMRqstTrigdQoSDelayThresholdRange reach-
        ing dot11RMRqstTrigdQoSDelayThreshold. This attribute is only valid if the
        dot11RMRqstType is 9, indicating a transmit stream/category request, and
        is ignored otherwise."
    DEFVAL { false }
    ::= { dot11RMRequestEntry 38 }

dot11RMRqstTrigdQoSAverageThreshold OBJECT-TYPE
    SYNTAX Unsigned32 (1..255)
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when requesting a measure-
        ment.
        Changes take effect when dot11RMRqstRowStatus is set to Active.

        This attribute indicates the trigger threshold for triggered Transmit
        Stream/Category Measurement based on average MSDUs discarded. Trigger
        occurs if the number of MSDUs discarded over the moving average number of
        transmitted MSDUs in dot11RMRqstTrigdQoSMeasurementCount reaches this
        threshold. This attribute is only valid if the dot11RMRqstType is 9, indi-
        cating a transmit stream/category request, and is ignored otherwise."
    DEFVAL { 10 }
    ::= { dot11RMRequestEntry 39 }

dot11RMRqstTrigdQoSConsecutiveThreshold OBJECT-TYPE
    SYNTAX Unsigned32 (1..255)
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when requesting a measure-
        ment.
        Changes take effect when dot11RMRqstRowStatus is set to Active.

        This attribute indicates the trigger threshold for triggered Transmit
        Stream/Category Measurement based on consecutive MSDUs discarded. Trigger
        occurs if the consecutive number of MSDUs discarded reaches this thresh-
        old. This attribute is only valid if the dot11RMRqstType is 9, indicating
        a transmit stream/category request, and is ignored otherwise."
    DEFVAL { 5 }
    ::= { dot11RMRequestEntry 40 }

dot11RMRqstTrigdQoSDelayThresholdRange OBJECT-TYPE
    SYNTAX Unsigned32 (0..3)
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when requesting a measure-
        ment.
        Changes take effect when dot11RMRqstRowStatus is set to Active.

        This attribute indicates the minimum transmit delay for delayed MSDU
        counts. Trigger occurs if the a consecutive number of MSDUs experience a
        transmit delay greater than or equal to the lower bound of the bin of the
        Transmit Delay Histogram given by the value of this attribute + 2, e.g. if
        this attribute is 1 the lower bound of bin 3. This attribute is only valid
        if the dot11RMRqstType is 9, indicating a transmit stream/category
        request, and is ignored otherwise."
    DEFVAL { 1 }
    ::= { dot11RMRequestEntry 41 }

dot11RMRqstTrigdQoSDelayThreshold OBJECT-TYPE
    SYNTAX Unsigned32 (1..255)
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when requesting a measure-
        ment.
        Changes take effect when dot11RMRqstRowStatus is set to Active.

        This attribute indicates the number of consecutive delayed MSDUs needed
        for trigger. Trigger occurs if the consecutive number of MSDUs that expe-
        rience a transmit delay greater than dot11RMRqstQoSDelayThresholdRange
        reaches this value. This attribute is only valid if the dot11RMRqstType is
        9, indicating a transmit stream/category request, and is ignored other-
        wise."
    DEFVAL { 20 }
    ::= { dot11RMRequestEntry 42 }

dot11RMRqstTrigdQoSMeasurementCount OBJECT-TYPE
    SYNTAX Unsigned32 (1..255)
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when requesting a measure-
        ment.
        Changes take effect when dot11RMRqstRowStatus is set to Active.

        This attribute indicates the number of MSDUs to be used as a moving aver-
        age count in the average error threshold and in determining the scope of
        the reported Transmit Stream/Category measurement in a triggered measure-
        ment report. This attribute is only valid if the dot11RMRqstType is 9,
        indicating a transmit stream/category request, and is ignored otherwise."
    DEFVAL { 100 }
    ::= { dot11RMRequestEntry 43 }

dot11RMRqstTrigdQoSTimeout OBJECT-TYPE
    SYNTAX Unsigned32 (1..255)
    UNITS "100 TUs"
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when requesting a measure-
        ment.
        Changes take effect when dot11RMRqstRowStatus is set to Active.

        This attribute indicates the timeout interval during which a measuring STA
        does not generate further triggered Transmit Stream/Category measurement
```

```
      reports after a trigger condition has been met and a report generated.
      This attribute is only valid if the dot11RMRqstType is 9, indicating a
      transmit stream/category request, and is ignored otherwise."
   DEFVAL { 20 }
   ::= { dot11RMRequestEntry 44 }

dot11RMRqstChannelLoadReportingCondition OBJECT-TYPE
   SYNTAX INTEGER {
      afterEveryMeasurement(0),
      chanLoadAboveReference(1),
      chanLoadBelowReference(2) }
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity when requesting a measure-
      ment.
      Changes take effect when dot11RMRqstRowStatus is set to Active.

      This attribute indicates when the Channel Load Measurement results are to
      be reported to the requesting STA. This attribute is only valid if the
      dot11RMRqstType is 3, indicating a channel load request, and is ignored
      otherwise."
   REFERENCE
      "IEEE 802.11, Table 8-61-Reporting Condition values for Channel Load
      Request element"
   DEFVAL {0}
   ::= { dot11RMRequestEntry 45 }

dot11RMRqstChannelLoadReference OBJECT-TYPE
   SYNTAX Unsigned32 (0..255)
   UNITS "1/255"
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity when requesting a measure-
      ment.
      Changes take effect when dot11RMRqstRowStatus is set to Active.

      This attribute indicates the channel load reporting condition reference
      value. The measured Channel Load is compared to this reference value and a
      report is issued if the reporting condition is satisfied. The reference
      value is in the same units as Channel Load and represents the fractional
      time of the measurement duration during which the STA determined the chan-
      nel to be busy. This attribute is only valid if the dot11RMRqstType is 3,
      indicating a channel load request, and is ignored otherwise."
   DEFVAL { 5 }
   ::= { dot11RMRequestEntry 46 }

dot11RMRqstNoiseHistogramReportingCondition OBJECT-TYPE
   SYNTAX INTEGER {
      afterEveryMeasurement(0),
      aNPIAboveReference(1),
      aNPIBelowReference(2) }
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity when requesting a measure-
      ment.
      Changes take effect when dot11RMRqstRowStatus is set to Active.

      This attribute indicates when the Noise Histogram Measurement results are
      to be reported to the requesting STA. This attribute is only valid if the
      dot11RMRqstType is 4, indicating a noise histogram request, and is ignored
      otherwise."
   REFERENCE
   "IEEE 802.11, Table 8-63-Reporting Condition for Noise Histogram Report"
   DEFVAL {0}
   ::= { dot11RMRequestEntry 47 }

dot11RMRqstAnpiReference OBJECT-TYPE
   SYNTAX Unsigned32 (0..255)
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity when requesting a measure-
      ment.
      Changes take effect when dot11RMRqstRowStatus is set to Active.

      This attribute indicates the noise histogram reporting condition ANPI ref-
      erence value. The measured ANPI is compared to this reference value and a
      report is issued if the indicated reporting condition is satisfied.
      ANPIval = Int[(ANPIpower in dBm + 110)*2], for ANPI in the range -110 dBm
      to 0 dBm. ANPIval = 220 for ANPI > 0 dBm. ANPIval = 255 when ANPI is not
      available. This attribute is only valid if the dot11RMRqstType is 4, indi-
      cating a noise histogram request, and is ignored otherwise."
   DEFVAL { 5 }
   ::= { dot11RMRequestEntry 48 }

dot11RMRqstAPChannelReport OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(0..255))
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity when requesting a measure-
      ment.
      Changes take effect when dot11RMRqstRowStatus is set to Active.

      This attribute indicates the specific channels to be used for the
      requested beacon measurements. The default value is null. Each octet indi-
      cates a different channel within the indicated Operating Class. This list
      of channels is the Channel List in the AP Channel Report element described
      in 8.4.2.38. This attribute is only valid if the dot11RMRqstType is 5,
      indicating a beacon request, and is ignored otherwise."
   DEFVAL { ''H }
   ::= { dot11RMRequestEntry 49 }

dot11RMRqstSTAStatPeerSTAAddress OBJECT-TYPE
   SYNTAX MacAddress
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity when requesting a measure-
      ment.
      Changes take effect when dot11RMRqstRowStatus is set to Active.

      This attribute indicates the peer STA address to be measured for a statis-
      tics request. This attribute is only valid if the dot11RMRqstType is 7,
      indicating a statistics request, and is ignored otherwise."
   ::= { dot11RMRequestEntry 50 }

dot11RMRqstFrameTransmitterAddress OBJECT-TYPE
   SYNTAX MacAddress
```

```
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when requesting a measure-
        ment.
        Changes take effect when dot11RMRqstRowStatus is set to Active.

        This attribute indicates the Transmitter Address (TA) of the frames to be
        counted in this frame request. This attribute is only valid if the
        dot11RMRqstType is 6, indicating a frame request, and is ignored other-
        wise."
    ::= { dot11RMRequestEntry 51 }

dot11RMRqstVendorSpecific OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(0..255))
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when requesting a measure-
        ment.
        Changes take effect when dot11RMRqstRowStatus is set to Active.

        This attribute provides an envelope for any optional vendor specific sub-
        elements which may be included in a measurement request element. The
        default value is null. This attribute is valid for all requests."
    DEFVAL { ''H }
    ::= { dot11RMRequestEntry 52 }

dot11RMRqstSTAStatTrigMeasCount OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This attribute indicates the number of MSDUs or MPDUs over which the
        trigger criterion is applied. This attribute is only valid if
        dot11RMRqstType is 7 (STA Statistics) and if the value of the attribute is
        not equal to 0."
    DEFVAL { 0 }
    ::= { dot11RMRequestEntry 53 }

dot11RMRqstSTAStatTrigTimeout OBJECT-TYPE
    SYNTAX Unsigned32(0..65535)
    UNITS "100 TUs"
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This attribute indicates the interval during which a measuring STA does
        not generate further triggered STA Statistics Reports after a trigger con-
        dition has been met. This attribute is only valid if dot11RMRqstType is 7
        (STA Statistics)."
    DEFVAL { 0 }
    ::= { dot11RMRequestEntry 54 }

dot11RMRqstSTAStatTrigCondition OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(2))
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This attribute indicates the trigger values used when requesting trig-
        gered STA Statistics
        reporting. The format of the STA Counter Trigger Condition field is shown
        in Figure 8-118."
    ::= { dot11RMRequestEntry 55 }

dot11RMRqstSTAStatTrigSTAFailedCntThresh OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This attribute indicates that a STA Statistics Report should be generated
        (triggered) when the dot11FailedCount value has increased more than the
        threshold value indicated here. The counter increase is measured over the
        last n MSDUs or MPDUs, where n is the value of
        dot11RMRqstSTAStatTrigMeasCount. This attribute is only valid if
        dot11RMRqstType is 7 (STA Statistics), and if
        dot11RMRqstSTAStatRqstGroupID is 0 (dot11CountersTable) and if the value
        of the attribute is not equal to 0."
    DEFVAL { 0 }
    ::= { dot11RMRequestEntry 56 }

dot11RMRqstSTAStatTrigSTAFCSErrCntThresh OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This attribute indicates that a STA Statistics Report should be generated
        (triggered) when the dot11FCSErrorCount value has increased more than the
        threshold value indicated here. The counter increase is measured over the
        last n MSDUs or MPDUs, where n is the value of
        dot11RMRqstSTAStatTrigMeasCount. This attribute is only valid if
        dot11RMRqstType is 7 (STA Statistics), and if
        dot11RMRqstSTAStatRqstGroupID is 0 (dot11CountersTable) and if the value
        of the attribute is not equal to 0."
    DEFVAL { 0 }
    ::= { dot11RMRequestEntry 57 }

dot11RMRqstSTAStatTrigSTAMultRetryCntThresh OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This attribute indicates that a STA Statistics Report should be generated
        (triggered) when the dot11MultipleRetryCount value has increased more than
        the threshold value indicated here. The counter increase is measured over
        the last n MSDUs or MPDUs, where n is the value of
        dot11RMRqstSTAStatTrigMeasCount. This attribute is only valid if
        dot11RMRqstType is 7 (STA Statistics), and if
        dot11RMRqstSTAStatRqstGroupID is 0 (dot11CountersTable) and if the value
        of the attribute is not equal to 0."
    DEFVAL { 0 }
    ::= { dot11RMRequestEntry 58 }

dot11RMRqstSTAStatTrigSTAFrameDupeCntThresh OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This attribute indicates that a STA Statistics Report should be generated
        (triggered) when the dot11FrameDuplicateCount value has increased more
        than the threshold value indicated here. The counter increase is measured
        over the last n MSDUs or MPDUs, where n is the value of
        dot11RMRqstSTAStatTrigMeasCount. This attribute is only valid if
        dot11RMRqstType is 7 (STA Statistics), and if
        dot11RMRqstSTAStatRqstGroupID is 0 (dot11CountersTable) and if the value
        of the attribute is not equal to 0."
    DEFVAL { 0 }
    ::= { dot11RMRequestEntry 59 }

dot11RMRqstSTAStatTrigSTARTSFailCntThresh OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This attribute indicates that a STA Statistics Report should be generated
        (triggered) when the dot11RTSFailureCount value has increased more than
        the threshold value indicated here. The counter increase is measured over
        the last n MSDUs or MPDUs, where n is the value of
        dot11RMRqstSTAStatTrigMeasCount. This attribute is only valid if
        dot11RMRqstType is 7 (STA Statistics), and if
        dot11RMRqstSTAStatRqstGroupID is 0 (dot11CountersTable) and if the value
        of the attribute is not equal to 0."
    DEFVAL { 0 }
    ::= { dot11RMRequestEntry 60 }

dot11RMRqstSTAStatTrigSTAAckFailCntThresh OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This attribute indicates that a STA Statistics Report should be generated
        (triggered) when the dot11ACKFailureCount value has increased more than
        the threshold value indicated here. The counter increase is measured over
        the last n MSDUs or MPDUs, where n is the value of
        dot11RMRqstSTAStatTrigMeasCount. This attribute is only valid if
        dot11RMRqstType is 7 (STA Statistics), and if
        dot11RMRqstSTAStatRqstGroupID is 0 (dot11CountersTable) and if the value
        of the attribute is not equal to 0."
    DEFVAL { 0 }
    ::= { dot11RMRequestEntry 61 }

dot11RMRqstSTAStatTrigSTARetryCntThresh OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This attribute indicates that a STA Statistics Report should be generated
        (triggered) when the dot11RetryCount value has increased more than the
        threshold value indicated here. The counter increase is measured over the
        last n MSDUs or MPDUs, where n is the value of
        dot11RMRqstSTAStatTrigMeasCount. This attribute is only valid if
        dot11RMRqstType is 7 (STA Statistics), and if
        dot11RMRqstSTAStatRqstGroupID is 0 (dot11CountersTable) and if the value
        of the attribute is not equal to 0."
    DEFVAL { 0 }
    ::= { dot11RMRequestEntry 62 }

dot11RMRqstSTAStatTrigQoSTrigCondition OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(2))
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This attribute indicates the trigger values used when requesting trig-
        gered QoS STA Statistics
        reporting. The format of the STA Counter Trigger Condition field is shown
        in Figure 8-120."
    ::= { dot11RMRequestEntry 63 }

dot11RMRqstSTAStatTrigQoSFailedCntThresh OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This attribute indicates that a STA Statistics Report should be generated
        (triggered) when the dot11QosFailedCount value has increased more than the
        threshold value indicated here. The counter increase is measured over the
        last n MSDUs or MPDUs, where n is the value of
        dot11RMRqstSTAStatTrigMeasCount. This attribute is only valid if
        dot11RMRqstType is 7 (STA Statistics), and if
        dot11RMRqstSTAStatRqstGroupID is 2 to 9 (dot11QosCountersTable) and if the
        value of the attribute is not equal to 0."
    DEFVAL { 0 }
    ::= { dot11RMRequestEntry 64 }

dot11RMRqstSTAStatTrigQoSRetryCntThresh OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This attribute indicates that a STA Statistics Report should be generated
        (triggered) when the dot11QosRetryCount value has increased more than the
        threshold value indicated here. The counter increase is measured over the
        last n MSDUs or MPDUs, where n is the value of
        dot11RMRqstSTAStatTrigMeasCount. This attribute is only valid if
        dot11RMRqstType is 7 (STA Statistics), and if
        dot11RMRqstSTAStatRqstGroupID is 2 to 9 (dot11QosCountersTable) and if the
        value of the attribute is not equal to 0."
    DEFVAL { 0 }
    ::= { dot11RMRequestEntry 65 }

dot11RMRqstSTAStatTrigQoSMultRetryCntThresh OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This attribute indicates that a STA Statistics Report should be generated
        (triggered) when the dot11QosMultipleRetryCount value has increased more
        than the threshold value indicated here. The counter increase is measured
        over the last n MSDUs or MPDUs, where n is the value of
        dot11RMRqstSTAStatTrigMeasCount. This attribute is only valid if
        dot11RMRqstType is 7 (STA Statistics), and if
        dot11RMRqstSTAStatRqstGroupID is 2 to 9 (dot11QosCountersTable) and if the
        value of the attribute is not equal to 0."
    DEFVAL { 0 }
    ::= { dot11RMRequestEntry 66 }

dot11RMRqstSTAStatTrigQoSFrameDupeCntThresh OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This attribute indicates that a STA Statistics Report should be generated
        (triggered) when the dot11QosFrameDuplicateCount value has increased more
        than the threshold value indicated here. The counter increase is measured
        over the last n MSDUs or MPDUs, where n is the value of
        dot11RMRqstSTAStatTrigMeasCount. This attribute is only valid if
        dot11RMRqstType is 7 (STA Statistics), and if
        dot11RMRqstSTAStatRqstGroupID is 2 to 9 (dot11QosCountersTable) and if the
        value of the attribute is not equal to 0."
    DEFVAL { 0 }
    ::= { dot11RMRequestEntry 67 }

dot11RMRqstSTAStatTrigQoSRTSFailCntThresh OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This attribute indicates that a STA Statistics Report should be generated
        (triggered) when the dot11QosRTSFailureCount value has increased more than
        the threshold value indicated here. The counter increase is measured over
        the last n MSDUs or MPDUs, where n is the value of
        dot11RMRqstSTAStatTrigMeasCount. This attribute is only valid if
        dot11RMRqstType is 7 (STA Statistics), and if
        dot11RMRqstSTAStatRqstGroupID is 2 to 9 (dot11QosCountersTable) and if the
        value of the attribute is not equal to 0."
    DEFVAL { 0 }
    ::= { dot11RMRequestEntry 68 }

dot11RMRqstSTAStatTrigQoSAckFailCntThresh OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This attribute indicates that a STA Statistics Report should be generated
        (triggered) when the dot11QosACKFailureCount value has increased more than
        the threshold value indicated here. The counter increase is measured over
        the last n MSDUs or MPDUs, where n is the value of
        dot11RMRqstSTAStatTrigMeasCount. This attribute is only valid if
        dot11RMRqstType is 7 (STA Statistics), and if
        dot11RMRqstSTAStatRqstGroupID is 2 to 9 (dot11QosCountersTable) and if the
        value of the attribute is not equal to 0."
    DEFVAL { 0 }
    ::= { dot11RMRequestEntry 69 }

dot11RMRqstSTAStatTrigQoSDiscardCntThresh OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This attribute indicates that a STA Statistics Report should be generated
        (triggered) when the dot11QosDiscardedFrameCount value has increased more
        than the threshold value indicated here. The counter increase is measured
        over the last n MSDUs or MPDUs, where n is the value of
        dot11RMRqstSTAStatTrigMeasCount. This attribute is only valid if
        dot11RMRqstType is 7 (STA Statistics), and if
        dot11RMRqstSTAStatRqstGroupID is 2 to 9 (dot11QosCountersTable) and if the
        value of the attribute is not equal to 0."
    DEFVAL { 0 }
    ::= { dot11RMRequestEntry 70 }

dot11RMRqstSTAStatTrigRsnaTrigCondition OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(2))
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This attribute indicates the trigger values used when requesting trig-
        gered RSNA STA Statistics
        reporting. The format of the STA Counter Trigger Condition field is shown
        in Figure 8-122."
    ::= { dot11RMRequestEntry 71 }

dot11RMRqstSTAStatTrigRsnaCMACICVErrCntThresh OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This attribute indicates that a STA Statistics Report should be generated
        (triggered) when the dot11RSNAStatsCMACICVErrors value has increased more
        than the threshold value indicated here. The counter increase is measured
        over the last n MSDUs or MPDUs, where n is the value of
        dot11RMRqstSTAStatTrigMeasCount. This attribute is only valid if
        dot11RMRqstType is 7 (STA Statistics), and if
        dot11RMRqstSTAStatRqstGroupID is 16 (dot11RSNAStatsTable) and if the value
        of the attribute is not equal to 0."
    DEFVAL { 0 }
    ::= { dot11RMRequestEntry 72 }

dot11RMRqstSTAStatTrigRsnaCMACReplayCntThresh OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This attribute indicates that a STA Statistics Report should be generated
        (triggered) when the dot11RSNAStatsCMACReplays value has increased more
        than the threshold value indicated here. The counter increase is measured
        over the last n MSDUs or MPDUs, where n is the value of
        dot11RMRqstSTAStatTrigMeasCount. This attribute is only valid if
        dot11RMRqstType is 7 (STA Statistics), and if
        dot11RMRqstSTAStatRqstGroupID is 16 (dot11RSNAStatsTable) and if the value
        of the attribute is not equal to 0."
    DEFVAL { 0 }
    ::= { dot11RMRequestEntry 73 }

dot11RMRqstSTAStatTrigRsnaRobustCCMPReplayCntThresh OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This attribute indicates that a STA Statistics Report should be generated
        (triggered) when the dot11RSNAStatsRobustMgmtCCMPReplays value has
        increased more than the threshold value indicated here. The counter
        increase is measured over the last n MSDUs or MPDUs, where n is the value
        of dot11RMRqstSTAStatTrigMeasCount. This attribute is only valid if
        dot11RMRqstType is 7 (STA Statistics), and if
        dot11RMRqstSTAStatRqstGroupID is 16 (dot11RSNAStatsTable) and if the value
        of the attribute is not equal to 0."
    DEFVAL { 0 }
    ::= { dot11RMRequestEntry 74 }

dot11RMRqstSTAStatTrigRsnaTKIPICVErrCntThresh OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This attribute indicates that a STA Statistics Report should be generated
        (triggered) when the dot11RSNAStatsTKIPICVErrors value has increased more
        than the threshold value indicated here. The counter increase is measured
        over the last n MSDUs or MPDUs, where n is the value of
        dot11RMRqstSTAStatTrigMeasCount. This attribute is only valid if
        dot11RMRqstType is 7 (STA Statistics), and if
        dot11RMRqstSTAStatRqstGroupID is 16 (dot11RSNAStatsTable) and if the value
        of the attribute is not equal to 0."
    DEFVAL { 0 }
    ::= { dot11RMRequestEntry 75 }

dot11RMRqstSTAStatTrigRsnaTKIPReplayCntThresh OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This attribute indicates that a STA Statistics Report should be generated
        (triggered) when the dot11RSNAStatsTKIPReplays value has increased more
        than the threshold value indicated here. The counter increase is measured
        over the last n MSDUs or MPDUs, where n is the value of
        dot11RMRqstSTAStatTrigMeasCount. This attribute is only valid if
        dot11RMRqstType is 7 (STA Statistics), and if
        dot11RMRqstSTAStatRqstGroupID is 16 (dot11RSNAStatsTable) and if the value
        of the attribute is not equal to 0."
    DEFVAL { 0 }
    ::= { dot11RMRequestEntry 76 }

dot11RMRqstSTAStatTrigRsnaCCMPDecryptErrCntThresh OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This attribute indicates that a STA Statistics Report should be generated
        (triggered) when the dot11RSNAStatsCCMPDecryptErrors value has increased
        more than the threshold value indicated here. The counter increase is mea-
        sured over the last n MSDUs or MPDUs, where n is the value of
        dot11RMRqstSTAStatTrigMeasCount. This attribute is only valid if
        dot11RMRqstType is 7 (STA Statistics), and if
        dot11RMRqstSTAStatRqstGroupID is 16 (dot11RSNAStatsTable) and if the value
        of the attribute is not equal to 0."
    DEFVAL { 0 }
    ::= { dot11RMRequestEntry 77 }

dot11RMRqstSTAStatTrigRsnaCCMPReplayCntThresh OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This attribute indicates that a STA Statistics Report should be generated
        (triggered) when the dot11RSNAStatsCCMPReplays value has increased more
        than the threshold value indicated here. The counter increase is measured
        over the last n MSDUs or MPDUs, where n is the value of
        dot11RMRqstSTAStatTrigMeasCount. This attribute is only valid if
        dot11RMRqstType is 7 (STA Statistics), and if
        dot11RMRqstSTAStatRqstGroupID is 16 (dot11RSNAStatsTable) and if the value
        of the attribute is not equal to 0."
    DEFVAL { 0 }
    ::= { dot11RMRequestEntry 78 }

-- **********************************************************************
-- * End of dot11RMRequest TABLE
-- **********************************************************************

-- ********************************************************************
-- * Radio Measurement Reports
-- * Report tables contain measurement reports received by this STA or
-- * results of measurements performed by this STA.
-- ********************************************************************
    dot11RMReport OBJECT IDENTIFIER ::= { dot11RadioMeasurement 2 }

-- ********************************************************************
-- * dot11ChannelLoadReport TABLE
-- ********************************************************************
dot11ChannelLoadReportTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11ChannelLoadReportEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
    "This table contains the current list of Channel Load reports that have been
        received by the MLME. The report tables are maintained as a FIFO to pre-
        serve freshness, thus the rows in this table can be deleted for memory
        constraints or other implementation constraints determined by the vendor.
        New rows have different RprtIndex values than those deleted within the
        range limitation of the index. One easy way is to monotonically increase
        RprtIndex for new reports being written in the table."
    ::= { dot11RMReport 1 }

dot11ChannelLoadReportEntry OBJECT-TYPE
    SYNTAX Dot11ChannelLoadReportEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry in the dot11ChannelLoadReportTable Indexed by
        dot11ChannelLoadRprtIndex."
    INDEX { dot11ChannelLoadRprtIndex }
    ::= { dot11ChannelLoadReportTable 1 }

Dot11ChannelLoadReportEntry ::=
    SEQUENCE {
        dot11ChannelLoadRprtIndex                         Unsigned32,
        dot11ChannelLoadRprtRqstToken                     OCTET STRING,
        dot11ChannelLoadRprtIfIndex                       InterfaceIndex,
        dot11ChannelLoadMeasuringSTAAddr                  MacAddress,
        dot11ChannelLoadRprtChanNumber                    Unsigned32,
        dot11ChannelLoadRprtOperatingClass                Unsigned32,
        dot11ChannelLoadRprtActualStartTime               TSFType,
        dot11ChannelLoadRprtMeasurementDuration           Unsigned32,
        dot11ChannelLoadRprtChannelLoad                   Unsigned32,
        dot11ChannelLoadRprtVendorSpecific                OCTET STRING,
        dot11ChannelLoadRprtMeasurementMode               INTEGER }

dot11ChannelLoadRprtIndex OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Index for Channel Load Report elements in dot11ChannelLoadReportTable,
        greater than 0."
    ::= { dot11ChannelLoadReportEntry 1 }

dot11ChannelLoadRprtRqstToken OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the request token that was indicated in the Mea-
        surement request that generated this measurement report. This should be an
        exact match to the original dot11RMRqstToken attribute. Note that there
        may be  multiple entries in the table that match this value since a single
        request may generate multiple measurement reports."
    ::= { dot11ChannelLoadReportEntry 2 }

dot11ChannelLoadRprtIfIndex OBJECT-TYPE
    SYNTAX InterfaceIndex
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The ifIndex of the interface over which this ChannelLoad Report was
        received."
    ::= { dot11ChannelLoadReportEntry 3 }
```

```
dot11ChannelLoadMeasuringSTAAddr OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The MAC address of the measuring STA for this row of the Channel Load
        report."
    ::= { dot11ChannelLoadReportEntry 4 }

dot11ChannelLoadRprtChanNumber OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the channel number used for this Channel Load
        Report. The Channel Number is only defined within the indicated Operating
        Class for this measurement report."
    ::= { dot11ChannelLoadReportEntry 5 }

dot11ChannelLoadRprtOperatingClass OBJECT-TYPE
    SYNTAX Unsigned32(1..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the channel set for this measurement report.
        Country, Operating Class and Channel Number together specify the channel
        frequency and spacing for this measurement report. Valid values of Operat-
        ing Class are shown in Annex E."
    REFERENCE "Annex E"
    ::= { dot11ChannelLoadReportEntry 6 }

dot11ChannelLoadRprtActualStartTime OBJECT-TYPE
    SYNTAX TSFType
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the TSF value at the time when the
        measurement started."
    ::= { dot11ChannelLoadReportEntry 7 }

dot11ChannelLoadRprtMeasurementDuration OBJECT-TYPE
    SYNTAX Unsigned32
    UNITS "TUs"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the duration over which the ChannelLoad Report
        was measured."
    ::= { dot11ChannelLoadReportEntry 8 }
```

```
dot11ChannelLoadRprtChannelLoad OBJECT-TYPE
    SYNTAX Unsigned32(0..255)
    UNITS "1/255"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        Channel Load contains the fractional duration over which the measuring STA
        determined the channel to be busy during the measurement duration."
    REFERENCE "8.4.2.24.5"
    ::= { dot11ChannelLoadReportEntry 9 }

dot11ChannelLoadRprtVendorSpecific OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(0..255))
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute provides an envelope for any optional vendor specific sub-
        elements which may be included in a measurement report element. The
        default value is null."
    DEFVAL { ''H }
    ::= { dot11ChannelLoadReportEntry 10 }

dot11ChannelLoadRprtMeasurementMode OBJECT-TYPE
SYNTAX INTEGER {
    success(0),
    incapableBit(1),
    refusedBit(2)
    }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the outcome status for the measurement request
        which generated this measurement report; status is indicated using the
        following reason codes: 1 indicates this STA is incapable of generating
        the report, 2 indicates this STA is refusing to generate the report, 0
        indicates the STA successfully carried out the measurement request."
    DEFVAL { 0 }
    ::= { dot11ChannelLoadReportEntry 11 }

-- **********************************************************************
-- * End of dot11ChannelLoadReport TABLE
-- **********************************************************************

-- **********************************************************************
-- * dot11NoiseHistogramReport TABLE
-- **********************************************************************
dot11NoiseHistogramReportTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11NoiseHistogramReportEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
    "This table contains the current list of Noise Histogram reports that have
        been received by the MLME. The report tables are maintained as a FIFO to
        preserve freshness, thus the rows in this table can be deleted for memory
        constraints or other implementation constraints determined by the vendor.
        New rows have different RprtIndex values than those deleted within the
        range limitation of the index. One easy way is to monotonically increase
        RprtIndex for new reports being written in the table."
    ::= { dot11RMReport 2 }

dot11NoiseHistogramReportEntry OBJECT-TYPE
    SYNTAX Dot11NoiseHistogramReportEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry in the dot11NoiseHistogramReportTable Indexed by
        dot11NoiseHistogramRprtIndex."
    INDEX { dot11NoiseHistogramRprtIndex }
    ::= { dot11NoiseHistogramReportTable 1 }

Dot11NoiseHistogramReportEntry ::=
    SEQUENCE {
        dot11NoiseHistogramRprtIndex                    Unsigned32,
        dot11NoiseHistogramRprtRqstToken                OCTET STRING,
        dot11NoiseHistogramRprtIfIndex                  InterfaceIndex,
        dot11NoiseHistogramMeasuringSTAAddr             MacAddress,
        dot11NoiseHistogramRprtChanNumber               Unsigned32,
        dot11NoiseHistogramRprtOperatingClass           Unsigned32,
        dot11NoiseHistogramRprtActualStartTime          TSFType,
        dot11NoiseHistogramRprtMeasurementDuration      Unsigned32,
        dot11NoiseHistogramRprtAntennaID                Unsigned32,
        dot11NoiseHistogramRprtANPI                     Unsigned32,
        dot11NoiseHistogramRprtIPIDensity0              Unsigned32,
        dot11NoiseHistogramRprtIPIDensity1              Unsigned32,
        dot11NoiseHistogramRprtIPIDensity2              Unsigned32,
        dot11NoiseHistogramRprtIPIDensity3              Unsigned32,
        dot11NoiseHistogramRprtIPIDensity4              Unsigned32,
        dot11NoiseHistogramRprtIPIDensity5              Unsigned32,
        dot11NoiseHistogramRprtIPIDensity6              Unsigned32,
        dot11NoiseHistogramRprtIPIDensity7              Unsigned32,
        dot11NoiseHistogramRprtIPIDensity8              Unsigned32,
        dot11NoiseHistogramRprtIPIDensity9              Unsigned32,
        dot11NoiseHistogramRprtIPIDensity10             Unsigned32,
        dot11NoiseHistogramRprtVendorSpecific           OCTET STRING,
        dot11NoiseHistogramRprtMeasurementMode          INTEGER}

dot11NoiseHistogramRprtIndex OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Index for Noise Histogram elements in dot11NoiseHistogramReportTable,
        greater than 0."
    ::= { dot11NoiseHistogramReportEntry 1 }

dot11NoiseHistogramRprtRqstToken OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the request token that was indicated in the mea-
        surement request that generated this measurement report. This should be an
        exact match to the original dot11RMRqstToken attribute. Note that there
        may be multiple entries in the table that match this value since a single
        request may generate multiple measurement reports."
    ::= { dot11NoiseHistogramReportEntry 2 }

dot11NoiseHistogramRprtIfIndex OBJECT-TYPE
    SYNTAX InterfaceIndex
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The ifIndex of the interface over which this Noise Histogram Report was
        received. "
    ::= { dot11NoiseHistogramReportEntry 3 }

dot11NoiseHistogramMeasuringSTAAddr OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The MAC address of the measuring STA for this row of Noise Histogram
        report."
    ::= { dot11NoiseHistogramReportEntry 4 }

dot11NoiseHistogramRprtChanNumber OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the channel number used for this Noise Histogram
        Report. The Channel Number is only defined within the indicated Operating
        Class for this measurement report."
    ::= { dot11NoiseHistogramReportEntry 5 }

dot11NoiseHistogramRprtOperatingClass OBJECT-TYPE
    SYNTAX Unsigned32(1..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the channel set for this measurement report.
        Country, Operating Class and Channel Number together specify the channel
        frequency and spacing for this measurement report. Valid values of Operat-
        ing Class are shown in Annex E."
    REFERENCE "Annex E"
    ::= { dot11NoiseHistogramReportEntry 6 }

dot11NoiseHistogramRprtActualStartTime OBJECT-TYPE
    SYNTAX TSFType
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the TSF value at the time when the
        measurement started."
    ::= { dot11NoiseHistogramReportEntry 7 }

dot11NoiseHistogramRprtMeasurementDuration OBJECT-TYPE
    SYNTAX Unsigned32
    UNITS "TUs"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the duration over which the Noise Histogram
        Report was measured."
    ::= { dot11NoiseHistogramReportEntry 8 }

dot11NoiseHistogramRprtAntennaID OBJECT-TYPE
    SYNTAX Unsigned32(0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the identifying number for the antenna used for
        this measurement. The value 0 indicates that the antenna identifier is
        unknown. The value 255 indicates that the measurement was made with multi-
        ple antennas or that the antenna ID is unknown. The value 1 is used for a
        STA with only one antenna. STAs with more than one antenna assign Antenna
        IDs to each antenna as consecutive, ascending numbers. Each  Antenna ID
        number represents a unique antenna characterized by a fixed relative posi-
        tion, a fixed relative direction, and a peak gain for that position and
        direction."
    ::= { dot11NoiseHistogramReportEntry 9 }

dot11NoiseHistogramRprtANPI OBJECT-TYPE
    SYNTAX Unsigned32(0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the Average Noise Power Indicator (ANPI) for this
        Noise Histogram measurement. The ANPI value represents the average noise
        plus interference power on the measured channel at the antenna connector
        during the measurement duration. To calculate ANPI, the STA measures and
        uses IPI in the indicated channel when NAV is equal to 0 (when virtual CS
        mechanism indicates idle channel) except during frame transmission or
        reception."
    ::= { dot11NoiseHistogramReportEntry 10 }

dot11NoiseHistogramRprtIPIDensity0 OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the measured IPI density for non-IEEE-802.11 sig-
        nals with measured power satisfying the condition: Power <= -92dBm."
    ::= { dot11NoiseHistogramReportEntry 11 }

dot11NoiseHistogramRprtIPIDensity1 OBJECT-TYPE
```

```
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the measured IPI density for non-IEEE-802.11 sig-
        nals with measured power satisfying the condition: -92dBm < Power <= -
        89dBm."
    ::= { dot11NoiseHistogramReportEntry 12 }

dot11NoiseHistogramRprtIPIDensity2 OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the measured IPI density for non-IEEE-802.11 sig-
        nals with measured power satisfying the condition: -89dBm < Power <= -
        86dBm."
    ::= { dot11NoiseHistogramReportEntry 13 }

dot11NoiseHistogramRprtIPIDensity3 OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the measured IPI density for non-IEEE-802.11 sig-
        nals with measured power satisfying the condition: -86dBm < Power <= -
        83dBm."
    ::= { dot11NoiseHistogramReportEntry 14 }

dot11NoiseHistogramRprtIPIDensity4 OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the measured IPI density for non-IEEE-802.11 sig-
        nals with measured power satisfying the condition: -83dBm < Power <= -
        80dBm."
    ::= { dot11NoiseHistogramReportEntry 15 }

dot11NoiseHistogramRprtIPIDensity5 OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the measured IPI density for non-IEEE-802.11 sig-
        nals with measured power satisfying the condition: -80dBm < Power <= -
        75dBm."
    ::= { dot11NoiseHistogramReportEntry 16 }
```

```
dot11NoiseHistogramRprtIPIDensity6 OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      This attribute indicates the measured IPI density for non-IEEE-802.11 sig-
      nals with measured power satisfying the condition:
      -75dBm < Power <= -70dBm."
   ::= { dot11NoiseHistogramReportEntry 17 }

dot11NoiseHistogramRprtIPIDensity7 OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      This attribute indicates the measured IPI density for non-IEEE-802.11 sig-
      nals with measured power satisfying the condition:
      -70dBm < Power <= -65dBm."
   ::= { dot11NoiseHistogramReportEntry 18 }

dot11NoiseHistogramRprtIPIDensity8 OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      This attribute indicates the measured IPI density for non-IEEE-802.11 sig-
      nals with measured power satisfying the condition:
      -65dBm < Power <= -60dBm."
   ::= { dot11NoiseHistogramReportEntry 19 }

dot11NoiseHistogramRprtIPIDensity9 OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      This attribute indicates the measured IPI density for non-IEEE-802.11 sig-
      nals with measured power satisfying the condition:
      -60dBm < Power <= -55dBm."
   ::= { dot11NoiseHistogramReportEntry 20 }

dot11NoiseHistogramRprtIPIDensity10 OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      This attribute indicates the measured IPI density for non-IEEE-802.11 sig-
      nals with measured power satisfying the condition:
      -55dBm < Power."
   ::= { dot11NoiseHistogramReportEntry 21 }
```

```
dot11NoiseHistogramRprtVendorSpecific OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(0..255))
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      This attribute provides an envelope for any optional vendor specific sub-
      elements which may be included in a measurement report element. The
      default value is null."
   DEFVAL { ''H }
   ::= { dot11NoiseHistogramReportEntry 22 }

dot11NoiseHistogramRprtMeasurementMode OBJECT-TYPE
   SYNTAX INTEGER {
      success(0),
      incapableBit(1),
      refusedBit(2) }
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      This attribute indicates the outcome status for the measurement request
      which generated this measurement report; status is indicated using the
      following reason codes: 1 indicates this STA is incapable of generating
      the report, 2 indicates this STA is refusing to generate the report, 0
      indicates the STA successfully carried out the measurement request."
   DEFVAL { 0 }
   ::= { dot11NoiseHistogramReportEntry 23 }

-- **********************************************************************
-- * End of dot11NoiseHistogramReport TABLE
-- **********************************************************************

-- **********************************************************************
-- * dot11BeaconReport TABLE
-- **********************************************************************
dot11BeaconReportTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11BeaconReportEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
   "This table contains the current list of Beacon reports that have been
      received by the MLME. The report tables are maintained as FIFO to preserve
      freshness, thus the rows in this table can be deleted for memory con-
      straints or other implementation constraints determined by the vendor. New
      rows have different RprtIndex values than those deleted within the range
      limitation of the index. One easy way is to monotonically increase RprtIn-
      dex for new reports being written in the table."
   ::= { dot11RMReport 3 }

dot11BeaconReportEntry OBJECT-TYPE
   SYNTAX Dot11BeaconReportEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "An entry in the dot11BeaconReportTable Indexed by dot11BeaconRprtIndex."
   INDEX { dot11BeaconRprtIndex }
   ::= { dot11BeaconReportTable 1 }
```

```
Dot11BeaconReportEntry ::=
    SEQUENCE {
        dot11BeaconRprtIndex                             Unsigned32,
        dot11BeaconRprtRqstToken                         OCTET STRING,
        dot11BeaconRprtIfIndex                           InterfaceIndex,
        dot11BeaconMeasuringSTAAddr                      MacAddress,
        dot11BeaconRprtChanNumber                        Unsigned32,
        dot11BeaconRprtOperatingClass                    Unsigned32,
        dot11BeaconRprtActualStartTime                   TSFType,
        dot11BeaconRprtMeasurementDuration               Unsigned32,
        dot11BeaconRprtPhyType                           INTEGER,
        dot11BeaconRprtReportedFrameType                 INTEGER,
        dot11BeaconRprtRCPI                              Unsigned32,
        dot11BeaconRprtRSNI                              Unsigned32,
        dot11BeaconRprtBSSID                             MacAddress,
        dot11BeaconRprtAntennaID                         Unsigned32,
        dot11BeaconRprtParentTSF                         TSFType,
        dot11BeaconRprtReportedFrameBody                 OCTET STRING,
        dot11BeaconRprtVendorSpecific                    OCTET STRING,
        dot11BeaconRprtMeasurementMode                   INTEGER}

dot11BeaconRprtIndex OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Index for Beacon Reports in dot11BeaconReportTable, greater than 0."
    ::= { dot11BeaconReportEntry 1 }

dot11BeaconRprtRqstToken OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the request token that was indicated in the mea-
        surement request that generated this measurement report. This should be an
        exact match to the original dot11RMRqstToken attribute. Note that there
        may be multiple entries in the table that match this value since a single
        request may generate multiple measurement reports."
    ::= { dot11BeaconReportEntry 2 }

dot11BeaconRprtIfIndex OBJECT-TYPE
    SYNTAX InterfaceIndex
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The ifIndex of the interface over which this Beacon Report was received."
    ::= { dot11BeaconReportEntry 3 }

dot11BeaconMeasuringSTAAddr OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The MAC address of the measuring STA for this row of Beacon report."
    ::= { dot11BeaconReportEntry 4 }

dot11BeaconRprtChanNumber OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the channel number used for this Beacon Report.
        The Channel Number is only defined within the indicated Operating Class
        for this measurement report."
    ::= { dot11BeaconReportEntry 5 }

dot11BeaconRprtOperatingClass OBJECT-TYPE
    SYNTAX Unsigned32(1..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the channel set for this measurement report.
        Country, Operating Class and Channel Number together specify the channel
        frequency and spacing for this measurement report. Valid values of Operat-
        ing Class are shown in Annex E."
    REFERENCE "Annex E"
    ::= { dot11BeaconReportEntry 6 }

dot11BeaconRprtActualStartTime OBJECT-TYPE
    SYNTAX TSFType
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the TSF value at the time when the measurement
        started."
    ::= { dot11BeaconReportEntry 7 }

dot11BeaconRprtMeasurementDuration OBJECT-TYPE
    SYNTAX Unsigned32
    UNITS "TUs"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the duration over which the Beacon Report was
        measured."
    ::= { dot11BeaconReportEntry 8 }

dot11BeaconRprtPhyType OBJECT-TYPE
    SYNTAX INTEGER {
        fhss(1),
        dsss(2),
        irbaseband(3),
        ofdm(4),
        hrdsss(5),
        erp(6),
        ht(7) }
    UNITS "dot11PHYType"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the PHY Type for this row of Beacon Report."
    ::= { dot11BeaconReportEntry 9 }

dot11BeaconRprtReportedFrameType OBJECT-TYPE
    SYNTAX INTEGER { beaconOrProbeResponse(0), measurementPilot(1) }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the frame type reported in
        dot11BeaconRprtReportedFrameBody"
    ::= { dot11BeaconReportEntry 10 }

dot11BeaconRprtRCPI OBJECT-TYPE
    SYNTAX Unsigned32(0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the received channel power of the beacon or probe
        response frame in dBm, as defined in the RCPI measurement subclause for
        the indicated PHY Type.
        RCPIval = Int[(RCPIpower in dBm + 110)*2], for RCPI in the range -110 dBm
        to 0 dBm. RCPIval = 220 for RCPI > 0 dBm. RCPIval = 255 when RCPI is not
        available."
    ::= { dot11BeaconReportEntry 11 }

dot11BeaconRprtRSNI OBJECT-TYPE
    SYNTAX Unsigned32(0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the received signal to noise ratio of the beacon
        or probe response frame in dB. RSNI is the received signal to noise plus
        interference ratio derived from the measured RCPI for the received frame
        and from the measured ANPI for the channel used to receive the frame. RSNI
        is calculated by the ratio of  the received signal power (RCPI - ANPI)
        over the noise plus interference power (ANPI) where
        RSNI = [(ratio(dB) + 10) * 2], for ratios in the range -10dB to +118dB."
    ::= { dot11BeaconReportEntry 12 }

dot11BeaconRprtBSSID OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the BSSID of the beacon for this row of Beacon
        Report."
    ::= { dot11BeaconReportEntry 13 }

dot11BeaconRprtAntennaID OBJECT-TYPE
    SYNTAX Unsigned32(0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the identifying number for the antenna used for
        this measurement. The value 0 indicates that the antenna identifier is
        unknown. The value 255 indicates that this measurement was made with mul-
        tiple antennas. The value 1 is used for a STA with only one antenna. STAs
        with more than one antenna assign Antenna IDs to each antenna as consecu-
        tive, ascending numbers. Each Antenna ID number represents a unique
        antenna characterized by a fixed relative position, a fixed relative
        direction, and a peak gain for that position and direction."
    ::= { dot11BeaconReportEntry 14 }

dot11BeaconRprtParentTSF OBJECT-TYPE
    SYNTAX TSFType
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the TSF value of the serving measuring STA's TSF
        value at the time the measuring STA received the beacon or probe response
        frame."
    ::= { dot11BeaconReportEntry 15 }

dot11BeaconRprtReportedFrameBody OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(0..100))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the fixed fields and elements from the frame body
        of the Beacon, Measurement Pilot or Probe Response frame being received.
        All reported TIM elements are truncated to 4 octets."
    ::= { dot11BeaconReportEntry 16 }

dot11BeaconRprtVendorSpecific OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(0..255))
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute provides an envelope for any optional vendor specific sub-
        elements which may be included in a measurement report element. The
        default value is null."
    DEFVAL { ''H }
    ::= { dot11BeaconReportEntry 17 }

dot11BeaconRprtMeasurementMode OBJECT-TYPE
    SYNTAX INTEGER {
        success(0),
        incapableBit(1),
        refusedBit(2) }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the outcome status for the measurement request
        which generated this measurement report; status is indicated using the
        following reason codes: 1 indicates this STA is incapable of generating
        the report, 2 indicates this STA is refusing to generate the report, 0
        indicates the STA successfully carried out the measurement request."
    DEFVAL { 0 }
    ::= { dot11BeaconReportEntry 18 }

-- **********************************************************************
-- * End of dot11BeaconReport TABLE
-- **********************************************************************

-- **********************************************************************
-- * dot11FrameReport TABLE
-- **********************************************************************
dot11FrameReportTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11FrameReportEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
    "This table contains the current list of Frame reports that have been
        received by the MLME. The report tables are maintained as a FIFO to pre-
        serve freshness, thus the rows in this table can be deleted for memory
        constraints or other implementation constraints determined by the vendor.
        New rows have different RprtIndex values than those deleted within the
        range limitation of the index. One easy way is to monotonically increase
        RprtIndex for new reports being written in the table."
    ::= { dot11RMReport 4 }

dot11FrameReportEntry OBJECT-TYPE
    SYNTAX Dot11FrameReportEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry in the dot11FrameReportTable Indexed by dot11FrameRprtIndex."
    INDEX { dot11FrameRprtIndex }
    ::= { dot11FrameReportTable 1 }

Dot11FrameReportEntry ::=
    SEQUENCE {
        dot11FrameRprtIndex                       Unsigned32,
        dot11FrameRprtIfIndex                     InterfaceIndex,
        dot11FrameRprtRqstToken                   Unsigned32,
        dot11FrameRprtChanNumber                  Unsigned32,
        dot11FrameRprtOperatingClass              Unsigned32,
        dot11FrameRprtActualStartTime             TSFType,
        dot11FrameRprtMeasurementDuration         Unsigned32,
        dot11FrameRprtTransmitSTAAddress          MacAddress,
        dot11FrameRprtBSSID                       MacAddress,
        dot11FrameRprtPhyType                     INTEGER,
        dot11FrameRprtAvgRCPI                     Unsigned32,
        dot11FrameRprtLastRSNI                    Unsigned32,
        dot11FrameRprtLastRCPI                    Unsigned32,
        dot11FrameRprtAntennaID                   Unsigned32,
        dot11FrameRprtNumberFrames                Unsigned32,
        dot11FrameRprtVendorSpecific              OCTET STRING,
    dot11FrameRptMeasurementMode               INTEGER}

dot11FrameRprtIndex OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "Index for Frame Report elements in dot11FrameReportTable, greater than
      0."
   ::= { dot11FrameReportEntry 1 }

dot11FrameRprtIfIndex OBJECT-TYPE
   SYNTAX InterfaceIndex
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      The ifIndex of the interface over which this Frame Report was received."
   ::= { dot11FrameReportEntry 2 }

dot11FrameRprtRqstToken OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

   INDEX for Frame Request elements in dot11FrameRequestTable that corresponds
      to this row of frame report. Since a single frame request can generate
      multiple rows in the frame report table, one per BSSID, this
      dot11FrameRprtRqstToken indicates which request this particular row indi-
      cates. If this row of report is received without a particular request,
      this attribute should be 0"
   ::= { dot11FrameReportEntry 3 }

dot11FrameRprtChanNumber OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      This attribute indicates the channel number used for this Frame Report.
      The Channel Number is only defined within the indicated Operating Class
      for this measurement report."
   ::= { dot11FrameReportEntry 4 }

dot11FrameRprtOperatingClass OBJECT-TYPE
   SYNTAX Unsigned32(1..255)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      This attribute indicates the channel set for this measurement report.
      Country, Operating Class and Channel Number together specify the channel
      frequency and spacing for this measurement report. Valid values of Operat-
      ing Class are shown in Annex E."
   REFERENCE "Annex E"
```

```
    ::= { dot11FrameReportEntry 5 }

dot11FrameRprtActualStartTime OBJECT-TYPE
    SYNTAX TSFType
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the TSF value at the time when measurement
        started."
    ::= { dot11FrameReportEntry 6 }

dot11FrameRprtMeasurementDuration OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the duration over which the Frame Report was mea-
        sured, expressed in units of TUs."
    ::= { dot11FrameReportEntry 7 }

dot11FrameRprtTransmitSTAAddress OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The MAC address of STA for this row of Frame report that it has been
        received from."
    ::= { dot11FrameReportEntry 8 }

dot11FrameRprtBSSID OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the BSSID of the STA that transmitted this
        frame."
    ::= { dot11FrameReportEntry 9 }

dot11FrameRprtPhyType OBJECT-TYPE
    SYNTAX INTEGER {
        fhss(1),
        dsss(2),
        irbaseband(3),
        ofdm(4),
        hrdsss(5),
        erp(6),
        ht(7) }
    UNITS "dot11PHYType"
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
```

```
      It is written by the SME when a measurement report is completed.

      This attribute indicates the PHY used for frame reception in this row of
      the frame report."
   ::= { dot11FrameReportEntry 10 }

dot11FrameRprtAvgRCPI OBJECT-TYPE
   SYNTAX Unsigned32(0..255)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      This attribute indicates the average value for the received channel power
      of all the frames received and counted in this Frame Report Entry, in dBm,
      as defined in the RCPI measurement subclause for the indicated PHY Type.
      RCPIval = Int[(RCPIpower in dBm + 110)*2], for RCPI in the range -110 dBm
      to 0 dBm. RCPIval = 220 for RCPI > 0 dBm. RCPIval = 255 when RCPI is not
      available."
   ::= { dot11FrameReportEntry 11 }

dot11FrameRprtLastRSNI OBJECT-TYPE
   SYNTAX Unsigned32(0..255)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      This attribute indicates the received signal to noise ratio of the
      received frame in dBm. RSNI is the received signal to noise plus interfer-
      ence ratio derived from the RCPI for the received frame and from the most
      recent ANPI value measured on the channel used to receive the frame. RSNI
      may be calculated by the ratio of  the received signal power (RCPI - ANPI)
      over the noise plus interference power (ANPI) where RSNI = [(ratio(dB) +
      10) * 2], for  ratios in the range -10dB to +118dB. Other measurement
      techniques are allowed."
   ::= { dot11FrameReportEntry 12 }

dot11FrameRprtLastRCPI OBJECT-TYPE
   SYNTAX Unsigned32(0..255)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      This attribute indicates the received channel power of the most recently
      measured frame in this Frame Report entry, in dBm, as defined in the RCPI
      measurement subclause for the indicated PHY Type.
      RCPIval = Int[(RCPIpower in dBm + 110)*2], for RCPI in the range -110 dBm
      to 0 dBm. RCPIval = 220 for RCPI > 0 dBm. RCPIval = 255 when RCPI is not
      available."
   ::= { dot11FrameReportEntry 13 }

dot11FrameRprtAntennaID OBJECT-TYPE
   SYNTAX Unsigned32(0..255)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.
      This attribute indicates the identifying number for the antenna used for
      this measurement. The value 0 indicates that the antenna identifier is
      unknown. The value 255 indicates that this measurement was made with mul-
      tiple antennas. The value 1 is used for a STA with only one antenna. STAs
      with more than one antenna assign Antenna IDs to each antenna as consecu-
      tive, ascending numbers. Each  Antenna ID number represents a unique
      antenna characterized by a fixed relative position, a fixed relative
      direction, and a peak gain for that position and direction."
   ::= { dot11FrameReportEntry 14 }

dot11FrameRprtNumberFrames  OBJECT-TYPE
   SYNTAX Unsigned32(0..65535)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      This attribute indicates the number of received frames in the Measurement
      Report frame for this row of the Frame Report."
   ::= { dot11FrameReportEntry 15 }

dot11FrameRprtVendorSpecific OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(0..255))
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      This attribute provides an envelope for any optional vendor specific sub-
      elements which may be included in a measurement report element. The
      default value is null."
   DEFVAL { ''H }
   ::= { dot11FrameReportEntry 16 }

dot11FrameRptMeasurementMode OBJECT-TYPE
   SYNTAX INTEGER {
      success(0),
      incapableBit(1),
      refusedBit(2) }
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      This attribute indicates the outcome status for the measurement request
      which generated this measurement report; status is indicated using the
      following reason codes: 1 indicates this STA is incapable of generating
      the report, 2 indicates this STA is refusing to generate the report, 0
      indicates the STA successfully carried out the measurement request."
   DEFVAL { 0 }
   ::= { dot11FrameReportEntry 17 }

-- ********************************************************************
-- * End of dot11FrameReport TABLE
-- ********************************************************************

-- ********************************************************************
-- * dot11STAStatisticsReport TABLE
-- ********************************************************************
dot11STAStatisticsReportTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11STAStatisticsReportEntry
```

```
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
    "This table contains the current list of STA Statistics reports that have
        been received by the MLME. The report tables are maintained as a FIFO to
        preserve freshness, thus the rows in this table can be deleted for memory
        constraints or other implementation constraints determined by the vendor.
        New rows have different RprtIndex values than those deleted within the
        range limitation of the index. One easy way is to monotonically increase
        RprtIndex for new reports being written in the table."
    ::= { dot11RMReport 5 }

dot11STAStatisticsReportEntry OBJECT-TYPE
    SYNTAX Dot11STAStatisticsReportEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry in the dot11STAStatisticsReportTable Indexed by
        dot11STAStatisticsReportIndex."
    INDEX { dot11STAStatisticsReportIndex }
    ::= { dot11STAStatisticsReportTable 1 }

Dot11STAStatisticsReportEntry ::=
    SEQUENCE {
        dot11STAStatisticsReportIndex                        Unsigned32,
        dot11STAStatisticsReportToken                        OCTET STRING,
        dot11STAStatisticsIfIndex                            InterfaceIndex,
        dot11STAStatisticsSTAAddress                         MacAddress,
        dot11STAStatisticsMeasurementDuration                Unsigned32,
        dot11STAStatisticsGroupID                            INTEGER,
        dot11STAStatisticsTransmittedFragmentCount           Counter32,
        dot11STAStatisticsGroupTransmittedFrameCount         Counter32,
        dot11STAStatisticsFailedCount                        Counter32,
        dot11STAStatisticsRetryCount                         Counter32,
        dot11STAStatisticsMultipleRetryCount                 Counter32,
        dot11STAStatisticsFrameDuplicateCount                Counter32,
        dot11STAStatisticsRTSSuccessCount                    Counter32,
        dot11STAStatisticsRTSFailureCount                    Counter32,
        dot11STAStatisticsACKFailureCount                    Counter32,
        dot11STAStatisticsQosTransmittedFragmentCount        Counter32,
        dot11STAStatisticsQosFailedCount                     Counter32,
        dot11STAStatisticsQosRetryCount                      Counter32,
        dot11STAStatisticsQosMultipleRetryCount              Counter32,
        dot11STAStatisticsQosFrameDuplicateCount             Counter32,
        dot11STAStatisticsQosRTSSuccessCount                 Counter32,
        dot11STAStatisticsQosRTSFailureCount                 Counter32,
        dot11STAStatisticsQosACKFailureCount                 Counter32,
        dot11STAStatisticsQosReceivedFragmentCount           Counter32,
        dot11STAStatisticsQosTransmittedFrameCount           Counter32,
        dot11STAStatisticsQosDiscardedFrameCount             Counter32,
        dot11STAStatisticsQosMPDUsReceivedCount              Counter32,
        dot11STAStatisticsQosRetriesReceivedCount            Counter32,
        dot11STAStatisticsReceivedFragmentCount              Counter32,
        dot11STAStatisticsGroupReceivedFrameCount            Counter32,
        dot11STAStatisticsFCSErrorCount                      Counter32,
        dot11STAStatisticsTransmittedFrameCount              Counter32,
        dot11STAStatisticsAPAverageAccessDelay               Unsigned32,
        dot11STAStatisticsAverageAccessDelayBestEffort       Unsigned32,
        dot11STAStatisticsAverageAccessDelayBackground       Unsigned32,
        dot11STAStatisticsAverageAccessDelayVideo            Unsigned32,
        dot11STAStatisticsAverageAccessDelayVoice            Unsigned32,
        dot11STAStatisticsStationCount                       Unsigned32,
        dot11STAStatisticsChannelUtilization                 Unsigned32,
        dot11STAStatisticsVendorSpecific                     OCTET STRING,
       dot11STAStatisticsRprtMeasurementMode               INTEGER,
       dot11STAStatisticsRSNAStatsCMACICVErrors            Counter32,
       dot11STAStatisticsRSNAStatsCMACReplays              Counter32,
       dot11STAStatisticsRSNAStatsRobustMgmtCCMPReplays    Counter32,
       dot11STAStatisticsRSNAStatsTKIPICVErrors            Counter32,
       dot11STAStatisticsRSNAStatsTKIPReplays              Counter32,
       dot11STAStatisticsRSNAStatsCCMPDecryptErrors        Counter32,
       dot11STAStatisticsRSNAStatsCCMPReplays              Counter32,
       dot11STAStatisticsReportingReasonSTACounters        OCTET STRING,
       dot11STAStatisticsReportingReasonQosCounters        OCTET STRING,
       dot11STAStatisticsReportingReasonRsnaCounters       OCTET STRING,
       dot11STAStatisticsTransmittedAMSDUCount             Counter32,
       dot11STAStatisticsFailedAMSDUCount                  Counter32,
       dot11STAStatisticsRetryAMSDUCount                   Counter32,
       dot11STAStatisticsMultipleRetryAMSDUCount           Counter32,
       dot11STAStatisticsTransmittedOctetsInAMSDUCount      Counter64,
       dot11STAStatisticsAMSDUAckFailureCount              Counter32,
       dot11STAStatisticsReceivedAMSDUCount                Counter32,
       dot11STAStatisticsReceivedOctetsInAMSDUCount        Counter64,
       dot11STAStatisticsTransmittedAMPDUCount             Counter32,
       dot11STAStatisticsTransmittedMPDUsInAMPDUCount      Counter32,
       dot11STAStatisticsTransmittedOctetsInAMPDUCount     Counter64,
       dot11STAStatisticsAMPDUReceivedCount                Counter32,
       dot11STAStatisticsMPDUInReceivedAMPDUCount          Counter32,
       dot11STAStatisticsReceivedOctetsInAMPDUCount        Counter64,
       dot11STAStatisticsAMPDUDelimiterCRCErrorCount       Counter32,
       dot11STAStatisticsImplicitBARFailureCount           Counter32,
       dot11STAStatisticsExplicitBARFailureCount           Counter32,
       dot11STAStatisticsChannelWidthSwitchCount           Counter32,
       dot11STAStatisticsTwentyMHzFrameTransmittedCount    Counter32,
       dot11STAStatisticsFortyMHzFrameTransmittedCount     Counter32,
       dot11STAStatisticsTwentyMHzFrameReceivedCount       Counter32,
       dot11STAStatisticsFortyMHzFrameReceivedCount        Counter32,
       dot11STAStatisticsPSMPUTTGrantDuration              Counter32,
       dot11STAStatisticsPSMPUTTUsedDuration               Counter32,
       dot11STAStatisticsGrantedRDGUsedCount               Counter32,
       dot11STAStatisticsGrantedRDGUnusedCount             Counter32,
       dot11STAStatisticsTransmittedFramesInGrantedRDGCount Counter32,
       dot11STAStatisticsTransmittedOctetsInGrantedRDGCount Counter64,
       dot11STAStatisticsDualCTSSuccessCount               Counter32,
       dot11STAStatisticsDualCTSFailureCount               Counter32,
       dot11STAStatisticsRTSLSIGSuccessCount               Counter32,
       dot11STAStatisticsRTSLSIGFailureCount               Counter32,
       dot11STAStatisticsBeamformingFrameCount             Counter32,
       dot11STAStatisticsSTBCCTSSuccessCount               Counter32,
       dot11STAStatisticsSTBCCTSFailureCount               Counter32,
       dot11STAStatisticsnonSTBCCTSSuccessCount            Counter32,
       dot11STAStatisticsnonSTBCCTSFailureCount            Counter32
    }

dot11STAStatisticsReportIndex OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
       "Index for STA Statistics Report elements in
       dot11STAStatisticsReportTable, greater than 0."
    ::= { dot11STAStatisticsReportEntry 1 }

dot11STAStatisticsReportToken OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
```

```
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the token that was indicated in the measurement
        request that generated this measurement report. This should be an exact
        match to the original dot11RMRqstToken attribute. Note that there may be
        multiple entries in the table that match this value since a single request
        may generate multiple measurement reports."
    ::= { dot11STAStatisticsReportEntry 2 }

dot11STAStatisticsIfIndex OBJECT-TYPE
    SYNTAX InterfaceIndex
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        Identifies the Interface that this row of STA Statistics Report has been
        received on"
    ::= { dot11STAStatisticsReportEntry 3 }

dot11STAStatisticsSTAAddress OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The MAC address of the STA that returned this STA Statistics Report."
    ::= { dot11STAStatisticsReportEntry 4 }

dot11STAStatisticsMeasurementDuration OBJECT-TYPE
    SYNTAX Unsigned32
    UNITS "TUs"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the duration over which the STA Statistics was
        measured. The value 0 for this attribute indicates that the reported sta-
        tistics are a current snapshot of the statistics variables. A nonzero
        value for this attribute indicates that the reported statistics contain
        the difference in the corresponding statistics variables over the indi-
        cated duration."
    ::= { dot11STAStatisticsReportEntry 5 }

dot11STAStatisticsGroupID OBJECT-TYPE
    SYNTAX INTEGER {
        dot11CountersTable(0),
        dot11MacStatistics(1),
        dot11QosCountersTableforUP0(2),
        dot11QosCountersTableforUP1(3),
        dot11QosCountersTableforUP2(4),
        dot11QosCountersTableforUP3(5),
        dot11QosCountersTableforUP4(6),
        dot11QosCountersTableforUP5(7),
        dot11QosCountersTableforUP6(8),
        dot11QosCountersTableforUP7(9),
        bSSAverageAccessDelays(10),
        dot11CountersGroup3Tablefor31(11),
        dot11CountersGroup3Tablefor32(12),
        dot11CountersGroup3Tablefor33(13),
        dot11CountersGroup3Tablefor34(14),
        dot11CountersGroup3Tablefor35(15),
        dot11RSNAStatsTable(16) }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the value of dot11RMRqstSTAStatRqstGroupID
        returned from the STA in this STA Statistics Report."
    DEFVAL { 0 }
    ::= { dot11STAStatisticsReportEntry 6 }

dot11STAStatisticsTransmittedFragmentCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        If dot11STAStatisticsMeasurementDuration is 0, this attribute indicates
        the value of dot11TransmittedFragmentCount returned from the STA in this
        STA Statistics Report. If dot11STAStatisticsMeasurementDuration indicates
        a nonzero value, this attribute indicates the difference in the referenced
        dot11 variable over the indicated duration. This attribute is only valid
        if the dot11STAStatisticsGroupID is 0, and is ignored otherwise."
    ::= { dot11STAStatisticsReportEntry 7 }

dot11STAStatisticsGroupTransmittedFrameCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        If dot11STAStatisticsMeasurementDuration is 0, this attribute indicates
        the value of dot11GroupTransmittedFrameCount returned from the STA in this
        STA Statistics Report. If dot11STAStatisticsMeasurementDuration indicates
        a nonzero value, this attribute indicates the difference in the referenced
        dot11 variable over the indicated duration. This attribute is only valid
        if the dot11STAStatisticsGroupID is 0, and is ignored otherwise."
    ::= { dot11STAStatisticsReportEntry 8 }

dot11STAStatisticsFailedCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        If dot11STAStatisticsMeasurementDuration is 0, this attribute indicates
        the value of dot11FailedCount returned from the STA in this STA Statistics
        Report. If dot11STAStatisticsMeasurementDuration indicates a nonzero
        value, this attribute indicates the difference in the referenced dot11
        variable over the indicated duration. This attribute is only valid if the
        dot11STAStatisticsGroupID is 0, and is ignored otherwise."
    ::= { dot11STAStatisticsReportEntry 9 }
```

```
dot11STAStatisticsRetryCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      If dot11STAStatisticsMeasurementDuration is 0, this attribute indicates
      the value of dot11RetryCount returned from the STA in this STA Statistics
      Report. If dot11STAStatisticsMeasurementDuration indicates a nonzero
      value, this attribute indicates the difference in the referenced dot11
      variable over the indicated duration. This attribute is only valid if the
      dot11STAStatisticsGroupID is 1, and is ignored otherwise."
   ::= { dot11STAStatisticsReportEntry 10 }

dot11STAStatisticsMultipleRetryCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      If dot11STAStatisticsMeasurementDuration is 0, this attribute indicates
      the value of dot11MultipleRetryCount returned from the STA in this STA
      Statistics Report. If dot11STAStatisticsMeasurementDuration indicates a
      nonzero value, this attribute indicates the difference in the referenced
      dot11 variable over the indicated duration. This attribute is only valid
      if the dot11STAStatisticsGroupID is 1, and is ignored otherwise."
   ::= { dot11STAStatisticsReportEntry 11 }

dot11STAStatisticsFrameDuplicateCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      If dot11STAStatisticsMeasurementDuration is 0, this attribute indicates
      the value of dot11FrameDuplicateCount returned from the STA in this STA
      Statistics Report. If dot11STAStatisticsMeasurementDuration indicates a
      nonzero value, this attribute indicates the difference in the referenced
      dot11 variable over the indicated duration. This attribute is only valid
      if the dot11STAStatisticsGroupID is 1, and is ignored otherwise."
   ::= { dot11STAStatisticsReportEntry 12 }

dot11STAStatisticsRTSSuccessCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      If dot11STAStatisticsMeasurementDuration is 0, this attribute indicates
      the value of dot11RTSSuccessCount returned from the STA in this STA Sta-
      tistics Report. If dot11STAStatisticsMeasurementDuration indicates a non-
      zero value, this attribute indicates the difference in the referenced
      dot11 variable over the indicated duration. This attribute is only valid
      if the dot11STAStatisticsGroupID is 1, and is ignored otherwise."
   ::= { dot11STAStatisticsReportEntry 13 }
```

```
dot11STAStatisticsRTSFailureCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      If dot11STAStatisticsMeasurementDuration is 0, this attribute indicates
      the value of dot11RTSFailureCount returned from the STA in this STA Sta-
      tistics Report. If dot11STAStatisticsMeasurementDuration indicates a non-
      zero value, this attribute indicates the difference in the referenced
      dot11 variable over the indicated duration. This attribute is only valid
      if the dot11STAStatisticsGroupID is 1, and is ignored otherwise."
   ::= { dot11STAStatisticsReportEntry 14 }

dot11STAStatisticsACKFailureCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      If dot11STAStatisticsMeasurementDuration is 0, this attribute indicates
      the value of dot11ACKFailureCount returned from the STA in this STA Sta-
      tistics Report. If dot11STAStatisticsMeasurementDuration indicates a non-
      zero value, this attribute indicates the difference in the referenced
      dot11 variable over the indicated duration. This attribute is only valid
      if the dot11STAStatisticsGroupID is 1, and is ignored otherwise."
   ::= { dot11STAStatisticsReportEntry 15 }

dot11STAStatisticsQosTransmittedFragmentCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      If dot11STAStatisticsMeasurementDuration is 0, this attribute indicates
      the value of dot11QosTransmittedFragmentCount returned from the STA in
      this STA Statistics Report. If dot11STAStatisticsMeasurementDuration indi-
      cates a nonzero value, this attribute indicates the difference in the ref-
      erenced dot11 variable over the indicated duration. This attribute is only
      valid if the dot11STAStatisticsGroupID is 2-9, and is ignored otherwise."
   ::= { dot11STAStatisticsReportEntry 16 }

dot11STAStatisticsQosFailedCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      If dot11STAStatisticsMeasurementDuration is 0, this attribute indicates
      the value of dot11QosFailedCount returned from the STA in this STA Statis-
      tics Report. If dot11STAStatisticsMeasurementDuration indicates a nonzero
      value, this attribute indicates the difference in the referenced dot11
      variable over the indicated duration. This attribute is only valid if the
      dot11STAStatisticsGroupID is 2-9, and is ignored otherwise."
   ::= { dot11STAStatisticsReportEntry 17 }
```

```
dot11STAStatisticsQosRetryCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      If dot11STAStatisticsMeasurementDuration is 0, this attribute indicates
      the value of dot11QosRetryCount returned from the STA in this STA Statis-
      tics Report. If dot11STAStatisticsMeasurementDuration indicates a nonzero
      value, this attribute indicates the difference in the referenced dot11
      variable over the indicated duration. This attribute is only valid if the
      dot11STAStatisticsGroupID is 2-9, and is ignored otherwise."
   ::= { dot11STAStatisticsReportEntry 18 }

dot11STAStatisticsQosMultipleRetryCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      If dot11STAStatisticsMeasurementDuration is 0, this attribute indicates
      the value of dot11QosMultipleRetryCount returned from the STA in this STA
      Statistics Report. If dot11STAStatisticsMeasurementDuration indicates a
      nonzero value, this attribute indicates the difference in the referenced
      dot11 variable over the indicated duration. This attribute is only valid
      if the dot11STAStatisticsGroupID is 2-9, and is ignored otherwise."
   ::= { dot11STAStatisticsReportEntry 19 }

dot11STAStatisticsQosFrameDuplicateCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      If dot11STAStatisticsMeasurementDuration is 0, this attribute indicates
      the value of dot11QosFrameDuplicateCount returned from the STA in this STA
      Statistics Report. If dot11STAStatisticsMeasurementDuration indicates a
      nonzero value, this attribute indicates the difference in the referenced
      dot11 variable over the indicated duration. This attribute is only valid
      if the dot11STAStatisticsGroupID is 2-9, and is ignored otherwise."
   ::= { dot11STAStatisticsReportEntry 20 }

dot11STAStatisticsQosRTSSuccessCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      If dot11STAStatisticsMeasurementDuration is 0, this attribute indicates
      the value of dot11QosRTSSuccessCount returned from the STA in this STA
      Statistics Report. If dot11STAStatisticsMeasurementDuration indicates a
      nonzero value, this attribute indicates the difference in the referenced
      dot11 variable over the indicated duration. This attribute is only valid
      if the dot11STAStatisticsGroupID is 2-9, and is ignored otherwise."
   ::= { dot11STAStatisticsReportEntry 21 }
```

```
dot11STAStatisticsQosRTSFailureCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        If dot11STAStatisticsMeasurementDuration is 0, this attribute indicates
        the value of dot11QosRTSFailureCount returned from the STA in this STA
        Statistics Report. If dot11STAStatisticsMeasurementDuration indicates a
        nonzero value, this attribute indicates the difference in the referenced
        dot11 variable over the indicated duration. This attribute is only valid
        if the dot11STAStatisticsGroupID is 2-9, and is ignored otherwise."
    ::= { dot11STAStatisticsReportEntry 22 }

dot11STAStatisticsQosACKFailureCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        If dot11STAStatisticsMeasurementDuration is 0, this attribute indicates
        the value of dot11QosACKFailureCount returned from the STA in this STA
        Statistics Report. If dot11STAStatisticsMeasurementDuration indicates a
        nonzero value, this attribute indicates the difference in the referenced
        dot11 variable over the indicated duration. This attribute is only valid
        if the dot11STAStatisticsGroupID is 2-9, and is ignored otherwise."
    ::= { dot11STAStatisticsReportEntry 23 }

dot11STAStatisticsQosReceivedFragmentCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        If dot11STAStatisticsMeasurementDuration is 0, this attribute indicates
        the value of dot11QosReceivedFragmentCount returned from the STA in this
        STA Statistics Report. If dot11STAStatisticsMeasurementDuration indicates
        a nonzero value, this attribute indicates the difference in the referenced
        dot11 variable over the indicated duration. This attribute is only valid
        if the dot11STAStatisticsGroupID is 2-9, and is ignored otherwise."
    ::= { dot11STAStatisticsReportEntry 24 }

dot11STAStatisticsQosTransmittedFrameCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        If dot11STAStatisticsMeasurementDuration is 0, this attribute indicates
        the value of dot11QosTransmittedFrameCount returned from the STA in this
        STA Statistics Report. If dot11STAStatisticsMeasurementDuration indicates
        a nonzero value, this attribute indicates the difference in the referenced
        dot11 variable over the indicated duration. This attribute is only valid
        if the dot11STAStatisticsGroupID is 2-9, and is ignored otherwise."
    ::= { dot11STAStatisticsReportEntry 25 }
```

```
dot11STAStatisticsQosDiscardedFrameCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      If dot11STAStatisticsMeasurementDuration is 0, this attribute indicates
      the value of dot11QosDiscardedFrameCount returned from the STA in this STA
      Statistics Report. If dot11STAStatisticsMeasurementDuration indicates a
      nonzero value, this attribute indicates the difference in the referenced
      dot11 variable over the indicated duration. This attribute is only valid
      if the dot11STAStatisticsGroupID is 2-9, and is ignored otherwise."
   ::= { dot11STAStatisticsReportEntry 26 }

dot11STAStatisticsQosMPDUsReceivedCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      If dot11STAStatisticsMeasurementDuration is 0, this attribute indicates
      the value of dot11QosMPDUsReceivedCount returned from the STA in this STA
      Statistics Report. If dot11STAStatisticsMeasurementDuration indicates a
      nonzero value, this attribute indicates the difference in the referenced
      dot11 variable over the indicated duration. This attribute is only valid
      if the dot11STAStatisticsGroupID is 2-9, and is ignored otherwise."
   ::= { dot11STAStatisticsReportEntry 27 }

dot11STAStatisticsQosRetriesReceivedCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      If dot11STAStatisticsMeasurementDuration is 0, this attribute indicates
      the value of dot11QosRetriesReceivedCount returned from the STA in this
      STA Statistics Report. If dot11STAStatisticsMeasurementDuration indicates
      a nonzero value, this attribute indicates the difference in the referenced
      dot11 variable over the indicated duration. This attribute is only valid
      if the dot11STAStatisticsGroupID is 2-9, and is ignored otherwise."
   ::= { dot11STAStatisticsReportEntry 28 }

dot11STAStatisticsReceivedFragmentCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      If dot11STAStatisticsMeasurementDuration is 0, this attribute indicates
      the value of dot11ReceivedFragmentCount returned from the STA in this STA
      Statistics Report. If dot11STAStatisticsMeasurementDuration indicates a
      nonzero value, this attribute indicates the difference in the referenced
      dot11 variable over the indicated duration. This attribute is only valid
      if the dot11STAStatisticsGroupID is 0, and is ignored otherwise."
   ::= { dot11STAStatisticsReportEntry 29 }
```

```
dot11STAStatisticsGroupReceivedFrameCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      If dot11STAStatisticsMeasurementDuration is 0, this attribute indicates
      the value of dot11GroupReceivedFrameCount returned from the STA in this
      STA Statistics Report. If dot11STAStatisticsMeasurementDuration indicates
      a nonzero value, this attribute indicates the difference in the referenced
      dot11 variable over the indicated duration. This attribute is only valid
      if the dot11STAStatisticsGroupID is 0, and is ignored otherwise."
   ::= { dot11STAStatisticsReportEntry 30 }

dot11STAStatisticsFCSErrorCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      If dot11STAStatisticsMeasurementDuration is 0, this attribute indicates
      the value of dot11FCSErrorCount returned from the STA in this STA Statis-
      tics Report. If dot11STAStatisticsMeasurementDuration indicates a nonzero
      value, this attribute indicates the difference in the referenced dot11
      variable over the indicated duration. This attribute is only valid if the
      dot11STAStatisticsGroupID is 0, and is ignored otherwise."
   ::= { dot11STAStatisticsReportEntry 31 }

dot11STAStatisticsTransmittedFrameCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      If dot11STAStatisticsMeasurementDuration is 0, this attribute indicates
      the value of dot11TransmittedFrameCount returned from the STA in this STA
      Statistics Report. If dot11STAStatisticsMeasurementDuration indicates a
      nonzero value, this attribute indicates the difference in the referenced
      dot11 variable over the indicated duration. This attribute is only valid
      if the dot11STAStatisticsGroupID is 0, and is ignored otherwise."
   ::= { dot11STAStatisticsReportEntry 32 }

dot11STAStatisticsAPAverageAccessDelay OBJECT-TYPE
   SYNTAX Unsigned32 (0..255)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      If dot11STAStatisticsMeasurementDuration is 0, this attribute indicates
      the value of the AP Average Access Delay (AAD)  returned from the STA in
      this STA Statistics Report. If dot11STAStatisticsMeasurementDuration indi-
      cates a nonzero value, this attribute indicates the difference in the ref-
      erenced access delay value over the indicated duration. This attribute is
      only valid if the dot11STAStatisticsGroupID is 10, and is ignored other-
      wise."
   REFERENCE
```

```
      "IEEE 802.11 8.4.2.41"
   ::= { dot11STAStatisticsReportEntry 33 }

dot11STAStatisticsAverageAccessDelayBestEffort OBJECT-TYPE
   SYNTAX Unsigned32 (0..255)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      If dot11STAStatisticsMeasurementDuration is 0, this attribute indicates
      the value of the Average Access Delay (AAD) for the Best Effort Access
      Category returned from the STA in this STA Statistics Report. If
      dot11STAStatisticsMeasurementDuration indicates a nonzero value, this
      attribute indicates the difference in the referenced access delay value
      over the indicated duration. This attribute is only valid if the
      dot11STAStatisticsGroupID is 10, and is ignored otherwise."
   REFERENCE
      "IEEE 802.11 8.4.2.46"
   ::= { dot11STAStatisticsReportEntry 34 }

dot11STAStatisticsAverageAccessDelayBackground OBJECT-TYPE
   SYNTAX Unsigned32 (0..255)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      If dot11STAStatisticsMeasurementDuration is 0, this attribute indicates
      the value of the Average Access Delay (AAD) for the Background Access Cat-
      egory returned from the STA in this STA Statistics Report. If
      dot11STAStatisticsMeasurementDuration indicates a nonzero value, this
      attribute indicates the difference in the referenced access delay value
      over the indicated duration. This attribute is only valid if the
      dot11STAStatisticsGroupID is 10, and is ignored otherwise."
   REFERENCE
      "IEEE 802.11 8.4.2.46"
   ::= { dot11STAStatisticsReportEntry 35 }

dot11STAStatisticsAverageAccessDelayVideo OBJECT-TYPE
   SYNTAX Unsigned32 (0..255)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      If dot11STAStatisticsMeasurementDuration is 0, this attribute indicates
      the value of the Average Access Delay (AAD) for the Video Access Category
      returned from the STA in this STA Statistics Report. If
      dot11STAStatisticsMeasurementDuration indicates a nonzero value, this
      attribute indicates the difference in the referenced access delay value
      over the indicated duration. This attribute is only valid if the
      dot11STAStatisticsGroupID is 10, and is ignored otherwise."
   REFERENCE
      "IEEE 802.11 8.4.2.46"
   ::= { dot11STAStatisticsReportEntry 36 }

dot11STAStatisticsAverageAccessDelayVoice OBJECT-TYPE
   SYNTAX Unsigned32 (0..255)
   MAX-ACCESS read-only
   STATUS current
```

```
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        If dot11STAStatisticsMeasurementDuration is 0, this attribute indicates
        the value of the Average Access Delay (AAD) for the Voice Access Category
        returned from the STA in this STA Statistics Report. If
        dot11STAStatisticsMeasurementDuration indicates a nonzero value, this
        attribute indicates the difference in the referenced access delay value
        over the indicated duration. This attribute is only valid if the
        dot11STAStatisticsGroupID is 10, and is ignored otherwise."
    REFERENCE
        "IEEE 802.11 8.4.2.46"
    ::= { dot11STAStatisticsReportEntry 37 }

dot11STAStatisticsStationCount OBJECT-TYPE
    SYNTAX Unsigned32 (0..65535)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        If dot11STAStatisticsMeasurementDuration is 0, this attribute indicates
        the value of dot11AssociatedStationCount returned from the STA in this STA
        Statistics Report. If dot11STAStatisticsMeasurementDuration indicates a
        nonzero value, this attribute indicates the difference in the referenced
        dot11 variable over the indicated duration. This attribute is only valid
        if the dot11STAStatisticsGroupID is 10, and is ignored otherwise."
    ::= { dot11STAStatisticsReportEntry 38 }

dot11STAStatisticsChannelUtilization OBJECT-TYPE
    SYNTAX Unsigned32 (0..255)
    UNITS "1/255"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        If dot11STAStatisticsMeasurementDuration is 0, this attribute indicates
        the value of the Channel Utilization returned from the STA in this STA
        Statistics Report. If dot11STAStatisticsMeasurementDuration indicates a
        nonzero value, this attribute indicates the difference in the Channel Uti-
        lization value over the indicated duration. The Channel Utilization is the
        time fraction during which the AP sensed the channel busy. This attribute
        is only valid if the dot11STAStatisticsGroupID is 10, and is ignored oth-
        erwise."
    REFERENCE
        "IEEE 802.11 8.4.2.30"
    ::= { dot11STAStatisticsReportEntry 39 }

dot11STAStatisticsVendorSpecific OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(0..255))
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute provides an envelope for any optional vendor specific sub-
        elements which may be included in a measurement report element. The
        default value is null."
    DEFVAL { ''H }
    ::= { dot11STAStatisticsReportEntry 40 }

dot11STAStatisticsRprtMeasurementMode OBJECT-TYPE
SYNTAX INTEGER {
    success(0),
    incapableBit(1),
    refusedBit(2)
    }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the outcome status for the measurement request
        which generated this measurement report; status is indicated using the
        following reason codes: 1 indicates this STA is incapable of generating
        the report, 2 indicates this STA is refusing to generate the report, 0
        indicates the STA successfully carried out the measurement request."
    DEFVAL { 0 }
    ::= { dot11STAStatisticsReportEntry 41 }

dot11STAStatisticsRSNAStatsCMACICVErrors OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        If dot11STAStatisticsMeasurementDuration is 0, this attribute indicates
        the value of dot11RSNAStatsCMACICVErrors returned from the STA in this STA
        Statistics Report. If dot11STAStatisticsMeasurementDuration indicates a
        nonzero value, this attribute indicates the difference in the referenced
        dot11 variable over the indicated duration. This attribute is only valid
        if the dot11STAStatisticsGroupID is 16, and is ignored otherwise."
    ::= { dot11STAStatisticsReportEntry 42 }

dot11STAStatisticsRSNAStatsCMACReplays OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        If dot11STAStatisticsMeasurementDuration is 0, this attribute indicates
        the value of dot11RSNAStatsCMACReplays returned from the STA in this STA
        Statistics Report. If dot11STAStatisticsMeasurementDuration indicates a
        nonzero value, this attribute indicates the difference in the referenced
        dot11 variable over the indicated duration. This attribute is only valid
        if the dot11STAStatisticsGroupID is 16, and is ignored otherwise."
    ::= { dot11STAStatisticsReportEntry 43 }

dot11STAStatisticsRSNAStatsRobustMgmtCCMPReplays OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        If dot11STAStatisticsMeasurementDuration is 0, this attribute indicates
        the value of dot11RSNAStatsRobustMgmtCCMPReplays returned from the STA in
```

```
      this STA Statistics Report. If dot11STAStatisticsMeasurementDuration indi-
      cates a nonzero value, this attribute indicates the difference in the ref-
      erenced dot11 variable over the indicated duration. This attribute is only
      valid if the dot11STAStatisticsGroupID is 16, and is ignored otherwise."
   ::= { dot11STAStatisticsReportEntry 44 }

dot11STAStatisticsRSNAStatsTKIPICVErrors OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      If dot11STAStatisticsMeasurementDuration is 0, this attribute indicates
      the value of dot11RSNAStatsTKIPICVErrors returned from the STA in this STA
      Statistics Report. If dot11STAStatisticsMeasurementDuration indicates a
      nonzero value, this attribute indicates the difference in the referenced
      dot11 variable over the indicated duration. This attribute is only valid
      if the dot11STAStatisticsGroupID is 16, and is ignored otherwise."
   ::= { dot11STAStatisticsReportEntry 45 }

dot11STAStatisticsRSNAStatsTKIPReplays OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      If dot11STAStatisticsMeasurementDuration is 0, this attribute indicates
      the value of dot11RSNAStatsTKIPReplays returned from the STA in this STA
      Statistics Report. If dot11STAStatisticsMeasurementDuration indicates a
      nonzero value, this attribute indicates the difference in the referenced
      dot11 variable over the indicated duration. This attribute is only valid
      if the dot11STAStatisticsGroupID is 16, and is ignored otherwise."
   ::= { dot11STAStatisticsReportEntry 46 }

dot11STAStatisticsRSNAStatsCCMPDecryptErrors OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      If dot11STAStatisticsMeasurementDuration is 0, this attribute indicates
      the value of dot11RSNAStatsCCMPDecryptErrors returned from the STA in this
      STA Statistics Report. If dot11STAStatisticsMeasurementDuration indicates
      a nonzero value, this attribute indicates the difference in the referenced
      dot11 variable over the indicated duration. This attribute is only valid
      if the dot11STAStatisticsGroupID is 16, and is ignored otherwise."
   ::= { dot11STAStatisticsReportEntry 47 }

dot11STAStatisticsRSNAStatsCCMPReplays OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      If dot11STAStatisticsMeasurementDuration is 0, this attribute indicates
      the value of dot11RSNAStatsCCMPReplays returned from the STA in this STA
      Statistics Report. If dot11STAStatisticsMeasurementDuration indicates a
      nonzero value, this attribute indicates the difference in the referenced
      dot11 variable over the indicated duration. This attribute is only valid
      if the dot11STAStatisticsGroupID is 16, and is ignored otherwise."
   ::= { dot11STAStatisticsReportEntry 48 }

dot11STAStatisticsReportingReasonSTACounters OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(0..1))
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      This attribute indicates the trigger reason(s) for this Statistics Report.
      Each bit indicates a different trigger condition. When the bit is set to
      1, it indicates that the listed trigger threshold has been exceeded:
      B0 (least significant bit): dot11Failed,
      B1: dotFCSError,
      B2: dot11MultipleRetry,
      B3: dot11FrameDuplicate,
      B4: dot11RTSFailure,
      B5: dot11ACKFailure,
      B6: dot11Retry,
      B7: Reserved.
      This attribute is only valid if the dot11STAStatisticsGroupID is 0, and is
      ignored otherwise."
   ::= { dot11STAStatisticsReportEntry 49 }

dot11STAStatisticsReportingReasonQosCounters OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(0..1))
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      This attribute indicates the trigger reason(s) for this Statistics Report.
      Each bit indicates a different trigger condition. When the bit is set to
      1, it indicates that the listed trigger threshold has been exceeded:
      B0 (least significant bit): dot11QoSFailed,
      B1: dotQoSRetry,
      B2: dot11QoSMultipleRetry,
      B3: dot11QoSFrameDuplicate,
      B4: dot11QoSRTSFailure,
      B5: dot11QoSACKFailure,
      B6: dot11QoSDiscarded,
      B7: Reserved.
      This attribute is only valid if the dot11STAStatisticsGroupID is 2-9, and
      is ignored otherwise."
   ::= { dot11STAStatisticsReportEntry 50 }

dot11STAStatisticsReportingReasonRsnaCounters OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(0..1))
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      This attribute indicates the trigger reason(s) for this Statistics Report.
      Each bit indicates a different trigger condition. When the bit is set to
      1, it indicates that the listed trigger threshold has been exceeded:
      B0 (least significant bit): dot11RSNAStatsCMACICVErrors,
      B1: dotRSNAStatsCMACReplays,
      B2: dot11RSNAStatsRobustMgmtCCMPReplays,
      B3: dot11RSNAStatsTKIPICVErrors,
      B4: dot11RSNAStatsCCMPReplays,
      B5: dot11RSNAStatsCCMPDecryptErrors,
      B6: dot11RSNAStatsCCMPReplays,
      B7: Reserved.
      This attribute is only valid if the dot11STAStatisticsGroupID is 16, and
      is ignored otherwise."
   ::= { dot11STAStatisticsReportEntry 51 }

dot11STAStatisticsTransmittedAMSDUCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      If dot11STAStatisticsMeasurementDuration is zero, this attribute indicates
      the value of dot11TransmittedAMSDUCount returned from the STA in this STA
      Statistics Report. If dot11STAStatisticsMeasurementDuration indicates a
      nonzero value, this attribute indicates the difference in the referenced
      dot11 variable over the indicated duration. This attribute is only valid
      if the dot11STAStatisticsGroupID is 11, and is ignored otherwise."
   ::= { dot11STAStatisticsReportEntry 52 }

dot11STAStatisticsFailedAMSDUCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      If dot11STAStatisticsMeasurementDuration is zero, this attribute indicates
      the value of dot11FailedAMSDUCount returned from the STA in this STA Sta-
      tistics Report. If dot11STAStatisticsMeasurementDuration indicates a non-
      zero value, this attribute indicates the difference in the referenced
      dot11 variable over the indicated duration. This attribute is only valid
      if the dot11STAStatisticsGroupID is 11, and is ignored otherwise."
   ::= { dot11STAStatisticsReportEntry 53 }

dot11STAStatisticsRetryAMSDUCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      If dot11STAStatisticsMeasurementDuration is zero, this attribute indicates
      the value of dot11RetryAMSDUCount returned from the STA in this STA Sta-
      tistics Report. If dot11STAStatisticsMeasurementDuration indicates a non-
      zero value, this attribute indicates the difference in the referenced
      dot11 variable over the indicated duration. This attribute is only valid
      if the dot11STAStatisticsGroupID is 11, and is ignored otherwise."
   ::= { dot11STAStatisticsReportEntry 54 }

dot11STAStatisticsMultipleRetryAMSDUCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
```

```
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      If dot11STAStatisticsMeasurementDuration is zero, this attribute indicates
      the value of dot11MultipleRetryAMSDUCount returned from the STA in this
      STA Statistics Report. If dot11STAStatisticsMeasurementDuration indicates
      a nonzero value, this attribute indicates the difference in the referenced
      dot11 variable over the indicated duration. This attribute is only valid
      if the dot11STAStatisticsGroupID is 11, and is ignored otherwise."
   ::= { dot11STAStatisticsReportEntry 55 }

dot11STAStatisticsTransmittedOctetsInAMSDUCount OBJECT-TYPE
   SYNTAX Counter64
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      If dot11STAStatisticsMeasurementDuration is zero, this attribute indicates
      the value of dot11TransmittedOctetsInAMSDUCount returned from the STA in
      this STA Statistics Report. If dot11STAStatisticsMeasurementDuration indi-
      cates a nonzero value, this attribute indicates the difference in the ref-
      erenced dot11 variable over the indicated duration. This attribute is only
      valid if the dot11STAStatisticsGroupID is 11, and is ignored otherwise."
   ::= { dot11STAStatisticsReportEntry 56 }

dot11STAStatisticsAMSDUAckFailureCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      If dot11STAStatisticsMeasurementDuration is zero, this attribute indicates
      the value of dot11AMSDUAckFailureCount returned from the STA in this STA
      Statistics Report. If dot11STAStatisticsMeasurementDuration indicates a
      nonzero value, this attribute indicates the difference in the referenced
      dot11 variable over the indicated duration. This attribute is only valid
      if the dot11STAStatisticsGroupID is 11, and is ignored otherwise."
   ::= { dot11STAStatisticsReportEntry 57 }

dot11STAStatisticsReceivedAMSDUCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      If dot11STAStatisticsMeasurementDuration is zero, this attribute indicates
      the value of dot11ReceivedAMSDUCount returned from the STA in this STA
      Statistics Report. If dot11STAStatisticsMeasurementDuration indicates a
      nonzero value, this attribute indicates the difference in the referenced
      dot11 variable over the indicated duration. This attribute is only valid
      if the dot11STAStatisticsGroupID is 11, and is ignored otherwise."
   ::= { dot11STAStatisticsReportEntry 58 }

dot11STAStatisticsReceivedOctetsInAMSDUCount OBJECT-TYPE
   SYNTAX Counter64
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
```

```
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        If dot11STAStatisticsMeasurementDuration is zero, this attribute indicates
        the value of dot11ReceivedOctetsInAMSDUCount returned from the STA in this
        STA Statistics Report. If dot11STAStatisticsMeasurementDuration indicates
        a nonzero value, this attribute indicates the difference in the referenced
        dot11 variable over the indicated duration. This attribute is onl/y valid
        if the dot11STAStatisticsGroupID is 11, and is ignored otherwise."
    ::= { dot11STAStatisticsReportEntry 59 }

dot11STAStatisticsTransmittedAMPDUCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        If dot11STAStatisticsMeasurementDuration is zero, this attribute indicates
        the value of dot11TransmittedAMPDUCount returned from the STA in this STA
        Statistics Report. If dot11STAStatisticsMeasurementDuration indicates a
        nonzero value, this attribute indicates the difference in the referenced
        dot11 variable over the indicated duration. This attribute is only valid
        if the dot11STAStatisticsGroupID is 12, and is ignored otherwise."
    ::= { dot11STAStatisticsReportEntry 60 }

dot11STAStatisticsTransmittedMPDUsInAMPDUCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        If dot11STAStatisticsMeasurementDuration is zero, this attribute indicates
        the value of dot11TransmittedMPDUsInAMPDUCount returned from the STA in
        this STA Statistics Report. If dot11STAStatisticsMeasurementDuration indi-
        cates a nonzero value, this attribute indicates the difference in the ref-
        erenced dot11 variable over the indicated duration. This attribute is only
        valid if the dot11STAStatisticsGroupID is 12, and is ignored otherwise."
    ::= { dot11STAStatisticsReportEntry 61 }

dot11STAStatisticsTransmittedOctetsInAMPDUCount OBJECT-TYPE
    SYNTAX Counter64
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        If dot11STAStatisticsMeasurementDuration is zero, this attribute indicates
        the value of dot11TransmittedOctetsInAMPDUCount returned from the STA in
        this STA Statistics Report. If dot11STAStatisticsMeasurementDuration indi-
        cates a nonzero value, this attribute indicates the difference in the ref-
        erenced dot11 variable over the indicated duration. This attribute is only
        valid if the dot11STAStatisticsGroupID is 12, and is ignored otherwise."
    ::= { dot11STAStatisticsReportEntry 62 }

dot11STAStatisticsAMPDUReceivedCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
```

```
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      If dot11STAStatisticsMeasurementDuration is zero, this attribute indicates
      the value of dot11AMPDUReceivedCount returned from the STA in this STA
      Statistics Report. If dot11STAStatisticsMeasurementDuration indicates a
      nonzero value, this attribute indicates the difference in the referenced
      dot11 variable over the indicated duration. This attribute is only valid
      if the dot11STAStatisticsGroupID is 12, and is ignored otherwise."
   ::= { dot11STAStatisticsReportEntry 63 }


dot11STAStatisticsMPDUInReceivedAMPDUCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      If dot11STAStatisticsMeasurementDuration is zero, this attribute indicates
      the value of dot11MPDUInReceivedAMPDUCount returned from the STA in this
      STA Statistics Report. If dot11STAStatisticsMeasurementDuration indicates
      a nonzero value, this attribute indicates the difference in the referenced
      dot11 variable over the indicated duration. This attribute is only valid
      if the dot11STAStatisticsGroupID is 12, and is ignored otherwise."
   ::= { dot11STAStatisticsReportEntry 64 }


dot11STAStatisticsReceivedOctetsInAMPDUCount OBJECT-TYPE
   SYNTAX Counter64
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      If dot11STAStatisticsMeasurementDuration is zero, this attribute indicates
      the value of dot11ReceivedOctetsInAMPDUCount returned from the STA in this
      STA Statistics Report. If dot11STAStatisticsMeasurementDuration indicates
      a nonzero value, this attribute indicates the difference in the referenced
      dot11 variable over the indicated duration. This attribute is only valid
      if the dot11STAStatisticsGroupID is 12, and is ignored otherwise."
   ::= { dot11STAStatisticsReportEntry 65 }


dot11STAStatisticsAMPDUDelimiterCRCErrorCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      If dot11STAStatisticsMeasurementDuration is zero, this attribute indicates
      the value of dot11AMPDUDelimiterCRCErrorCount returned from the STA in
      this STA Statistics Report. If dot11STAStatisticsMeasurementDuration indi-
      cates a nonzero value, this attribute indicates the difference in the ref-
      erenced dot11 variable over the indicated duration. This attribute is only
      valid if the dot11STAStatisticsGroupID is 12, and is ignored otherwise."
   ::= { dot11STAStatisticsReportEntry 66 }

dot11STAStatisticsImplicitBARFailureCount OBJECT-TYPE
   SYNTAX Counter32
```

```
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        If dot11STAStatisticsMeasurementDuration is zero, this attribute indicates
        the value of dot11ImplicitBARFailureCount returned from the STA in this
        STA Statistics Report. If dot11STAStatisticsMeasurementDuration indicates
        a nonzero value, this attribute indicates the difference in the referenced
        dot11 variable over the indicated duration. This attribute is only valid
        if the dot11STAStatisticsGroupID is 13, and is ignored otherwise."
    ::= { dot11STAStatisticsReportEntry 67 }

dot11STAStatisticsExplicitBARFailureCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        If dot11STAStatisticsMeasurementDuration is zero, this attribute indicates
        the value of dot11ExplicitBARFailureCount returned from the STA in this
        STA Statistics Report. If dot11STAStatisticsMeasurementDuration indicates
        a nonzero value, this attribute indicates the difference in the referenced
        dot11 variable over the indicated duration. This attribute is only valid
        if the dot11STAStatisticsGroupID is 13, and is ignored otherwise."
    ::= { dot11STAStatisticsReportEntry 68 }

dot11STAStatisticsChannelWidthSwitchCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        If dot11STAStatisticsMeasurementDuration is zero, this attribute indicates
        the value of dot11ChannelWidthSwitchCount returned from the STA in this
        STA Statistics Report. If dot11STAStatisticsMeasurementDuration indicates
        a nonzero value, this attribute indicates the difference in the referenced
        dot11 variable over the indicated duration. This attribute is only valid
        if the dot11STAStatisticsGroupID is 13, and is ignored otherwise."
    ::= { dot11STAStatisticsReportEntry 69 }

dot11STAStatisticsTwentyMHzFrameTransmittedCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        If dot11STAStatisticsMeasurementDuration is zero, this attribute indicates
        the value of dot11TwentyMHzFrameTransmittedCount returned from the STA in
        this STA Statistics Report. If dot11STAStatisticsMeasurementDuration indi-
        cates a nonzero value, this attribute indicates the difference in the ref-
        erenced dot11 variable over the indicated duration. This attribute is only
        valid if the dot11STAStatisticsGroupID is 13, and is ignored otherwise."
    ::= { dot11STAStatisticsReportEntry 70 }

dot11STAStatisticsFortyMHzFrameTransmittedCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        If dot11STAStatisticsMeasurementDuration is zero, this attribute indicates
        the value of dot11FortyMHzFrameTransmittedCount returned from the STA in
        this STA Statistics Report. If dot11STAStatisticsMeasurementDuration indi-
        cates a nonzero value, this attribute indicates the difference in the ref-
        erenced dot11 variable over the indicated duration. This attribute is only
        valid if the dot11STAStatisticsGroupID is 13, and is ignored otherwise."
    ::= { dot11STAStatisticsReportEntry 71 }

dot11STAStatisticsTwentyMHzFrameReceivedCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        If dot11STAStatisticsMeasurementDuration is zero, this attribute indicates
        the value of dot11TwentyMHzFrameReceivedCount returned from the STA in
        this STA Statistics Report. If dot11STAStatisticsMeasurementDuration indi-
        cates a nonzero value, this attribute indicates the difference in the ref-
        erenced dot11 variable over the indicated duration. This attribute is only
        valid if the dot11STAStatisticsGroupID is 13, and is ignored otherwise."
    ::= { dot11STAStatisticsReportEntry 72 }

dot11STAStatisticsFortyMHzFrameReceivedCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        If dot11STAStatisticsMeasurementDuration is zero, this attribute indicates
        the value of dot11FortyMHzFrameReceivedCount returned from the STA in this
        STA Statistics Report. If dot11STAStatisticsMeasurementDuration indicates
        a nonzero value, this attribute indicates the difference in the referenced
        dot11 variable over the indicated duration. This attribute is only valid
        if the dot11STAStatisticsGroupID is 13, and is ignored otherwise."
    ::= { dot11STAStatisticsReportEntry 73 }

dot11STAStatisticsPSMPUTTGrantDuration OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        If dot11STAStatisticsMeasurementDuration is zero, this attribute indicates
        the value of dot11PSMPUTTGrantDuration returned from the STA in this STA
        Statistics Report. If dot11STAStatisticsMeasurementDuration indicates a
        nonzero value, this attribute indicates the difference in the referenced
        dot11 variable over the indicated duration. This attribute is only valid
        if the dot11STAStatisticsGroupID is 13, and is ignored otherwise."
    ::= { dot11STAStatisticsReportEntry 74 }

dot11STAStatisticsPSMPUTTUsedDuration OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        If dot11STAStatisticsMeasurementDuration is zero, this attribute indicates
        the value of dot11PSMPUTTUsedDuration returned from the STA in this STA
        Statistics Report. If dot11STAStatisticsMeasurementDuration indicates a
        nonzero value, this attribute indicates the difference in the referenced
        dot11 variable over the indicated duration. This attribute is only valid
        if the dot11STAStatisticsGroupID is 13, and is ignored otherwise."
    ::= { dot11STAStatisticsReportEntry 75 }

dot11STAStatisticsGrantedRDGUsedCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        If dot11STAStatisticsMeasurementDuration is zero, this attribute indicates
        the value of dot11GrantedRDGUsedCount returned from the STA in this STA
        Statistics Report. If dot11STAStatisticsMeasurementDuration indicates a
        nonzero value, this attribute indicates the difference in the referenced
        dot11 variable over the indicated duration. This attribute is only valid
        if the dot11STAStatisticsGroupID is 14, and is ignored otherwise."
    ::= { dot11STAStatisticsReportEntry 76 }

dot11STAStatisticsGrantedRDGUnusedCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        If dot11STAStatisticsMeasurementDuration is zero, this attribute indicates
        the value of dot11GrantedRDGUnusedCount returned from the STA in this STA
        Statistics Report. If dot11STAStatisticsMeasurementDuration indicates a
        nonzero value, this attribute indicates the difference in the referenced
        dot11 variable over the indicated duration. This attribute is only valid
        if the dot11STAStatisticsGroupID is 14, and is ignored otherwise."
    ::= { dot11STAStatisticsReportEntry 77 }

dot11STAStatisticsTransmittedFramesInGrantedRDGCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        If dot11STAStatisticsMeasurementDuration is zero, this attribute indicates
        the value of dot11TransmittedFramesInGrantedRDGCount returned from the STA
        in this STA Statistics Report. If dot11STAStatisticsMeasurementDuration
        indicates a nonzero value, this attribute indicates the difference in the
        referenced dot11 variable over the indicated duration. This attribute is
        only valid if the dot11STAStatisticsGroupID is 14, and is ignored other-
        wise."
    ::= { dot11STAStatisticsReportEntry 78 }

dot11STAStatisticsTransmittedOctetsInGrantedRDGCount OBJECT-TYPE
```

```
    SYNTAX Counter64
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        If dot11STAStatisticsMeasurementDuration is zero, this attribute indicates
        the value of dot11TransmittedOctetsInGrantedRDGCount returned from the STA
        in this STA Statistics Report. If dot11STAStatisticsMeasurementDuration
        indicates a nonzero value, this attribute indicates the difference in the
        referenced dot11 variable over the indicated duration. This attribute is
        only valid if the dot11STAStatisticsGroupID is 14, and is ignored other-
        wise."
    ::= { dot11STAStatisticsReportEntry 79 }

dot11STAStatisticsDualCTSSuccessCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        If dot11STAStatisticsMeasurementDuration is zero, this attribute indicates
        the value of dot11DualCTSSuccessCount returned from the STA in this STA
        Statistics Report. If dot11STAStatisticsMeasurementDuration indicates a
        nonzero value, this attribute indicates the difference in the referenced
        dot11 variable over the indicated duration. This attribute is only valid
        if the dot11STAStatisticsGroupID is 14, and is ignored otherwise."
    ::= { dot11STAStatisticsReportEntry 80 }

dot11STAStatisticsDualCTSFailureCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        If dot11STAStatisticsMeasurementDuration is zero, this attribute indicates
        the value of dot11DualCTSFailureCount returned from the STA in this STA
        Statistics Report. If dot11STAStatisticsMeasurementDuration indicates a
        nonzero value, this attribute indicates the difference in the referenced
        dot11 variable over the indicated duration. This attribute is only valid
        if the dot11STAStatisticsGroupID is 14, and is ignored otherwise."
    ::= { dot11STAStatisticsReportEntry 81 }

dot11STAStatisticsRTSLSIGSuccessCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        If dot11STAStatisticsMeasurementDuration is zero, this attribute indicates
        the value of dot11RTSLSIGSuccessCount returned from the STA in this STA
        Statistics Report. If dot11STAStatisticsMeasurementDuration indicates a
        nonzero value, this attribute indicates the difference in the referenced
        dot11 variable over the indicated duration. This attribute is only valid
        if the dot11STAStatisticsGroupID is 14, and is ignored otherwise."
    ::= { dot11STAStatisticsReportEntry 82 }
```

```
dot11STAStatisticsRTSLSIGFailureCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        If dot11STAStatisticsMeasurementDuration is zero, this attribute indicates
        the value of dot11RTSLSIGFailureCount returned from the STA in this STA
        Statistics Report. If dot11STAStatisticsMeasurementDuration indicates a
        nonzero value, this attribute indicates the difference in the referenced
        dot11 variable over the indicated duration. This attribute is only valid
        if the dot11STAStatisticsGroupID is 14, and is ignored otherwise."
    ::= { dot11STAStatisticsReportEntry 83 }

dot11STAStatisticsBeamformingFrameCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        If dot11STAStatisticsMeasurementDuration is zero, this attribute indicates
        the value of dot11BeamformingFrameCount returned from the STA in this STA
        Statistics Report. If dot11STAStatisticsMeasurementDuration indicates a
        nonzero value, this attribute indicates the difference in the referenced
        dot11 variable over the indicated duration. This attribute is only valid
        if the dot11STAStatisticsGroupID is 15, and is ignored otherwise."
    ::= { dot11STAStatisticsReportEntry 84 }

dot11STAStatisticsSTBCCTSSuccessCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        If dot11STAStatisticsMeasurementDuration is zero, this attribute indicates
        the value of dot11STBCCTSSuccessCount returned from the STA in this STA
        Statistics Report. If dot11STAStatisticsMeasurementDuration indicates a
        nonzero value, this attribute indicates the difference in the referenced
        dot11 variable over the indicated duration. This attribute is only valid
        if the dot11STAStatisticsGroupID is 15, and is ignored otherwise."
    ::= { dot11STAStatisticsReportEntry 85 }

dot11STAStatisticsSTBCCTSFailureCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        If dot11STAStatisticsMeasurementDuration is zero, this attribute indicates
        the value of dot11STBCCTSFailureCount returned from the STA in this STA
        Statistics Report. If dot11STAStatisticsMeasurementDuration indicates a
        nonzero value, this attribute indicates the difference in the referenced
        dot11 variable over the indicated duration. This attribute is only valid
        if the dot11STAStatisticsGroupID is 15, and is ignored otherwise."
    ::= { dot11STAStatisticsReportEntry 86 }
```

```
dot11STAStatisticsnonSTBCCTSSuccessCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        If dot11STAStatisticsMeasurementDuration is zero, this attribute indicates
        the value of dot11nonSTBCCTSSuccessCountt returned from the STA in this
        STA Statistics Report. If dot11STAStatisticsMeasurementDuration indicates
        a nonzero value, this attribute indicates the difference in the referenced
        dot11 variable over the indicated duration. This attribute is only valid
        if the dot11STAStatisticsGroupID is 15, and is ignored otherwise."
    ::= { dot11STAStatisticsReportEntry 87 }

dot11STAStatisticsnonSTBCCTSFailureCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        If dot11STAStatisticsMeasurementDuration is zero, this attribute indicates
        the value of dot11nonSTBCCTSFailureCount returned from the STA in this STA
        Statistics Report. If dot11STAStatisticsMeasurementDuration indicates a
        nonzero value, this attribute indicates the difference in the referenced
        dot11 variable over the indicated duration. This attribute is only valid
        if the dot11STAStatisticsGroupID is 15, and is ignored otherwise."
    ::= { dot11STAStatisticsReportEntry 88 }

-- ********************************************************************
-- * End of dot11STAStatisticsReport TABLE
-- ********************************************************************

-- ********************************************************************
-- * dot11LCIReport TABLE
-- ********************************************************************
dot11LCIReportTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11LCIReportEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
    "This table contains the current list of LCI reports that have been received
        by the MLME. The report tables are maintained as a FIFO to preserve fresh-
        ness, thus the rows in this table can be deleted for memory constraints or
        other implementation constraints determined by the vendor. New rows have
        different RprtIndex values than those deleted within the range limitation
        of the index. One easy way is to monotonically increase RprtIndex for new
        reports being written in the table."
    ::= { dot11RMReport 6 }

dot11LCIReportEntry OBJECT-TYPE
    SYNTAX Dot11LCIReportEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry in the dot11LCIReportTable Indexed by dot11LCIReportIndex."
    INDEX { dot11LCIReportIndex }
    ::= { dot11LCIReportTable 1 }

Dot11LCIReportEntry ::=
    SEQUENCE {
        dot11LCIReportIndex                                  Unsigned32,
        dot11LCIReportToken                                  OCTET STRING,
        dot11LCIIfIndex                                      InterfaceIndex,
        dot11LCISTAAddress                                   MacAddress,
        dot11LCILatitudeResolution                           Unsigned32,
        dot11LCILatitudeInteger                              Integer32,
        dot11LCILatitudeFraction                             Integer32,
        dot11LCILongitudeResolution                          Unsigned32,
        dot11LCILongitudeInteger                             Integer32,
        dot11LCILongitudeFraction                            Integer32,
        dot11LCIAltitudeType                                 INTEGER,
        dot11LCIAltitudeResolution                           Unsigned32,
        dot11LCIAltitudeInteger                              Integer32,
        dot11LCIAltitudeFraction                             Integer32,
        dot11LCIDatum                                        Unsigned32,
        dot11LCIAzimuthType                                  INTEGER,
        dot11LCIAzimuthResolution                            Unsigned32,
        dot11LCIAzimuth                                      Integer32,
        dot11LCIVendorSpecific                               OCTET STRING,
        dot11LCIRprtMeasurementMode                          INTEGER}

dot11LCIReportIndex OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Index for LCI Report elements in dot11LCIReportTable, greater than 0."
    ::= { dot11LCIReportEntry 1 }

dot11LCIReportToken OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the token that was indicated in the measurement
        request that generated this measurement report. This should be an exact
        match to the original dot11RMRqstToken attribute. Note that there may be
        multiple entries in the table that match this value since a single request
        may generate multiple measurement reports."
    ::= { dot11LCIReportEntry 2 }

dot11LCIIfIndex OBJECT-TYPE
    SYNTAX InterfaceIndex
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        Identifies the Interface that this row of LCI Report has been received on"
    ::= { dot11LCIReportEntry 3 }

dot11LCISTAAddress OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The MAC address of the STA that returned this LCI Report."
    ::= { dot11LCIReportEntry 4 }

dot11LCILatitudeResolution OBJECT-TYPE
    SYNTAX Unsigned32 (0..63)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the latitude resolution as 6 bits indicating the
        number of valid bits in the fixed-point value of Latitude. This field is
        derived from IETF RFC 3825."
    ::= { dot11LCIReportEntry 5 }

dot11LCILatitudeInteger OBJECT-TYPE
    SYNTAX Integer32 (-359..359)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the latitude as a 34 bit fixed point value con-
        sisting of 9 bits of integer and 25 bits of fraction. This field contains
        the 9 bits of integer portion of Latitude. This field is derived from IETF
        RFC 3825."
    ::= { dot11LCIReportEntry 6 }

dot11LCILatitudeFraction OBJECT-TYPE
    SYNTAX Integer32 (-16777215..16777215)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the latitude as a 34 bit fixed point value con-
        sisting of 9 bits of integer and 25 bits of fraction. This field contains
        the 25 bits of fraction portion of Latitude. This field is derived from
        IETF RFC 3825."
    ::= { dot11LCIReportEntry 7 }

dot11LCILongitudeResolution OBJECT-TYPE
    SYNTAX Unsigned32 (0..63)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the longitude resolution as 6 bits indicating the
        number of valid bits in the fixed-point value of Longitude. This field is
        derived from IETF RFC 3825."
    ::= { dot11LCIReportEntry 8 }

dot11LCILongitudeInteger OBJECT-TYPE
    SYNTAX Integer32 (-359..359)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.
        This attribute indicates the longitude as a 34 bit fixed point value con-
        sisting of 9 bits of integer and 25 bits of fraction. This field contains
        the 9 bits of integer portion of Longitude. This field is derived from
        IETF RFC 3825."
    ::= { dot11LCIReportEntry 9 }

dot11LCILongitudeFraction OBJECT-TYPE
    SYNTAX Integer32 (-16777215..16777215)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the longitude as a 34 bit fixed point value con-
        sisting of 9 bits of integer and 25 bits of fraction. This field contains
        the 25 bits of fraction portion of Longitude. This field is derived from
        IETF RFC 3825."
    ::= { dot11LCIReportEntry 10 }

dot11LCIAltitudeType OBJECT-TYPE
    SYNTAX INTEGER { meters(1), floors(2) }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the altitude Type. Codes defined are:
        meters : in 2s-complement fixed-point 22-bit integer part with 8-bit frac-
        tion
        floors : in 2s-complement fixed-point 22-bit integer part with 8-bit frac-
        tion.
        This field is derived from IETF RFC 3825."
    ::= { dot11LCIReportEntry 11 }

dot11LCIAltitudeResolution OBJECT-TYPE
    SYNTAX Unsigned32 (0..63)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the altitude resolution as 6 bits indicating the
        number of valid bits in the altitude. This field is derived from IETF RFC
        3825."
    ::= { dot11LCIReportEntry 12 }

dot11LCIAltitudeInteger OBJECT-TYPE
    SYNTAX Integer32 (-2097151..2097151)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the altitude as a 30 bit value defined by the
        Altitude type field. The field is encoded as a 2s-complement fixed-point
        22-bit integer Part with 8-bit fraction. This field contains the fixed-
        point Part of Altitude. This field is derived from IETF RFC 3825."
    ::= { dot11LCIReportEntry 13 }

dot11LCIAltitudeFraction OBJECT-TYPE
```

```
    SYNTAX Integer32 (-127..127)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the altitude as a 30 bit value defined by the
        Altitude type field. The field is encoded as a 2s-complement fixed-point
        22-bit integer Part with 8-bit fraction. This field contains the fraction
        part of Altitude. This field is derived from IETF RFC 3825."
    ::= { dot11LCIReportEntry 14 }

dot11LCIDatum OBJECT-TYPE
    SYNTAX Unsigned32 (0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the datum as an eight-bit value encoding the hor-
        izontal and vertical references used for the coordinates given in this
        LCI."
    ::= { dot11LCIReportEntry 15 }

dot11LCIAzimuthType OBJECT-TYPE
    SYNTAX INTEGER { frontSurfaceOfSTA(0), radioBeam(1) }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the azimuth Type as a one bit attribute encoding
        the type of Azimuth. Codes defined are: front surface of STA : in 2s-com-
        plement fixed-point 9-bit integer; and radio beam : in 2s-complement
        fixed-point 9-bit integer"
    ::= { dot11LCIReportEntry 16 }

dot11LCIAzimuthResolution OBJECT-TYPE
    SYNTAX Unsigned32 (0..15)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the azimuth Resolution as 4 bits indicating the
        number of valid bits in the azimuth."
    ::= { dot11LCIReportEntry 17 }

dot11LCIAzimuth OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the azimuth as a 9 bit value defined by the Azi-
        muth Type field.The field is encoded as a 2s-complement fixed-point 9-bit
        integer horizontal angle in degrees from true north."
    ::= { dot11LCIReportEntry 18 }
```

```
dot11LCIVendorSpecific OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(0..255))
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute provides an envelope for any optional vendor specific sub-
        elements which may be included in a measurement report element. The
        default value is null."
    DEFVAL { ''H }
    ::= { dot11LCIReportEntry 19 }

dot11LCIRprtMeasurementMode OBJECT-TYPE
    SYNTAX INTEGER {
        success(0),
        incapableBit(1),
        refusedBit(2) }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the outcome status for the measurement request
        which generated this measurement report; status is indicated using the
        following reason codes: 1 indicates this STA is incapable of generating
        the report, 2 indicates this STA is refusing to generate the report, 0
        indicates the STA successfully carried out the measurement request."
    DEFVAL { 0 }
    ::= { dot11LCIReportEntry 20 }

-- **********************************************************************
-- * End of dot11LCIReport TABLE
-- **********************************************************************

-- **********************************************************************
-- * dot11TransmitStreamReport TABLE
-- **********************************************************************

dot11TransmitStreamReportTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11TransmitStreamReportEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "This table contains the current list of Transmit Delay Metrics reports
        that have been received by the MLME. The report tables are maintained as a
        FIFO to preserve freshness, thus the rows in this table can be deleted for
        memory constraints or other implementation constraints determined by the
        vendor. New rows have different RprtIndex values than those deleted within
        the range limitation of the index. One easy way is to monotonically
        increase RprtIndex for new reports being written in the table."
    ::= { dot11RMReport 7 }

dot11TransmitStreamReportEntry OBJECT-TYPE
    SYNTAX Dot11TransmitStreamReportEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry in the dot11TransmitStreamReportTable Indexed by
        dot11TransmitStreamRprtIndex."
    INDEX { dot11TransmitStreamRprtIndex }
    ::= { dot11TransmitStreamReportTable 1 }

Dot11TransmitStreamReportEntry ::=
    SEQUENCE {
        dot11TransmitStreamRprtIndex                      Unsigned32,
        dot11TransmitStreamRprtRqstToken                  OCTET STRING,
        dot11TransmitStreamRprtIfIndex                    InterfaceIndex,
        dot11TransmitStreamMeasuringSTAAddr               MacAddress,
        dot11TransmitStreamRprtActualStartTime            TSFType,
        dot11TransmitStreamRprtMeasurementDuration        Unsigned32,
        dot11TransmitStreamRprtPeerSTAAddress             MacAddress,
        dot11TransmitStreamRprtTID                        Unsigned32,
        dot11TransmitStreamRprtAverageQueueDelay          Unsigned32,
        dot11TransmitStreamRprtAverageTransmitDelay       Unsigned32,
        dot11TransmitStreamRprtTransmittedMSDUCount       Unsigned32,
        dot11TransmitStreamRprtMSDUDiscardedCount         Unsigned32,
        dot11TransmitStreamRprtMSDUFailedCount            Unsigned32,
        dot11TransmitStreamRprtMultipleRetryCount         Unsigned32,
        dot11TransmitStreamRprtCFPollsLostCount           Unsigned32,
        dot11TransmitStreamRprtBin0Range                  Unsigned32,
        dot11TransmitStreamRprtDelayHistogram             OCTET STRING,
        dot11TransmitStreamRprtReason                     INTEGER,
        dot11TransmitStreamRprtVendorSpecific             OCTET STRING,
        dot11TransmitStreamRprtMeasurementMode            INTEGER}

dot11TransmitStreamRprtIndex OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Index for Transmit Delay Metrics Report elements in
        dot11TransmitStreamReportTable, greater than 0."
    ::= { dot11TransmitStreamReportEntry 1 }

dot11TransmitStreamRprtRqstToken OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the request token that was indicated in the mea-
        surement request that generated this measurement report. This should be an
        exact match to the original dot11RMRqstToken attribute. Note that there
        may be multiple entries in the table that match this value since a single
        request may generate multiple measurement reports."
    ::= { dot11TransmitStreamReportEntry 2 }

dot11TransmitStreamRprtIfIndex OBJECT-TYPE
    SYNTAX InterfaceIndex
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The ifIndex of the interface on which this TransmitStream Report was
        received."
    ::= { dot11TransmitStreamReportEntry 3 }

dot11TransmitStreamMeasuringSTAAddr OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The MAC address of the measuring STA for this row of Transmit Delay Met-
        rics report."
    ::= { dot11TransmitStreamReportEntry 4 }

dot11TransmitStreamRprtActualStartTime OBJECT-TYPE
    SYNTAX TSFType
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the TSF value at the time when the measurement
        started or for a triggered Transmit Stream/Category Measurement report the
        TSF value at the reporting QoS STA when the trigger condition was met."
    ::= { dot11TransmitStreamReportEntry 5 }

dot11TransmitStreamRprtMeasurementDuration OBJECT-TYPE
    SYNTAX Unsigned32
    UNITS "TUs"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the duration over which the Transmit Delay Met-
        rics Report was measured. For a triggered Transmit Stream/Category Mea-
        surement Report, metrics are reported over a number of transmitted MSDUs
        rather than a duration, hence Measurement Duration is equal to 0."
    ::= { dot11TransmitStreamReportEntry 6 }

dot11TransmitStreamRprtPeerSTAAddress OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The MAC address present in the Address 1 field of the measured  data
        frames for this row of Transmit Stream/Category Measurement report."
    ::= { dot11TransmitStreamReportEntry 7 }

dot11TransmitStreamRprtTID OBJECT-TYPE
    SYNTAX Unsigned32(0..16)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the TC or TS for which traffic is to be measured.
        Values 0 to 15 are defined. Values 16-255 are reserved."
    ::= { dot11TransmitStreamReportEntry 8 }

dot11TransmitStreamRprtAverageQueueDelay OBJECT-TYPE
    SYNTAX Unsigned32
    UNITS "TUs"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the average delay of the frames (MSDUs) that are
        passed to the MAC during the measurement duration for the indicated desti-
        nation and the indicated Traffic Identifier. Queue Delay is measured from
        the time the MSDU is passed to the MAC until the transmission starts and
        is expressed in units of TUs."
    ::= { dot11TransmitStreamReportEntry 9 }

dot11TransmitStreamRprtAverageTransmitDelay OBJECT-TYPE
    SYNTAX Unsigned32
    UNITS "TUs"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the average delay of the frames (MSDUs) that are
        successfully transmitted during the measurement duration for the indicated
        destination and the indicated Traffic Identifier. Delay is measured from
        the time the MSDU is passed to the MAC until ACK is received from the
        intermediate destination."
    ::= { dot11TransmitStreamReportEntry 10}

dot11TransmitStreamRprtTransmittedMSDUCount OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the number of MSDUs to the peer STA for the TC,
        or TS given by the Traffic Identifier successfully transmitted in the mea-
        surement duration."
    ::= {dot11TransmitStreamReportEntry 11}

dot11TransmitStreamRprtMSDUDiscardedCount OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the number of MSDUs to the peer STA for the TC,
        or TS given by the Traffic Identifier discarded due either to the number
        of transmit attempts exceeding dot11ShortRetryLimit or dot11LongRetryLimit
        as appropriate, or due to the MSDU lifetime having been reached."
    ::= {dot11TransmitStreamReportEntry 12}

dot11TransmitStreamRprtMSDUFailedCount OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.
        This attribute indicates the number of MSDUs to the peer STA for the TC,
        or TS given by the Traffic Identifier discarded during the measurement
        duration due to the number of transmit attempts exceeding
        dot11ShortRetryLimit or dot11LongRetryLimit as appropriate."
    ::= {dot11TransmitStreamReportEntry 13}

dot11TransmitStreamRprtMultipleRetryCount OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the number of MSDUs for the TC, or TS given by
        the Traffic Identifier that are successfully transmitted after more than
        one retransmission attempt."
    ::= {dot11TransmitStreamReportEntry 14}

dot11TransmitStreamRprtCFPollsLostCount OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the number of QoS (+)CF-Poll frames transmitted
        to the peer STA where there was no response from the QoS STA."
    ::= {dot11TransmitStreamReportEntry 15}

dot11TransmitStreamRprtBin0Range OBJECT-TYPE
    SYNTAX Unsigned32
    UNITS "TUs"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the delay range for Bin 0 of the delay histo-
        gram."
    ::= { dot11TransmitStreamReportEntry 16 }

dot11TransmitStreamRprtDelayHistogram OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE (6))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the histogram of delay of the frames (MSDUs) that
        are successfully transmitted during the measurement duration for the indi-
        cated Traffic Identifier and the indicated destination. Delay is measured
        from the time the MSDU is passed to the MAC until the ACK is received from
        the intermediate destination and is expressed in units of TUs. "
    ::= { dot11TransmitStreamReportEntry 17 }

dot11TransmitStreamRprtReason OBJECT-TYPE
    SYNTAX INTEGER {
        averageTrigger(0),
        consecutiveTrigger(1),
        delayTrigger(2) }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the Reason field indicating the reason that the
        measuring QoS STA sent the Transmit Stream/Category measurement report."
    DEFVAL { 0 }
    ::= { dot11TransmitStreamReportEntry 18 }

dot11TransmitStreamRprtVendorSpecific OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(0..255))
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute provides an envelope for any optional vendor specific sub-
        elements which may be included in a measurement report element. The
        default value is null."
    DEFVAL { ''H }
    ::= { dot11TransmitStreamReportEntry 19 }

dot11TransmitStreamRprtMeasurementMode OBJECT-TYPE
    SYNTAX INTEGER {
        success(0),
        incapableBit(1),
        refusedBit(2) }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the outcome status for the measurement request
        which generated this measurement report; status is indicated using the
        following reason codes: 1 indicates this STA is incapable of generating
        the report, 2 indicates this STA is refusing to generate the report, 0
        indicates the STA successfully carried out the measurement request."
    DEFVAL { 0 }
    ::= { dot11TransmitStreamReportEntry 20 }

-- **********************************************************************
-- * End of dot11TransmitStreamReport TABLE
-- **********************************************************************

-- **********************************************************************
-- * Radio Measurement Configuration Information
-- **********************************************************************
    dot11RMConfig OBJECT IDENTIFIER ::= { dot11RadioMeasurement 3 }

-- **********************************************************************
-- * dot11APChannelReport TABLE
-- **********************************************************************
dot11APChannelReportTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11APChannelReportEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "AP Channel Report information, in tabular form."
    ::= { dot11RMConfig 1 }
```

```
dot11APChannelReportEntry OBJECT-TYPE
    SYNTAX Dot11APChannelReportEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry in the dot11APChannelReportTable. Each entry in the table is
        indexed by dot11APChannelReportIndex."
    INDEX { dot11APChannelReportIndex }
    ::= { dot11APChannelReportTable 1 }

Dot11APChannelReportEntry ::=
    SEQUENCE {
        dot11APChannelReportIndex                         Unsigned32,
        dot11APChannelReportIfIndex                       InterfaceIndex,
        dot11APChannelReportOperatingClass                Unsigned32,
        dot11APChannelReportChannelList                   OCTET STRING}

dot11APChannelReportIndex OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Index for AP channel report entry in dot11APChannelReportTable, greater
        than 0."
    ::= { dot11APChannelReportEntry 1 }

dot11APChannelReportIfIndex OBJECT-TYPE
    SYNTAX InterfaceIndex
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The ifIndex for this row of the AP channel report."
    ::= { dot11APChannelReportEntry 2 }

dot11APChannelReportOperatingClass OBJECT-TYPE
    SYNTAX Unsigned32(1..255)
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the channel set for this AP Channel Report. Coun-
        try, Operating Class and Channel Number together specify the channel fre-
        quency and spacing for this measurement report. Valid values of Operating
        Class are shown in Annex E."
    REFERENCE "Annex E"
    ::= { dot11APChannelReportEntry 3 }

dot11APChannelReportChannelList OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(0..255))
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute lists the specific channels in this AP Channel Report. The
        default value is null. Each octet indicates a different channel within the
        indicated Operating Class. This list of channels is the Channel List in
        the AP Channel Report element described in 8.4.2.38. "
    DEFVAL { ''H }
    ::= { dot11APChannelReportEntry 4 }

-- **********************************************************************
-- * End of dot11APChannelReportTable TABLE
-- **********************************************************************

-- **********************************************************************
-- * dot11RMNeighborReport TABLE
-- **********************************************************************
dot11RMNeighborReportNextIndex OBJECT-TYPE
    SYNTAX Unsigned32(0..255)
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Identifies the next available index for managing the neighbor report
        table. If this attribute is 0, it indicates that the neighbor report fea-
        ture is not configurable via SNMP, or the table is full and new rows can-
        not be accepted."
    ::= { dot11RMConfig 2 }

dot11RMNeighborReportTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11RMNeighborReportEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
    "This table contains pertinent information on a collection of BSSID's that
        are candidates to which STAs can roam. The rows are created using create-
        AndWait method and fill in the attributes. When the rowStatus is set to
        active, the row can be included in Neighbor Report elements. If there is
        an error, the rowStatus is set to notReady by SME. Since this table con-
        tains all Neighbor Report element entries for all interfaces enabled with
        the neighbor report feature, it is possible to have too many entries for
        one interface, while still remaining under the MaxTableSize. In that sit-
        uation, SME includes neighbor report entries only with lower
        dot11RMNeighborReportIFIndex up to the maximum possible number of entries
        for a particular interface identified by ifIndex. SME sets the rowStatus
        to notInService for those rows that cannot be included in the Neighbor
        Report element for that interface."
    ::= { dot11RMConfig 3 }

dot11RMNeighborReportEntry OBJECT-TYPE
    SYNTAX Dot11RMNeighborReportEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry in the dot11RMNeighborReportTable"
    INDEX { dot11RMNeighborReportIndex }
    ::= { dot11RMNeighborReportTable 1 }

Dot11RMNeighborReportEntry ::=
    SEQUENCE {
        dot11RMNeighborReportIndex                      Unsigned32,
        dot11RMNeighborReportIfIndex                    InterfaceIndex,
        dot11RMNeighborReportBSSID                      MacAddress,
        dot11RMNeighborReportAPReachability             INTEGER,
        dot11RMNeighborReportSecurity                   TruthValue,
        dot11RMNeighborReportCapSpectrumMgmt            TruthValue,
        dot11RMNeighborReportCapQoS                     TruthValue,
        dot11RMNeighborReportCapAPSD                    TruthValue,
        dot11RMNeighborReportCapRM                      TruthValue,
        dot11RMNeighborReportCapDelayBlockAck           TruthValue,
        dot11RMNeighborReportCapImmediateBlockAck       TruthValue,
        dot11RMNeighborReportKeyScope                   TruthValue,
        dot11RMNeighborReportOperatingClass               Unsigned32,
        dot11RMNeighborReportChannelNumber                Unsigned32,
        dot11RMNeighborReportPhyType                      INTEGER,
        dot11RMNeighborReportNeighborTSFInfo              OCTET STRING,
        dot11RMNeighborReportPilotInterval                Unsigned32,
        dot11RMNeighborReportPilotMultipleBSSID           OCTET STRING,
        dot11RMNeighborReportRMEnabledCapabilities        OCTET STRING,
        dot11RMNeighborReportVendorSpecific               OCTET STRING,
        dot11RMNeighborReportRowStatus                    RowStatus,
        dot11RMNeighborReportMobilityDomain               TruthValue,
        dot11RMNeighborReportCapHT                        TruthValue,
        dot11RMNeighborReportHTLDPCCodingCap              TruthValue,
        dot11RMNeighborReportHTSupportedChannelWidthSet
                                                          TruthValue,
        dot11RMNeighborReportHTSMPowerSave                Unsigned32,
        dot11RMNeighborReportHTGreenfield                 TruthValue,
        dot11RMNeighborReportHTShortGIfor20MHz            TruthValue,
        dot11RMNeighborReportHTShortGIfor40MHz            TruthValue,
        dot11RMNeighborReportHTTxSTBC                     TruthValue,
        dot11RMNeighborReportHTRxSTBC                     Unsigned32,
        dot11RMNeighborReportHTDelayedBlockAck            TruthValue,
        dot11RMNeighborReportHTMaxAMSDULength             TruthValue,
        dot11RMNeighborReportHTDSSCCKModein40MHz          TruthValue,
        dot11RMNeighborReportHTFortyMHzIntolerant         TruthValue,
        dot11RMNeighborReportHTLSIGTXOPProtectionSupport  TruthValue,
        dot11RMNeighborReportHTMaxAMPDULengthExponent     Unsigned32,
        dot11RMNeighborReportHTMinMPDUStartSpacing        Unsigned32,
        dot11RMNeighborReportHTRxMCSBitMask               OCTET STRING,
        dot11RMNeighborReportHTRxHighestSupportedDataRate Unsigned32,
        dot11RMNeighborReportHTTxMCSSetDefined            TruthValue,
        dot11RMNeighborReportHTTxRxMCSSetNotEqual         TruthValue,
        dot11RMNeighborReportHTTxMaxNumberSpatialStreamsSupported
                                                          Unsigned32,
        dot11RMNeighborReportHTTxUnequalModulationSupported
                                                          TruthValue,
        dot11RMNeighborReportHTPCO                        TruthValue,
        dot11RMNeighborReportHTPCOTransitionTime          Unsigned32,
        dot11RMNeighborReportHTMCSFeedback                Unsigned32,
        dot11RMNeighborReportHTCSupport                   TruthValue,
        dot11RMNeighborReportHTRDResponder                TruthValue,
        dot11RMNeighborReportHTImplictTransmitBeamformingReceivingCap
                                                          TruthValue,
        dot11RMNeighborReportHTReceiveStaggeredSoundingCap
                                                          TruthValue,
        dot11RMNeighborReportHTTransmitStaggeredSoundingCap
                                                          TruthValue,
        dot11RMNeighborReportHTReceiveNDPCap              TruthValue,
        dot11RMNeighborReportHTTransmitNDPCap             TruthValue,
        dot11RMNeighborReportHTImplicitTransmitBeamformingCap
                                                          TruthValue,
        dot11RMNeighborReportHTTransmitBeamformingCalibration
                                                          Unsigned32,
        dot11RMNeighborReportHTExplicitCSITransmitBeamformingCap
                                                          TruthValue,
        dot11RMNeighborReportHTExplicitNonCompressedSteeringCap
                                                          TruthValue,
        dot11RMNeighborReportHTExplicitCompressedSteeringCap
                                                          TruthValue,
        dot11RMNeighborReportHTExplicitTransmitBeamformingFeedback
                                                          Unsigned32,
        dot11RMNbRprtHTExplicitNonCompressedBeamformingFeedbackCap
                                                          Unsigned32,
        dot11RMNeighborReportHTExplicitCompressedBeamformingFeedbackCap
                                                          Unsigned32,
       dot11RMNeighborReportHTTransmitBeamformingMinimalGrouping
                                                   Unsigned32,
       dot11RMNbRprtHTCSINumberofTxBeamformingAntennasSuppt
                                                   Unsigned32,
       dot11RMNbRprtHTNonCompressedSteeringNumofTxBmfmingAntennasSuppt
                                                   Unsigned32,
       dot11RMNbRprtHTCompressedSteeringNumberofTxBmfmingAntennasSuppt
                                                   Unsigned32,
       dot11RMNbRprtHTCSIMaxNumberofRowsTxBeamformingSuppt
                                                   Unsigned32,
       dot11RMNeighborReportHTTransmitBeamformingChannelEstimationCap
                                                   Unsigned32,
       dot11RMNeighborReportHTAntSelectionCap           TruthValue,
       dot11RMNeighborReportHTExplicitCSIFeedbackBasedTxASELCap
                                                   TruthValue,
       dot11RMNeighborReportHTAntIndicesFeedbackBasedTxASELCap
                                                   TruthValue,
       dot11RMNeighborReportHTExplicitCSIFeedbackBasedCap
                                                   TruthValue,
       dot11RMNeighborReportHTAntIndicesFeedbackCap      TruthValue,
       dot11RMNeighborReportHTRxASELCap                  TruthValue,
       dot11RMNeighborReportHTTxSoundingPPDUsCap         TruthValue,
       dot11RMNeighborReportHTInfoPrimaryChannel         Unsigned32,
       dot11RMNeighborReportHTInfoSecChannelOffset       Unsigned32,
       dot11RMNeighborReportHTInfoSTAChannelWidth        TruthValue,
       dot11RMNeighborReportHTInfoRIFSMode               TruthValue,
       dot11RMNeighborReportHTInfoProtection             Unsigned32,
       dot11RMNeighborReportHTInfoNonGreenfieldHTSTAsPresent
                                                   TruthValue,
       dot11RMNeighborReportHTInfoOBSSNonHTSTAsPresent
                                                   TruthValue,
       dot11RMNeighborReportHTInfoDualBeacon             TruthValue,
       dot11RMNeighborReportHTInfoDualCTSProtection      TruthValue,
       dot11RMNeighborReportHTInfoSTBCBeacon             TruthValue,
       dot11RMNeighborReportHTInfoLSIGTXOPProtectionSup
                                                   TruthValue,
       dot11RMNeighborReportHTInfoPCOActive              TruthValue,
       dot11RMNeighborReportHTInfoPCOPhase               TruthValue,
       dot11RMNeighborReportHTInfoBasicMCSSet            OCTET STRING,
       dot11RMNeighborReportHTSecChannelOffset           Unsigned32,
       dot11RMNeighborReportExtCapPSMPSupport            TruthValue,
       dot11RMNeighborReportExtCapSPSMPSup               TruthValue,
       dot11RMNeighborReportExtCapServiceIntervalGranularity
                                                   Unsigned32,
       dot11RMNeighborReportBSSTransitCandPreference     Unsigned32,
       dot11RMNeighborReportBSSTerminationTSF            TSFType,
       dot11RMNeighborReportBSSTerminationDuration       Unsigned32
   }

dot11RMNeighborReportIndex OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "Index for neighbor report configuration table in
      dot11RMNeighborReportTable, greater than 0."
   ::= { dot11RMNeighborReportEntry 1 }

dot11RMNeighborReportIfIndex OBJECT-TYPE
   SYNTAX InterfaceIndex
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      The ifIndex for this row of the neighbor report."
   ::= { dot11RMNeighborReportEntry 2 }

dot11RMNeighborReportBSSID OBJECT-TYPE
   SYNTAX MacAddress
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      This attribute indicates the BSSID of the AP described by this row of
      neighbor report."
   ::= { dot11RMNeighborReportEntry 3 }

dot11RMNeighborReportAPReachability OBJECT-TYPE
   SYNTAX INTEGER { notReachable(1), unknown(2), reachable(3) }
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      This attribute indicates the reachability of the AP represented by the
      dot11NeighborReportBSSID."
   ::= { dot11RMNeighborReportEntry 4 }

dot11RMNeighborReportSecurity OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      This attribute, when true, indicates that the neighbor AP identified by
      this BSSID supports the same security provisioning as used by the AP which
      provided this neighbor report. This attribute, when false, indicates
      either that the neighbor AP identified by this BSSID does not support the
      same security provisioning or that the security information for this
      neighbor AP is not available at this time."
   ::= { dot11RMNeighborReportEntry 5 }

dot11RMNeighborReportCapSpectrumMgmt OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      This attribute indicates the spectrum management capability of the AP rep-
      resented by dot11NeighborReportBSSID."
   ::= { dot11RMNeighborReportEntry 6 }

dot11RMNeighborReportCapQoS OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.
      This attribute indicates the QoS capability of the AP represented by
      dot11NeighborReportBSSID."
   ::= { dot11RMNeighborReportEntry 7 }

dot11RMNeighborReportCapAPSD OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      This attribute indicates the APSD capability of the AP represented by
      dot11NeighborReportBSSID."
   ::= { dot11RMNeighborReportEntry 8 }

dot11RMNeighborReportCapRM OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      This attribute indicates the RM capability of the AP represented by
      dot11NeighborReportBSSID."
   ::= { dot11RMNeighborReportEntry 9 }

dot11RMNeighborReportCapDelayBlockAck OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      This attribute indicates the Delayed BlockAck capability of the AP repre-
      sented by dot11NeighborReportBSSID."
   ::= { dot11RMNeighborReportEntry 10 }

dot11RMNeighborReportCapImmediateBlockAck OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      This attribute indicates the Immediate BlockAck capability of the AP rep-
      resented by dot11NeighborReportBSSID."
   ::= { dot11RMNeighborReportEntry 11 }

dot11RMNeighborReportKeyScope OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      This attribute, when true, indicates the neighbor AP identified by this
      BSSID has the same authenticator as the AP which provided this neighbor
      report. This attribute, when false, indicates that the neighbor AP identi-
      fied by this BSSID has a different authenticator or that authenticator
      information is not available."
   ::= { dot11RMNeighborReportEntry 12 }

dot11RMNeighborReportOperatingClass OBJECT-TYPE
   SYNTAX Unsigned32(1..255)
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      This attribute indicates the channel set for this neighbor report entry.
      Country, Operating Class and Channel Number together specify the channel
      frequency and spacing for this measurement report. Valid values of Operat-
      ing Class are shown in Annex E."
   REFERENCE
      "Annex E"
   ::= { dot11RMNeighborReportEntry 13 }

dot11RMNeighborReportChannelNumber OBJECT-TYPE
   SYNTAX Unsigned32 (1..255)
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      This attribute indicates the current operating channel of the AP repre-
      sented by the dot11NeighborReportBSSID. The Channel Number is only defined
      within the indicated Operating Class for this neighbor report entry."
   ::= { dot11RMNeighborReportEntry 14 }

dot11RMNeighborReportPhyType OBJECT-TYPE
   SYNTAX INTEGER {
      fhss(1),
      dsss(2),
      irbaseband(3),
      ofdm(4),
      hrdsss(5),
      erp(6),
      ht(7) }
   UNITS "dot11PHYType"
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      This attribute indicates the PHY Type of the neighbor AP identified by
      this BSSID."
   ::= { dot11RMNeighborReportEntry 15 }

dot11RMNeighborReportNeighborTSFInfo OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE (6))
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      This attribute indicates TSF timing information for the neighbor AP iden-
      tified by this BSSID. The TSF timing information includes the TSF Offset
      and the Beacon Interval, as defined in 8.4.2.39."
```

```
    ::= { dot11RMNeighborReportEntry 16 }

dot11RMNeighborReportPilotInterval OBJECT-TYPE
    SYNTAX Unsigned32
    UNITS "TUs"
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates Measurement Pilot Interval for the neighbor AP
        identified by this BSSID, as defined in 8.4.1.18."
    ::= { dot11RMNeighborReportEntry 17 }

dot11RMNeighborReportPilotMultipleBSSID OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(1))
    UNITS "BSSID LSBs"
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates n, where 2**n is the maximum number of BSSIDs in
        the multiple BSSID set, as described in 10.11.14."
    ::= { dot11RMNeighborReportEntry 18 }

dot11RMNeighborReportRMEnabledCapabilities OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(7))
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute indicates the detailed enabled capabilities of the AP rep-
        resented by the dot11NeighborReportBSSID, as defined in 8.4.2.47."
    REFERENCE
        "IEEE 802.11 - 8.4.2.47"
    ::= { dot11RMNeighborReportEntry 19 }

dot11RMNeighborReportVendorSpecific OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(0..255))
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        This attribute provides an envelope for any optional vendor specific sub-
        elements which may be included in a measurement report element. The
        default value is null."
    DEFVAL { ''H }
    ::= { dot11RMNeighborReportEntry 20 }

dot11RMNeighborReportRowStatus OBJECT-TYPE
    SYNTAX RowStatus
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.
```

```
      Contains the row status of the neighbor report, essentially used for indi-
      cating whether the row has all valid attributes filled in. Then set to
      active to be used in Neighbor Report elements. If any parameter is
      invalid, the SME sets this attribute back to notReady. It is the responsi-
      bility of the manager to correct the parameters."
   ::= { dot11RMNeighborReportEntry 21 }

dot11RMNeighborReportMobilityDomain OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      Indicates a common mobility domain identifier (MDID) and an identical
      value of the FT Capability and Policy value."
   ::= { dot11RMNeighborReportEntry 22 }

dot11RMNeighborReportCapHT OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      The High Throughput Bit when equal to 1 indicates that the AP represented
      by this BSSID is an HT AP including the HT Capabilities element in its
      Beacons and that the contents of that HT Capabilities element are identi-
      cal to the HT Capabilities element advertised by the AP sending the
      report. See 8.4.2.39"
   ::= { dot11RMNeighborReportEntry 23 }

dot11RMNeighborReportHTLDPCCodingCap OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      The HT LDPC coding capability indicates support for receiving LDPC coded
      packets, equal to false if not supported, equal to true if supported. See
      8.4.2.58.2"
   ::= { dot11RMNeighborReportEntry 24 }

dot11RMNeighborReportHTSupportedChannelWidthSet OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      The HT Supported channel width set indicates which channel widths the STA
      supports, equal to false if only 20 MHz operation is supported, equal to
      true if both 20 MHz and 40 MHz operation is supported. See 8.4.2.58.2"
   ::= { dot11RMNeighborReportEntry 25 }

dot11RMNeighborReportHTSMPowerSave OBJECT-TYPE
   SYNTAX Unsigned32 (0..3)
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      The HT SM Power Save indicates the SM power save mode, equal to 0 for
      static SM power save mode, equal to 1 for dynamic SM power save mode,
      equal to 3 for SM power save disabled, the value 2 is reserved; see
      8.4.2.58.2"
   ::= { dot11RMNeighborReportEntry 26 }

dot11RMNeighborReportHTGreenfield OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      The HT-greenfield indicates support for the reception of PPDUs with HT-
      greenfield format, equal to false if not supported, equal to true if sup-
      ported. See 8.4.2.58.2"
   ::= { dot11RMNeighborReportEntry 27 }

dot11RMNeighborReportHTShortGIfor20MHz OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      The HT Short GI for 20 MHZ indicates short GI support for the reception of
      20 MHz packets, equal to false if not supported, equal to true if sup-
      ported See 8.4.2.58.2"
   ::= { dot11RMNeighborReportEntry 28 }

dot11RMNeighborReportHTShortGIfor40MHz OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      The HT Short GI for 40 MHz indicates short GI support for the reception of
      40 MHz packets, equal to false if not supported, equal to true if sup-
      ported. See 8.4.2.58.2"
   ::= { dot11RMNeighborReportEntry 29}

dot11RMNeighborReportHTTxSTBC OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      The HT Tx STBC indicates support for the transmission of PPDUs using STBC,
      equal to false if not supported, equal to true if supported. See
      8.4.2.58.2"
   ::= { dot11RMNeighborReportEntry 30}

dot11RMNeighborReportHTRxSTBC OBJECT-TYPE
   SYNTAX Unsigned32
```

```
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The HT Rx STBC indicates support for the reception of PPDUs using STBC,
        equal to 0 for no support, equal to 1 for support of one spatial stream,
        equal to 2 for support of one and two spatial streams, equal to 3 for sup-
        port of one, two, and three spatial streams. See 8.4.2.58.2"
    ::= { dot11RMNeighborReportEntry 31}

dot11RMNeighborReportHTDelayedBlockAck OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The HT-delayed Block ACK indicates support for HT-delayed Block ACK oper-
        ation, equal to false if not supported, equal to true if supported. Sup-
        port indicates that the STA is able to accept an ADDBA request for HT-
        delayed Block ACK. See 8.4.2.58.2"
    ::= { dot11RMNeighborReportEntry 32 }

dot11RMNeighborReportHTMaxAMSDULength OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The HT Maximum A-MSDU length indicates maximum A-MSDU length, equal to
        false for 3839 octets, equal to true for 7935 octets. See 8.4.2.58.2"
    ::= { dot11RMNeighborReportEntry 33 }

dot11RMNeighborReportHTDSSCCKModein40MHz OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The HT DSSS/CCK Mode in 40 MHz indicates use of DSSS/CCK mode in a 40 MHz
        capable BSS operating in 20/40 MHz mode, equal to false if DSSS/CCK in 40
        MHz is not allowed, equal to true if the DSSS/CCK in 40 MHz is allowed.
        See 8.4.2.58.2"
    ::= { dot11RMNeighborReportEntry 34 }

dot11RMNeighborReportHTFortyMHzIntolerant OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The HT Forty MHz Intolerant indicates whether other BSSs receiving this
        information are required to prohibit 40 MHz transmissions, equal to true
        to prohibit 20/40 MHz BSS operation, otherwise equal to false. See
        8.4.2.58.2"
    ::= { dot11RMNeighborReportEntry 35 }

dot11RMNeighborReportHTLSIGTXOPProtectionSupport OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The HT L-SIG TXOP protection support indicates support for the LSIG TXOP
        protection mechanism, equal to false if not supported, equal to true if
        supported. See 8.4.2.58.2"
    ::= { dot11RMNeighborReportEntry 36 }

dot11RMNeighborReportHTMaxAMPDULengthExponent OBJECT-TYPE
    SYNTAX Unsigned32 (0..3)
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The HT Maximum A-MPDU Length Exponent indicates the maximum length of A-
        MPDU that the STA can receive. This field is an integer in the range 0 to
        3. The length defined by this field is equal to 2(13 + Maximum A-MPDU
        Length) - 1 octets. See 8.4.2.58.3"
    ::= { dot11RMNeighborReportEntry 37 }

dot11RMNeighborReportHTMinMPDUStartSpacing OBJECT-TYPE
    SYNTAX Unsigned32 (0..7)
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The HT Minimum MPDU Start Spacing determines the minimum time between the
        start of adjacent MPDUs within an AMPDU, measured at the PHY-SAP, equal to
        0 for no restriction, equal to 1 for 1/4 microsecond, equal to 2 for 1/2
        microsecond, equal to 3 for 1 microsecond, equal to 4 for 2 microseconds,
        equal to 5 for 4 microseconds, equal to 6 for 8 microseconds, equal to 7
        for 16 microseconds. See 8.4.2.58.3"
    ::= { dot11RMNeighborReportEntry 38 }

dot11RMNeighborReportHTRxMCSBitMask OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(10))
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The HT Rx MCS Bitmask is a 77 bit subfield that defines a set of MCS val-
        ues, where bit B0 (i.e., the lsb of the first octet) corresponds to MCS 0
        and bit B76 corresponds to MCS 76, equal to 0 when the MCS is not sup-
        ported, equal to 1 when the MCS is supported. See 8.4.2.58.4"
    ::= { dot11RMNeighborReportEntry 39}

dot11RMNeighborReportHTRxHighestSupportedDataRate OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
```

```
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The HT Highest Supported Data Rate is a 10 bit subfield that defines the
        highest data rate that the STA is able to receive, in units of 1 Mb/s,
        where 1 represents 1 Mb/s, and incrementing by 1 Mb/s steps to the value
        1023, which represents 1023 Mb/s. See 8.4.2.58.4"
    ::= { dot11RMNeighborReportEntry 40 }

dot11RMNeighborReportHTTxMCSSetDefined OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The HT Tx MCS Set Defined indicates if the Tx MCS set is defined, equal to
        false if no Tx MCS set is defined, equal to true if Tx MCS set is defined.
        See 8.4.2.58.4"
    ::= { dot11RMNeighborReportEntry 41 }

dot11RMNeighborReportHTTxRxMCSSetNotEqual OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The HT Tx RX MCS set not equal indicates if the Tx MCS set is defined to
        be equal to the Rx MCS set, equal to false where no Tx MCS set is defined
        or where the Tx MCS Set is defined to be equal to the RX MCS Set, equal to
        true where the TX MCS set may differ from the Rx MCS set. See 8.4.2.58.4"
    ::= { dot11RMNeighborReportEntry 42 }

dot11RMNeighborReportHTTxMaxNumberSpatialStreamsSupported OBJECT-TYPE
    SYNTAX Unsigned32 (0..3)
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The HT Tx maximum number spatial streams supported indicates maximum num-
        ber of spatial streams supported when the Tx MCS Set may differ from the
        Rx MCS set, equal to 0 where no TX MCS set is defined or where the Tx MCS
        set is defined to be equal to the RX MCS set or where the maximum number
        of spatial streams supported when transmitting is 1 spatial stream and the
        Tx MCS set may differ from the Rx MCS set, equal to 1 where the maximum
        number of spatial streams supported when transmitting is 2 spatial streams
        and the Tx MCS set may differ from the Rx MCS set, equal to 2 where the
        maximum number of spatial streams supported when transmitting is 3 spatial
        streams and the Tx MCS set may differ from the Rx MCS set, equal to 3 where
        the maximum number of spatial streams supported when transmitting is 4
        spatial streams and the Tx MCS set may differ from the Rx MCS set. See
        8.4.2.58.4"
    ::= { dot11RMNeighborReportEntry 43 }

dot11RMNeighborReportHTTxUnequalModulationSupported OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
```

```
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The HT Tx UEQM supported indicates whether transmit UEQM is supported when
        the Tx MCS set may differ from the Rx MCS set, equal to false where no TX
        MCS set is defined or where the Tx MCS set is defined to be equal to the
        RX MCS set or when UEQM is not supported and the Tx MCS set may differ from
        the Rx MCS set, equal to true when UEQM is supported and the Tx MCS set may
        differ from the Rx MCS set. See 8.4.2.58.4"
    ::= { dot11RMNeighborReportEntry 44 }

dot11RMNeighborReportHTPCO OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The HT PCO indicates support for PCO, equal to false if not supported,
        equal to true if supported. See 8.4.2.58.5"
    ::= { dot11RMNeighborReportEntry 45 }

dot11RMNeighborReportHTPCOTransitionTime OBJECT-TYPE
    SYNTAX Unsigned32 (0..3)
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The HT PCO transition time indicates that the STA can switch between 20
        MHz channel width and 40 MHz channel width within the indicated time,
        equal to 0 for no transition, equal to 1 for 400 microseconds, equal to 2
        for 1.5 ms, equal to 3 for 5 ms. For the no transition case (equal to 0)
        the PCO active STA does not change its operation channel width and is able
        to receive 40 MHz PPDUs during the 20 MHz phase. See 8.4.2.58.5"
    ::= { dot11RMNeighborReportEntry 46 }

dot11RMNeighborReportHTMCSFeedback OBJECT-TYPE
    SYNTAX Unsigned32 (0..3)
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The HT MFB indicates the capability of the STA to provide MFB, equal to 0
        if the STA does not provide MFB, equal to 2 if the STA provide only unso-
        licited MFB, equal to 3 if the STA can provide MFB in response to MRQ as
        well as unsolicited MFB. Note the value 1 is reserved. See 8.4.2.58.5"
    ::= { dot11RMNeighborReportEntry 47 }

dot11RMNeighborReportHTCSupport OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The HT +HTC support indicates support of the HT Control field, equal to
        false if not supported, equal to true if supported. See 8.4.2.58.5"
    ::= { dot11RMNeighborReportEntry 48 }
```

```
dot11RMNeighborReportHTRDResponder OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      The HT RD responder indicates support for acting as a revere direction
      responder, equal to false if not supported, equal to true if supported.
      See 8.4.2.58.5"
   ::= { dot11RMNeighborReportEntry 49}

dot11RMNeighborReportHTImplictTransmitBeamformingReceivingCap OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      The HT implicit Transmit Beamforming receiving capable indicates whether
      this STA can receive Transmit Beamforming steered frames using implicit
      feedback, equal to false if not supported, equal to true if supported. See
      8.4.2.58.6"
   ::= { dot11RMNeighborReportEntry 50 }

dot11RMNeighborReportHTReceiveStaggeredSoundingCap OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      The HT receive staggered sounding capable indicates whether this STA can
      receive staggered sounding frames, equal to false if not supported, equal
      to true if supported. See 8.4.2.58.6"
   ::= { dot11RMNeighborReportEntry 51 }

dot11RMNeighborReportHTTransmitStaggeredSoundingCap OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      The HT transmit staggered sounding capable indicates whether this STA can
      transmit staggered sounding frames, equal to false if not supported, equal
      to true if supported. See 8.4.2.58.6"
   ::= { dot11RMNeighborReportEntry 52 }

dot11RMNeighborReportHTReceiveNDPCap OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      The HT Receive NDP capable indicates whether this receiver can interpret
      NDPs as sounding frames, equal to false if not supported, equal to true if
      supported. See 8.4.2.58.6"
   ::= { dot11RMNeighborReportEntry 53 }

dot11RMNeighborReportHTTransmitNDPCap OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      The HT Transmit NDP capable indicates whether this STA can transmit NDPs
      as sounding frames, equal to false if not supported, equal to true if sup-
      ported. See 8.4.2.58.6"
   ::= { dot11RMNeighborReportEntry 54 }

dot11RMNeighborReportHTImplicitTransmitBeamformingCap OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      The HT Implicit Transmit Beamforming capable indicates whether this STA
      can apply implicit transmit beamforming, equal to false if not supported,
      equal to true if supported. See 8.4.2.58.6"
   ::= { dot11RMNeighborReportEntry 55 }

dot11RMNeighborReportHTTransmitBeamformingCalibration OBJECT-TYPE
   SYNTAX Unsigned32 (0..3)
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      The HT Beamforming Calibration indicates that the STA can participate in a
      calibration procedure initiated by another STA that is capable of generat-
      ing an immediate response Sounding PPDU and can provide a CSI report in
      response to the receipt of a Sounding PPDU, equal to 0 if not supported,
      equal to 1 is the STA can respond to a calibration request using the CSI
      report but cannot initiate calibration, equal to 3 if the STA can both
      initiate and respond to a calibration request. See 8.4.2.58.6"
   ::= { dot11RMNeighborReportEntry 56 }

dot11RMNeighborReportHTExplicitCSITransmitBeamformingCap OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      The HT explicit CSI Transmit Beamforming capable indicates whether this
      STA can apply transmit beamforming using SCI explicit feedback in its
      transmission, equal to false if not supported, equal to true if supported.
      See 8.4.2.58.6"
   ::= { dot11RMNeighborReportEntry 57 }

dot11RMNeighborReportHTExplicitNonCompressedSteeringCap OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-create
   STATUS current
```

```
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The HT explicit noncompressed steering capable indicates whether this STA
        can apply transmit beamforming using noncompressed beamforming feedback
        matrix explicit feedback in its transmission, equal to false if not sup-
        ported, equal to true if supported. See 8.4.2.58.6"
    ::= { dot11RMNeighborReportEntry 58 }

dot11RMNeighborReportHTExplicitCompressedSteeringCap OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The HT explicit compressed steering capable indicates whether this STA can
        apply transmit beamforming using compressed beamforming feedback matrix
        explicit feedback in its transmission, equal to false if not supported,
        equal to true if supported. See 8.4.2.58.6"
    ::= { dot11RMNeighborReportEntry 59}

dot11RMNeighborReportHTExplicitTransmitBeamformingFeedback OBJECT-TYPE
    SYNTAX Unsigned32 (0..3)
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The HT explicit Transmit Beamforming CSI feedback indicates whether this
        receiver can return CSI explicit feedback, equal to 0 if not supported,
        equal to 1 for delayed feedback, equal to 2 for immediate feedback, equal
        to 3 for delayed and immediate feedback. See 8.4.2.58.6"
    ::= { dot11RMNeighborReportEntry 60 }

dot11RMNbRprtHTExplicitNonCompressedBeamformingFeedbackCap OBJECT-TYPE
    SYNTAX Unsigned32 (0..3)
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The HT Explicit noncompressed beamforming feedback capable indicates
        whether this receiver can return noncompressed beamforming feedback matrix
        explicit feedback, equal to 0 if not supported, equal to 1 for delayed
        feedback, equal to 2 for immediate feedback, equal to 3 for delayed and
        immediate feedback. See 8.4.2.58.6"
    ::= { dot11RMNeighborReportEntry 61 }

dot11RMNeighborReportHTExplicitCompressedBeamformingFeedbackCap OBJECT-TYPE
    SYNTAX Unsigned32 (0..3)
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The HT explicit compressed beamforming feedback capable indicates whether
        this receiver can return compressed beamforming feedback matrix explicit
        feedback, equal to 0 if not supported, equal to 1 for delayed feedback,
      equal to 2 for immediate feedback, equal to 3 for delayed and immediate
      feedback. See 8.4.2.58.6"
   ::= { dot11RMNeighborReportEntry 62 }

dot11RMNeighborReportHTTransmitBeamformingMinimalGrouping OBJECT-TYPE
   SYNTAX Unsigned32 (0..3)
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      The HT Transmit Beamforming minimal grouping indicates the minimal group-
      ing used for explicit feedback reports, equal to 0 if the STA supports
      groups of 1 (no grouping), equal to 1 to indicate groups of 1, 2, equal to
      2 to indicate groups of 1, 4, equal to 3 to indicate groups of 1, 2, 4. See
      8.4.2.58.6"
   ::= { dot11RMNeighborReportEntry 63 }

dot11RMNbRprtHTCSINumberofTxBeamformingAntennasSuppt OBJECT-TYPE
   SYNTAX Unsigned32 (0..3)
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      The HT CSI number of beamformer antennas supported indicates the maximum
      number of beamformer antennas the beamformee can support when CSI feedback
      is required, equal to 0 for single Tx antenna sounding, equal to 1 for 2
      Tx antenna sounding, equal to 2 for 3 Tx antenna sounding, equal to 3 for
      4 Tx antenna sounding. See 8.4.2.58.6"
   ::= { dot11RMNeighborReportEntry 64 }

dot11RMNbRprtHTNonCompressedSteeringNumofTxBmfmingAntennasSuppt OBJECT-TYPE
   SYNTAX Unsigned32 (0..3)
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      The HT noncompressed steering number of beamformer antennas supported
      indicates the maximum number of beamformer antennas the beamformee can
      support when noncompressed beamforming feedback matrix is required, equal
      to 0 for single Tx antenna sounding, equal to 1 for 2 Tx antenna sounding,
      equal to 2 for 3 Tx antenna sounding, equal to 3 for 4 Tx antenna sound-
      ing. See 8.4.2.58.6"
   ::= { dot11RMNeighborReportEntry 65 }

dot11RMNbRprtHTCompressedSteeringNumberofTxBmfmingAntennasSuppt OBJECT-TYPE
   SYNTAX Unsigned32 (0..3)
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      The HT compressed steering number of beamformer antennas supported indi-
      cates the maximum number of beamformer antennas the beamformee can support
      when compressed beamforming feedback matrix is required, equal to 0 for
      single Tx antenna sounding, equal to 1 for 2 Tx antenna sounding, equal to
      2 for 3 Tx antenna sounding, equal to 3 for 4 Tx antenna sounding. See
      8.4.2.58.6"
```

```
    ::= { dot11RMNeighborReportEntry 66 }

dot11RMNbRprtHTCSIMaxNumberofRowsTxBeamformingSuppt OBJECT-TYPE
    SYNTAX Unsigned32 (0..3)
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The HT CSI max number of rows beamformer supported indicates the maximum
        number of rows of CSI explicit feedback from the beamformee or calibration
        responder or transmit ASEL responder that a beamformer or calibration ini-
        tiator or transmit ASEL initiator can support when SCI feedback is
        required, equal to 0 for a single row of CSI, equal to 1 for 2 rows of CSI,
        equal to 2 for 3 rows of CSI, equal to 3 for 4 rows of CSI. See 8.4.2.58.6"
    ::= { dot11RMNeighborReportEntry 67 }

dot11RMNeighborReportHTTransmitBeamformingChannelEstimationCap OBJECT-TYPE
    SYNTAX Unsigned32 (0..3)
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The HT channel estimation capability indicates the maximum number of
        space-time streams for which channel dimensions can be simultaneously
        estimated when receiving an NDP sounding PPDU or the extension portion of
        the HT-LTFs in a staggered sounding PPDU. Equal to 0 for 1 space-time
        stream, equal to 1 for 2 space-time streams, equal to 2 for 3 space-time
        streams, equal to 3 for 4 space-time streams. See 8.4.2.58.6"
    ::= { dot11RMNeighborReportEntry 68 }

dot11RMNeighborReportHTAntSelectionCap OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The HT ASEL capable indicates whether this STA supports ASEL, equal to
        false if not supported, equal to true if supported. See 8.4.2.58.7"
    ::= { dot11RMNeighborReportEntry 69}

dot11RMNeighborReportHTExplicitCSIFeedbackBasedTxASELCap OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The HT explicit CSI feedback based transmit ASEL capable indicates whether
        this STA has transmit ASEL capability based on explicit CSI feedback,
        equal to false if not supported, equal to true if supported. See
        8.4.2.58.7"
    ::= { dot11RMNeighborReportEntry 70 }

dot11RMNeighborReportHTAntIndicesFeedbackBasedTxASELCap OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The HT antenna indices feedback based transmit ASEL capable indicates
        whether this STA has transmit ASEL capability based on antenna indices
        feedback, equal to false if not supported, equal to true if supported. See
        8.4.2.58.7"
    ::= { dot11RMNeighborReportEntry 71 }

dot11RMNeighborReportHTExplicitCSIFeedbackBasedCap OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The explicit CSI feedback capable indicates whether this STA can compute
        CSI and feedback in support of ASEL, equal to false if not supported,
        equal to true is supported. See 8.4.2.58.7"
    ::= { dot11RMNeighborReportEntry 72 }

dot11RMNeighborReportHTAntIndicesFeedbackCap OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The HT antenna indices feedback capable indicates whether this STA has Rx
        ASEL capability, equal to false if not supported, equal to true if sup-
        ported. See 8.4.2.58.7"
    ::= { dot11RMNeighborReportEntry 73 }

dot11RMNeighborReportHTRxASELCap OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The HT receive ASEL capable indicates whether this STA has Rx ASEL capa-
        bility, equal to false if not supported, equal to true if supported. See
        8.4.2.58.7"
    ::= { dot11RMNeighborReportEntry 74 }

dot11RMNeighborReportHTTxSoundingPPDUsCap OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The HT transmit sounding PPDUs capable indicates whether this STA can
        transmit sounding PPDUs for ASEL training per request, equal to false if
        not supported, equal to true if supported. See 8.4.2.58.7"
    ::= { dot11RMNeighborReportEntry 75 }

dot11RMNeighborReportHTInfoPrimaryChannel OBJECT-TYPE
    SYNTAX Unsigned32
```

```
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      The HT Info primary channel indicates the channel number of the primary
      channel, encoding: channel number of the primary channel. See 8.4.2.59"
   ::= { dot11RMNeighborReportEntry 76 }

dot11RMNeighborReportHTInfoSecChannelOffset OBJECT-TYPE
   SYNTAX Unsigned32 (0..3)
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      The HT Info secondary channel offset indicates the offset of the secondary
      channel relative to the primary channel, equal to 1 if the secondary chan-
      nel is above the primary channel, equal to 3 if the secondary channel is
      below the primary channel, equal to 0 if no secondary channel is present.
      The value 2 is reserved. See 8.4.2.59"
   ::= { dot11RMNeighborReportEntry 77 }

dot11RMNeighborReportHTInfoSTAChannelWidth OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      The HT Info STA channel width defines the channel widths that may be used
      to transmit to the STA, equal to false for a 20 MHz channel width, equal
      to true allows use of any channel width in the supported channel width
      set. See 8.4.2.59"
   ::= { dot11RMNeighborReportEntry 78 }

dot11RMNeighborReportHTInfoRIFSMode OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      The HT Info RIFS mode indicates whether use of RIFS is permitted within
      the BSS, equal to false if use of RIFS is prohibited, equal to true if use
      of RIFS is permitted. See 8.4.2.59"
   ::= { dot11RMNeighborReportEntry 79}

dot11RMNeighborReportHTInfoProtection OBJECT-TYPE
   SYNTAX Unsigned32 (0..3)
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      The HT Info protection indicates protection requirements of HT transmis-
      sions. Equal to 0 if all STAs detected in the primary or the secondary
      channel or that are a member of this BSS are HT STAs and either all STAs
      that are known by the transmitting STA to be a member of this BSS are 20/
```

```
        40 MHz HT in a 20/40 MHz BSS or this BSS is a 20 MHz BSS. Equal to 1 (non-
        member protection mode) if there is at least one non-HT STA detected in
        either the primary or the secondary channel or in both the primary and
        secondary channels and that is not known by the transmitting STA to be a
        member of this BSS and all STAs that are known by the transmitting STA to
        be a member of this BSS are HT STAs. Equal to 2 if all STAs detected in the
        primary or the secondary channel or that are known by the transmitting STA
        to be a member of this BSS are HT STAs and this BSS is a 20/40 MHz BSS and
        there is at least one 20 MHz HT STA associated with this BSS. Equal to 3
        (non-HT mixed mode) otherwise. See 8.4.2.58.2"
    ::= { dot11RMNeighborReportEntry 80 }

dot11RMNeighborReportHTInfoNonGreenfieldHTSTAsPresent OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The HT Info nongreenfield HT STAs present indicates if any HT STAs that
        are not HT-greenfield capable have associated. Determines when a non-AP
        STA should use HT-greenfield protection. Present in Beacon and Probe
        Response frames transmitted by an AP. Equal to false if all HT STAs that
        are associated are HT-greenfield capable, equal to true if one or more HT
        STAs that are not HT-greenfield capable are associated. See 8.4.2.59"
    ::= { dot11RMNeighborReportEntry 81 }

dot11RMNeighborReportHTInfoOBSSNonHTSTAsPresent OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The HT Info OBSS non-HT STAs present indicates if the use of protection
        for non-HT STAs by OBSSs is determined to be desirable. Present in Beacon
        and Probe Response frames transmitted by an AP, equal to true if the use
        of protection for non-HT STAs by OBSSs is determined to be desirable,
        equal to false otherwise. See 8.4.2.59"
    ::= { dot11RMNeighborReportEntry 82 }

dot11RMNeighborReportHTInfoDualBeacon OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The HT Info dual beacon indicates whether the AP transmits an STBC beacon,
        equal to false if no STBC beacon is transmitted, equal to true if an STBC
        beacon is transmitted. See 8.4.2.59"
    ::= { dot11RMNeighborReportEntry 83 }

dot11RMNeighborReportHTInfoDualCTSProtection OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.
```

```
      The HT Info dual CTS protection is used by the AP to set a NAV at STAs that
      do not support STBC and at STAs that can associate solely through the sec-
      ondary beacon, equal to false if dual CTS protection is not required,
      equal to true if dual CTS protection is required. See 8.4.2.59"
   ::= { dot11RMNeighborReportEntry 84 }

dot11RMNeighborReportHTInfoSTBCBeacon OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      The HT Info STBC beacon indicates whether the beacon containing this ele-
      ment is a primary or a STBC beacon, equal to false in a primary beacon,
      equal to true in a STBC beacon. See 8.4.2.59"
   ::= { dot11RMNeighborReportEntry 85 }

dot11RMNeighborReportHTInfoLSIGTXOPProtectionSup OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      The HT Info L-SIG TXOP protection full support indicates whether all HT
      STA in the BSS support L-SIG TXOP protection, equal to false if one or
      more HT STA in the BSS do not support L-SIG TXOP protection, equal to true
      if all HT STA in the BSS support L-SIG TXOP protection. See 8.4.2.59"
   ::= { dot11RMNeighborReportEntry 86 }

dot11RMNeighborReportHTInfoPCOActive OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      The HT Info PCO active indicates whether PCO is active in the BSS, equal
      to false if PCO is not active in the BSS, equal to true if PCO is active
      in the BSS. See 8.4.2.59"
   ::= { dot11RMNeighborReportEntry 87}

dot11RMNeighborReportHTInfoPCOPhase OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a measurement report is completed.

      The HT Info PCO phase indicates the PCO phase of operation, equal to false
      indicates a switch to or continued 20 MHz phase, equal to true indicates a
      switch to or continuation of 40 MHz phase. See 8.4.2.59"
   ::= { dot11RMNeighborReportEntry 88 }

dot11RMNeighborReportHTInfoBasicMCSSet OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(16))
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
```

```
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The HT Info Basic MCS Set indicates values that are supported by all HT
        STAs in the BSS. The Basic MCS Set is a bitmap of size 128 bits. Bit 0 cor-
        responds to MCS 0. A bit is equal to 1 to indicate support for that MCS,
        equal to 0 otherwise. See 8.4.2.59"
    ::= { dot11RMNeighborReportEntry 89 }

dot11RMNeighborReportHTSecChannelOffset OBJECT-TYPE
    SYNTAX Unsigned32 (0..3)
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The HT secondary channel offset indicates the position of the secondary
        channel relative to the primary channel, equal to 1 to indicate that the
        secondary channel is above the primary channel, equal to 3 to indicate the
        secondary channel is below the primary channel, equal to 0 to indicate
        that no secondary channel is present. The value 2 is reserved. See
        8.4.2.22"
    ::= { dot11RMNeighborReportEntry 90 }

dot11RMNeighborReportExtCapPSMPSupport OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The Extended Capabilities PSMP support indicates support for PSMP opera-
        tion, equal to false if PSMP is not supported, equal to true if PSMP oper-
        ation is supported. See 8.4.2.29"
    ::= { dot11RMNeighborReportEntry 91 }

dot11RMNeighborReportExtCapSPSMPSup OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The HT Info S-PSMP support indicates support for scheduled PSMP, equal to
        false when PSMP is supported is equal to false and when PSMP support is
        equal to 1 if the STA does not support S-PSMP, equal to true when PSMP
        support is equal to 1 if the STA supports S-PSMP. See 8.4.2.29"
    ::= { dot11RMNeighborReportEntry 92 }

dot11RMNeighborReportExtCapServiceIntervalGranularity OBJECT-TYPE
    SYNTAX Unsigned32 (0..7)
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a measurement report is completed.

        The Extended Capabilities SI granularity indicates the duration of the
        shortest SI, equal to 0 for 5 ms, equal to 1 for 10 ms, equal to 2 for 15
        ms, equal to 3 for 20 ms, equal to 4 for 25 ms, equal to 5 for 30 ms, equal
        to 6 for 35 ms, equal to 7 for 40 ms. See 8.4.2.29"
    ::= { dot11RMNeighborReportEntry 93 }

dot11RMNeighborReportBSSTransitCandPreference OBJECT-TYPE
    SYNTAX Unsigned32 (0..255)
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This attribute indicates the network preference for BSS transition to the
        BSS listed in this BSS Transition Candidate List Entries field in the BSS
        Transition Management Request frame, BSS Transition Management Query frame
        and BSS Transition Management Response frame. The Preference field value
        is a number ranging from 0 to 255 indicating an ordering of preferences
        for the BSS transition candidates for this STA. The value 0 indicates an
        excluded BSS. The values 1-255 the preferred relative ordering of BSSs,
        with 255 indicating the most preferred candidate and 1 indicating the
        least preferred candidate. Additional details describing use of the Pref-
        erence field are provided in 10.23.6.3."
    ::= { dot11RMNeighborReportEntry 94 }

dot11RMNeighborReportBSSTerminationTSF OBJECT-TYPE
    SYNTAX TSFType
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This attribute indicates the value of the TSF counter when the BSS termi-
        nation will occur in the future. A BSS Termination TSF field value of 0
        indicates that termination of the BSS will occur imminently. Prior to ter-
        mination of the BSS, all associated STAs are disassociated by the AP."
    ::= { dot11RMNeighborReportEntry 95 }

dot11RMNeighborReportBSSTerminationDuration OBJECT-TYPE
    SYNTAX Unsigned32 (1..65535)
    UNITS "minutes"
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This attribute indicates the number of minutes for which the BSS is not
        present. The Duration field value of 0 is reserved. The Duration field
        value is set to 65 535 when the BSS is terminated for a period longer than
        or equal to 65 535 minutes."
    ::= { dot11RMNeighborReportEntry 96 }

-- ********************************************************************
-- * End of dot11RMNeighborReport TABLE
-- ********************************************************************

-- ********************************************************************
-- * END of Radio Measurement Interface MIB
-- ********************************************************************

-- ********************************************************************
-- * Wireless Network Management Interface MIB
-- ********************************************************************

dot11WirelessNetworkManagement OBJECT IDENTIFIER ::= { dot11smt 22 }

-- ********************************************************************
-- * Wireless network management requests
-- ********************************************************************

dot11WNMRequest OBJECT IDENTIFIER ::= { dot11WirelessNetworkManagement 1 }

-- ********************************************************************
-- * dot11WNMRequest TABLE
-- **********************************************************************
dot11WNMRequestNextIndex OBJECT-TYPE
   SYNTAX Unsigned32(0..4294967295)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when able to accept a new request.

      Identifies a hint for the next value of dot11WNMRqstIndex to be used in a
      row creation attempt for dot11WNMRequestTable. If no new rows can be cre-
      ated for some reason, such as memory, processing requirements, etc, the
      SME shall set this attribute to 0. It shall update this attribute to a
      proper value other than 0 as soon as it is capable of receiving new mea-
      surement requests. The nextIndex is not necessarily sequential nor mono-
      tonically increasing."
   ::= { dot11WNMRequest 1 }

dot11WNMRequestTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11WNMRequestEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "This group contains the current list of requests for WNM reports to be
      issued and have been issued until removed. A network manager adds a WNM
      request by creating a row with createAndWait row status and then filling
      in the request parameters/attributes. The request becomes active to be
      issued when the row status is set to Active. The columnar objects or
      attributes other than the rowStatus shall not be written if the rowStatus
      is Active. The request rows can be deleted, if commanded by a network man-
      ager via changing the value of dot11WNMRqstRowStatus to Destroy. This may
      leave orphaned rows if a manager crashes and forgets which rows are being
      used by it. One recommended way to manage orphaned or finished rows is to
      delete rows if their dot11WNMRqstRowStatus remains other than Active for
      longer than a period (recommend at least 5 minutes, IETF RFC 2579). Or
      another recommended way is to delete older rows as needed based on their
      dot11WNMRqstTimeStamp values. This can be done by the agent as well as the
      manager."
   ::= { dot11WNMRequest 2 }

dot11WNMRequestEntry OBJECT-TYPE
   SYNTAX Dot11WNMRequestEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "An entry in the dot11WNMRequestTable Indexed by dot11WNMRqstIndex."
   INDEX { dot11WNMRqstIndex }
   ::= { dot11WNMRequestTable 1 }

Dot11WNMRequestEntry ::=
   SEQUENCE {
      dot11WNMRqstIndex                                      Unsigned32,
      dot11WNMRqstRowStatus                                  RowStatus,
      dot11WNMRqstToken                                      OCTET STRING,
      dot11WNMRqstIfIndex                                    InterfaceIndex,
      dot11WNMRqstType                                       INTEGER,
      dot11WNMRqstTargetAdd                                  MacAddress,
      dot11WNMRqstTimeStamp                                  TimeTicks,
      dot11WNMRqstRndInterval                                Unsigned32,
      dot11WNMRqstDuration                                   Unsigned32,
      dot11WNMRqstMcstGroup                                  MacAddress,
      dot11WNMRqstMcstTrigCon                                OCTET STRING,
      dot11WNMRqstMcstTrigInactivityTimeout                  Unsigned32,
      dot11WNMRqstMcstTrigReactDelay                         Unsigned32,
        dot11WNMRqstLCRRqstSubject                          INTEGER,
        dot11WNMRqstLCRIntervalUnits                        INTEGER,
        dot11WNMRqstLCRServiceInterval                      Unsigned32,
        dot11WNMRqstLIRRqstSubject                          INTEGER,
        dot11WNMRqstLIRIntervalUnits                        INTEGER,
        dot11WNMRqstLIRServiceInterval                      Unsigned32,
        dot11WNMRqstEventToken                              Unsigned32,
        dot11WNMRqstEventType                               INTEGER,
        dot11WNMRqstEventResponseLimit                      Unsigned32,
        dot11WNMRqstEventTargetBssid                        MacAddress,
        dot11WNMRqstEventSourceBssid                        MacAddress,
        dot11WNMRqstEventTransitTimeThresh                  Unsigned32,
        dot11WNMRqstEventTransitMatchValue                  OCTET STRING,
        dot11WNMRqstEventFreqTransitCountThresh             Unsigned32,
        dot11WNMRqstEventFreqTransitInterval                Unsigned32,
        dot11WNMRqstEventRsnaAuthType                       OCTET STRING,
        dot11WNMRqstEapType                                 Unsigned32,
        dot11WNMRqstEapVendorId                             OCTET STRING,
        dot11WNMRqstEapVendorType                           OCTET STRING,
        dot11WNMRqstEventRsnaMatchValue                     OCTET STRING,
        dot11WNMRqstEventPeerMacAddress                     MacAddress,
        dot11WNMRqstOperatingClass                          Unsigned32,
        dot11WNMRqstChanNumber                              Unsigned32,
        dot11WNMRqstDiagToken                               Unsigned32,
        dot11WNMRqstDiagType                                INTEGER,
        dot11WNMRqstDiagTimeout                             Unsigned32,
        dot11WNMRqstDiagBssid                               MacAddress,
        dot11WNMRqstDiagProfileId                           Unsigned32,
        dot11WNMRqstDiagCredentials                         INTEGER,
        dot11WNMRqstLocConfigLocIndParams                   OCTET STRING,
        dot11WNMRqstLocConfigChanList                       OCTET STRING,
        dot11WNMRqstLocConfigBcastRate                      Unsigned32,
        dot11WNMRqstLocConfigOptions                        OCTET STRING,
        dot11WNMRqstBssTransitQueryReason                   INTEGER,
        dot11WNMRqstBssTransitReqMode                       OCTET STRING,
        dot11WNMRqstBssTransitDisocTimer                    Unsigned32,
        dot11WNMRqstBssTransitSessInfoURL                   OCTET STRING,
        dot11WNMRqstBssTransitCandidateList                 OCTET STRING,
        dot11WNMRqstColocInterfAutoEnable                   TruthValue,
        dot11WNMRqstColocInterfRptTimeout                   Unsigned32,
        dot11WNMRqstVendorSpecific                          OCTET STRING,
        dot11WNMRqstDestinationURI                          OCTET STRING
    }

dot11WNMRqstIndex OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Index for WNM Request elements in dot11WNMRequestTable, greater than 0."
    ::= { dot11WNMRequestEntry 1 }

dot11WNMRqstRowStatus OBJECT-TYPE
    SYNTAX RowStatus
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when requesting a measure-
        ment, and by the SME when accepting a management request.

        The Row Status column of the current row, used for tracking status of an
        individual request. When this attribute is set to Active, AND a measure-
        ment request can be unambiguously created based on the parameters in the
        row, then the MLME may proceed to issue the request to its intended tar-
        gets when appropriate. If not, this attribute may be set to Not-ready
        immediately to indicate parametric errors. However, it is the network man-
        agers
        responsibility to correct the error. If the request is successfully issued
        to the target STA, then the rowStatus is set to notInService."
    ::= { dot11WNMRequestEntry 2 }

dot11WNMRqstToken OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when the table entry is
        created, i.e., when requesting a measurement. Changes take effect when
        dot11RMRqstRowStatus is set to Active.

        This attribute indicates a unique string to identify this request. To
        guarantee the uniqueness of this token across multiple network managers,
        it is recommended that this token be prefixed with the IP address of the
        network manager creating this row. This token is not necessarily equiva-
        lent to the measurement tokens in WNM request frames."
    ::= { dot11WNMRequestEntry 3 }

dot11WNMRqstIfIndex OBJECT-TYPE
    SYNTAX InterfaceIndex
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "The ifIndex for this row of WNM Request to be issued on."
    ::= { dot11WNMRequestEntry 4 }

dot11WNMRqstType OBJECT-TYPE
    SYNTAX INTEGER {
        mcastDiagnostics(0),
        locationCivic(1),
        locationIdentifier(2),
        event(3),
        dignostic(4),
        locationConfiguration(5),
        bssTransitionQuery(6),
        bssTransitionRqst(7),
        fms(8),
        colocInterference(9)
        }
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when making a management
        request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

        This attribute indicates the request type of this WNM request row."
    ::= { dot11WNMRequestEntry 5 }

dot11WNMRqstTargetAdd OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when making a management
        request. Changes take effect when dot11WNMRqstRowStatus is set to Active.
```

```
      The MAC address of STA for this row of WNM Request is to be issued to. If
      this attribute matches the MAC address of the dot11WNMRqstIfIndex, then
      measurement request is for this STA itself to carry out."
   ::= { dot11WNMRequestEntry 6 }

dot11WNMRqstTimeStamp OBJECT-TYPE
   SYNTAX TimeTicks
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity when making a management
      request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

      This attribute indicates the SysUpTime Value the last time when the
      dot11WNMRqstRowStatus is set to active or when this row is created the
      first time. This attribute shall be set by this STA or AP automatically,
      not by an SNMP manager."
   ::= { dot11WNMRequestEntry 7 }

dot11WNMRqstRndInterval OBJECT-TYPE
   SYNTAX Unsigned32
   UNITS "TUs"
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity when making a management
      request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

      This attribute indicates the upper bound of the random delay to be used
      prior to making the measurement, expressed in units of TUs. See 10.11.3."
   DEFVAL { 0 }
   ::= { dot11WNMRequestEntry 8 }

dot11WNMRqstDuration OBJECT-TYPE
   SYNTAX Unsigned32
   UNITS "TUs"
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity when making a management
      request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

      This attribute indicates the preferred or mandatory measurement duration
      for this Measurement Request."
   DEFVAL { 0 }
   ::= { dot11WNMRequestEntry 9 }

dot11WNMRqstMcstGroup OBJECT-TYPE
   SYNTAX MacAddress
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity when making a management
      request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

      Multicast Group address indicates the MAC address of the multicast group
      for which diagnostics are requested. The BSSID shall be set to the wild-
      card BSSID when the measurement is to be performed on any mulitical group
      on the operating channel. This attribute is only valid if the
```

```
        dot11WNMRqstType is 10, indicating a multicast diagnostic request, and is
        ignored otherwise."
    DEFVAL { 'FFFFFFFFFFFF'H }
    ::= { dot11WNMRequestEntry 10 }

dot11WNMRqstMcstTrigCon OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(1))
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when making a management
        request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

        This attribute indicates the trigger condition for the Multicast Diagnos-
        tic request."
    ::= { dot11WNMRequestEntry 11 }

dot11WNMRqstMcstTrigInactivityTimeout OBJECT-TYPE
    SYNTAX Unsigned32 (1..255)
    UNITS "100 TUs"
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when making a management
        request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

        This attribute indicates the time interval value in units of 100 TU to be
        use as the threshold value for Trigger Inactivity Timeout trigger condi-
        tion."
    ::= { dot11WNMRequestEntry 12 }

dot11WNMRqstMcstTrigReactDelay OBJECT-TYPE
    SYNTAX Unsigned32 (1..255)
    UNITS "100 TUs"
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when making a management
        request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

        This attribute indicates the time interval value in units of 100 TU during
        which a measuring STA does not generate further Multicast Triggered
        Reports after a trigger condition has been met."
    ::= { dot11WNMRequestEntry 13 }

dot11WNMRqstLCRRqstSubject OBJECT-TYPE
    SYNTAX INTEGER {
        local(0),
        remote(1)
        }
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when making a management
        request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

        The attribute indicates the subject of the Location Civic Request."
    DEFVAL { 0 }
    ::= { dot11WNMRequestEntry 14 }
```

```
dot11WNMRqstLCRIntervalUnits OBJECT-TYPE
    SYNTAX INTEGER {
        seconds(0),
        minutes(1),
        hours(2)
        }
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when making a management
        request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

        This attribute indicates the units used in the Location Civic Request Ser-
        vice Interval."
    ::= { dot11WNMRequestEntry 15 }

dot11WNMRqstLCRServiceInterval OBJECT-TYPE
    SYNTAX Unsigned32 (0..65535)
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when making a management
        request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

        This attribute indicates the time interval, expressed in the units indi-
        cated in the Location Civic Request Service Interval Units field, at which
        the STA requests to receive Location Civic Reports.  A Location Civic
        Request Service Interval of 0 indicates that only a single Location Civic
        Report is requested."
    ::= { dot11WNMRequestEntry 16 }

dot11WNMRqstLIRRqstSubject OBJECT-TYPE
    SYNTAX INTEGER {
        local(0),
        remote(1)
        }
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when making a management
        request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

        The attribute indicates the subject of the Location Identifier Request."
    DEFVAL { 0 }
    ::= { dot11WNMRequestEntry 17 }

dot11WNMRqstLIRIntervalUnits OBJECT-TYPE
    SYNTAX INTEGER {
        seconds(0),
        minutes(1),
        hours(2)
        }
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when making a management
        request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

        This attribute indicates the units used in the Location Identifier Request
        Service Interval."
```

```
   ::= { dot11WNMRequestEntry 18 }

dot11WNMRqstLIRServiceInterval OBJECT-TYPE
   SYNTAX Unsigned32 (0..65535)
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity when making a management
      request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

      This attribute indicates the time interval, expressed in the units indi-
      cated in the Location Identifier Request Interval Units field, at which
      the STA requests to receive Location Identifier Reports. A Location Iden-
      tifier Request Service Interval of 0 indicates that only a single Location
      Identifier Report is requested."
   ::= { dot11WNMRequestEntry 19 }

dot11WNMRqstEventToken OBJECT-TYPE
   SYNTAX Unsigned32 (1..255)
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity when making a management
      request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

      This attribute indicates a unique string to identify this request."
   ::= { dot11WNMRequestEntry 20 }

dot11WNMRqstEventType OBJECT-TYPE
   SYNTAX INTEGER {
      transition(0),
      rsna(1),
      peerToPeer(2),
      wnmLog(3),
      vendorSpecific(221)
      }
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity when making a management
      request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

      This attribute indicates the request type of this WNM Event request."
   ::= { dot11WNMRequestEntry 21 }

dot11WNMRqstEventResponseLimit OBJECT-TYPE
   SYNTAX Unsigned32 (0..255)
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity when making a management
      request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

      This attribute indicates the maximum number of requested Event Reports to
      be included in the Event Report element. A value of 0 indicates that no
      limit is set on the number of Event Reports to be included in the Event
      Report element."
   ::= { dot11WNMRequestEntry 22 }

dot11WNMRqstEventTargetBssid OBJECT-TYPE
```

```
    SYNTAX MacAddress
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
       "This is a control variable.
       It is written by an external management entity when making a management
       request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

       This attribute is used to request that a Transition or RSNA Event Report
       includes the event entry when the target BSSID is equal to the indicated
       BSSID. A transition event is a STA movement or attempted movement from one
       BSS (the source BSS) in one ESS to another BSS (the target BSS) within the
       same ESS. The BSSID shall be set to the wildcard BSSID when the transi-
       tions to any BSSID is requested."
    DEFVAL { 'FFFFFFFFFFFF'H }
    ::= { dot11WNMRequestEntry 23 }

dot11WNMRqstEventSourceBssid OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
       "This is a control variable.
       It is written by an external management entity when making a management
       request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

       This attribute is used to request that a Transition Event Report includes
       the transition event entry when the source BSSID is equal to the indicated
       BSSID. A transition event is a STA movement or attempted movement from one
       BSS (the source BSS) in one ESS to another BSS (the target BSS) within the
       same ESS. The BSSID shall be set to the wildcard BSSID when the transi-
       tions from any BSSID is requested."
    DEFVAL { 'FFFFFFFFFFFF'H }
    ::= { dot11WNMRequestEntry 24 }

dot11WNMRqstEventTransitTimeThresh OBJECT-TYPE
    SYNTAX Unsigned32 (0..65535)
    UNITS "TUs"
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
       "This is a control variable.
       It is written by an external management entity when making a management
       request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

       This attribute indicates a value representing the transition time to be
       used as the threshold value for the Transition Time condition in TUs. The
       Transition Time is defined in 10.23.2.2"
    ::= { dot11WNMRequestEntry 25 }

dot11WNMRqstEventTransitMatchValue OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(1))
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
       "This is a control variable.
       It is written by an external management entity when making a management
       request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

       This attribute indicates a request for the specified transition results
       that match the bit descriptions of this field. b0 indicates match when
       transition is successful. b1 indicates match when transition fails."
    ::= { dot11WNMRequestEntry 26 }
```

```
dot11WNMRqstEventFreqTransitCountThresh OBJECT-TYPE
   SYNTAX Unsigned32 (0..255)
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity when making a management
      request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

      This attribute indicates the minimum number of matching transitions
      detected in the measurement duration to generate a Transition Event
      Report."
   ::= { dot11WNMRequestEntry 27 }

dot11WNMRqstEventFreqTransitInterval OBJECT-TYPE
   SYNTAX Unsigned32 (0..65535)
   UNITS "TUs"
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity when making a management
      request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

      This attribute indicates the sliding window time interval, in TUs, during
      which the STA detects matching transitions to determine if the Frequent
      Transition Count Threshold is exceeded in order to generate a Transition
      Event Report."
   ::= { dot11WNMRequestEntry 28 }

dot11WNMRqstEventRsnaAuthType OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(4))
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity when making a management
      request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

      This attribute is used to request that an RSNA Event Report include the
      event entry when its RSNA Authentication Type matches the indicated RSNA
      authentication type value."
   ::= { dot11WNMRequestEntry 29 }

dot11WNMRqstEapType OBJECT-TYPE
   SYNTAX Unsigned32 (0..255)
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity when making a management
      request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

      This attribute is used to request that an RSNA Event Report include the
      event entry when its EAP Type matches the indicated EAP type value. Valid
      EAP Type numbers are assigned by IANA and are defined at http://
      www.iana.org/assignments/eap-numbers."
   ::= { dot11WNMRequestEntry 30 }

dot11WNMRqstEapVendorId OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(0..3))
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
```

```
        "This is a control variable.
        It is written by an external management entity when making a management
        request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

        This attribute is used to request that an RSNA Event Report include the
        event entry when its EAP Vendor ID matches the indicated vendor ID value.
        The EAP Vendor ID field is included when the EAP Type field is set to 254,
        and is excluded otherwise."
    ::= { dot11WNMRequestEntry 31 }

dot11WNMRqstEapVendorType OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(0..4))
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when making a management
        request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

        This attribute is used to request that an RSNA Event Report include the
        event entry when its EAP Vendor Type matches the indicated EAP vendor type
        value. The EAP Vendor ID field is included when the EAP Type field is set
        to 254, and is excluded otherwise."
    ::= { dot11WNMRequestEntry 32 }

dot11WNMRqstEventRsnaMatchValue OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(1))
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when making a management
        request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

        This attribute indicates a request for the specified transition results
        that match the bit descriptions of this field. b0 (least significant bit)
        indicates match when RSNA is successful. b1 indicates match when RSNA
        fails."
    ::= { dot11WNMRequestEntry 33 }

dot11WNMRqstEventPeerMacAddress OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when making a management
        request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

        This attribute is used to request that a Peer-to-Peer Event Report
        includes the transition event entry when the MAC address of the peer STA
        or IBSS BSSID is equal to the indicated MAC address. The MAC address shall
        be set to the wildcard BSSID when the transitions from any peer STA or
        IBSS BSSID is requested."
    DEFVAL { 'FFFFFFFFFFFF'H }
    ::= { dot11WNMRequestEntry 34 }

dot11WNMRqstOperatingClass OBJECT-TYPE
    SYNTAX Unsigned32(1..255)
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when making a management
```

```
      request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

      This attribute indicates the channel set for this WNM request. Country,
      Operating Class and Channel Number together specify the channel frequency
      and spacing for this measurement request. Valid values of Operating Class
      are shown in Annex E."
   ::= { dot11WNMRequestEntry 35 }

dot11WNMRqstChanNumber OBJECT-TYPE
   SYNTAX Unsigned32 (1..255)
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity when making a management
      request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

      This attribute indicates the current operating channel for this WNM
      request. The Channel Number is only defined within the indicated Operating
      Class as shown in Annex E."
   ::= { dot11WNMRequestEntry 36 }

dot11WNMRqstDiagToken OBJECT-TYPE
   SYNTAX Unsigned32 (1..255)
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity when making a management
      request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

      This attribute indicates a unique string to identify this request."
   ::= { dot11WNMRequestEntry 37 }

dot11WNMRqstDiagType OBJECT-TYPE
   SYNTAX INTEGER {
      cancelRequest(0),
      manufacturerInfoStaRep(1),
      configurationProfile(2),
      associationDiag(3),
      ieee8021xAuthDiag(4),
      vendorSpecific(221)
      }
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity when making a management
      request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

      This attribute indicates the request type of this WNM Diagnostic request."
   ::= { dot11WNMRequestEntry 38 }

dot11WNMRqstDiagTimeout OBJECT-TYPE
   SYNTAX Unsigned32 (0..65535)
   UNITS "seconds"
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity when making a management
      request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

      This attribute indicates a value representing the time interval after a
```

```
       Diagnostic Report is generated during which no additional Diagnostic
       Reports shall be sent."
    ::= { dot11WNMRequestEntry 39 }

dot11WNMRqstDiagBssid OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
       "This is a control variable.
       It is written by an external management entity when making a management
       request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

       This attribute indicates a request for a Diagnostic Report from the indi-
       cated BSSID. The BSSID shall be set to the wildcard BSSID when diagnostics
       from any BSSID is requested."
    DEFVAL { 'FFFFFFFFFFFF'H }
    ::= { dot11WNMRequestEntry 40 }

dot11WNMRqstDiagProfileId OBJECT-TYPE
    SYNTAX Unsigned32 (1..255)
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
       "This is a control variable.
       It is written by an external management entity when making a management
       request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

       This attribute indicates a unique identifier for referencing a configura-
       tion profile available on a device. The value of the identifier can be any
       arbitrary value, as long as it is uniquely associated to a single config-
       uration profile on the device sending the identifier."
    ::= { dot11WNMRequestEntry 41 }

dot11WNMRqstDiagCredentials OBJECT-TYPE
    SYNTAX INTEGER {
       none(0),
       preSharedKey(1),
       usernamePassword(2),
       x509Certificate(3),
       otherCertificate(4),
       oneTimePassword(5),
       token(6)
       }
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
       "This is a control variable.
       It is written by an external management entity when making a management
       request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

       This attribute indicates the type of credential used for the IEEE 802.1X
       authentication."
    ::= { dot11WNMRequestEntry 42 }

dot11WNMRqstLocConfigLocIndParams OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(16))
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
       "This is a control variable.
       It is written by an external management entity when making a management
       request. Changes take effect when dot11WNMRqstRowStatus is set to Active.
```

```
      This attribute indicates STA Location reporting characteristics. The for-
      mat of these Location Indication Parameters are detailed in 8.4.2.73.2."
   ::= { dot11WNMRequestEntry 43 }

dot11WNMRqstLocConfigChanList OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(0..252))
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity when making a management
      request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

      This attribute lists location reporting channel information for this Loca-
      tion Configuration request. The default value is null. Each pair of octets
      indicates a different operating class and channel number for this request.
      The detailed format for this list of channels is described in 8.4.2.73.3."
   DEFVAL { ''H }
   ::= { dot11WNMRequestEntry 44 }

dot11WNMRqstLocConfigBcastRate OBJECT-TYPE
   SYNTAX Unsigned32 (0..65535)
   UNITS "0.5Mb/s"
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity when making a management
      request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

      This attribute indicates the target data rate, in 0.5Mb/s units, at which
      the STA transmits Location Track Notification frames. A value of 0 indi-
      cates the STA transmits Location Track Notification frames at a rate cho-
      sen by the STA transmitting the Location Track Notification frames."
   ::= { dot11WNMRequestEntry 45 }

dot11WNMRqstLocConfigOptions OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(0..255))
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity when making a management
      request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

      This attribute indicates the location track indication options used; see
      8.4.2.73.9."
   DEFVAL { ''H }
   ::= { dot11WNMRequestEntry 46 }

dot11WNMRqstBssTransitQueryReason OBJECT-TYPE
   SYNTAX INTEGER {
      unspecified(0),
      excessiveFrameLossRatesPoorConditions(1),
      excessiveDelayForCurrentTrafficStreams(2),
      insufficientQosCapacityForCurrentTrafficStreams(3),
      firstAssociationToEss(4),
      loadBalancing(5),
      betterApFound(6),
      deauthenticatedDisassociatedFromPreviousAp(7),
      apFailedIeee8021XEapAuthentication(8),
      apFailed4wayHandshake(9),
      receivedTooManyReplayCounterFailures(10),
      receivedTooManyDataMICFailures(11),
        exceededMaxNumberOfRetransmissions(12),
        receivedTooManyBroadcastDisassociations(13),
        receivedTooManyBroadcastDeauthentications(14),
        previousTransitionFailed(15),
        lowRSSI(16)
        }
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when making a management
        request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

        This attribute indicates the reason for the BSS Transition Query. The
        format for this list of reasons is further detailed in 8.4.2.70.2."
    ::= { dot11WNMRequestEntry 47 }

dot11WNMRqstBssTransitReqMode OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(1))
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when making a management
        request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

        This attribute indicates the type of BSS request transition. b0 (least
        significant bit) indicates the Preferred Candidate list is included in
        this frame. b1 indicates an abridged format for all BSSIDs not listed in
        this frame. b2 indicates that the STA will be disassociated for the cur-
        rent AP. b3 indicates the BSS is shutting down and that the STA will be
        disassociated. b4 indicates that the will be disassociated from the ESS.
        The format for this field is detailed in 8.5.14.9."
    ::= { dot11WNMRequestEntry 48 }

dot11WNMRqstBssTransitDisocTimer OBJECT-TYPE
    SYNTAX Unsigned32 (0..65535)
    UNITS "TBTTs"
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when making a management
        request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

        This attribute indicates the number of beacon transmission times (TBTTs)
        until the serving AP sends a Disassociation frame to this STA. The value 0
        indicates unknown. If the Disassociation Imminent bit of the Request Mode
        field is set to 0, this field is ignored."
    ::= { dot11WNMRequestEntry 49 }

dot11WNMRqstBssTransitSessInfoURL OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity when making a management
        request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

        This attribute contains a variable-length field formatted in accordance
        with IETF RFC 3986-2005."
    ::= { dot11WNMRequestEntry 50 }
```

```
dot11WNMRqstBssTransitCandidateList OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(0..2304))
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity when making a management
      request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

      This attribute lists one or more Neighbor Report elements described in
      8.4.2.39. If the STA has no Transition Candidate information in response
      to the BSS Transition Management Query frame, the candidate list is null.
      "
   ::= { dot11WNMRequestEntry 51 }

dot11WNMRqstColocInterfAutoEnable OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity when making a management
      request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

      This attribute, when true, indicates that the requesting STA requests the
      receiving STA to send the Collocated Interference Response frames period-
      ically with the Report Period interval, as defined in 8.5.14.13, or when
      the STA detects a change in the collocated interference."
   ::= { dot11WNMRequestEntry 52 }

dot11WNMRqstColocInterfRptTimeout OBJECT-TYPE
   SYNTAX Unsigned32 (0..127)
   UNITS "100 TUs"
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity when making a management
      request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

      This attribute indicates the minimum duration between two consecutive Col-
      located Interference Response frames from the reporting STA."
   ::= { dot11WNMRequestEntry 53 }

dot11WNMRqstVendorSpecific OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(0..255))
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity when making a management
      request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

      This attribute provides an envelope for any optional vendor specific sub-
      elements that may be included in a WNM request element. The default value
      is null."
   DEFVAL { ''H }
   ::= { dot11WNMRequestEntry 54}

dot11WNMRqstDestinationURI OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(0..253))
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
```

```
      "This is a control variable.
      It is written by an external management entity when making a management
      request. Changes take effect when dot11WNMRqstRowStatus is set to Active.

      This attribute provides the Destination URI which defines an alternate
      destination for the WNM request. The alternate destination may be an
      internet address on an Ethernet adapter, for example, to be used when the
      wireless link to the requesting entity is unavailable or unreliable.  The
      default value is null."
   DEFVAL { ''H }
   ::= { dot11WNMRequestEntry 55}

-- **********************************************************************
-- * End of dot11WNMRequest TABLE
-- **********************************************************************

-- ********************************************************************
-- * Wireless network management reports:
-- * Report tables contain WNM reports received by this STA or
-- * results of WNM requests performed by this STA.
-- ********************************************************************

dot11WNMReport OBJECT IDENTIFIER ::= { dot11WirelessNetworkManagement 2 }

-- ********************************************************************
-- * dot11WNMVendorSpecificReport TABLE
-- ********************************************************************
dot11WNMVendorSpecificReportTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11WNMVendorSpecificReportEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "Group contains the current list of Vendor Specific reports that have been
      received by the MLME. The report tables shall be maintained as FIFO to
      preserve freshness, thus the rows in this table can be deleted for memory
      constraints or other implementation constraints determined by the vendor.
      New rows shall have different RprtIndex values than those deleted within
      the range limitation of the index. One easy way is to monotonically
      increase RprtIndex for new reports being written in the table."
   ::= { dot11WNMReport 1 }

dot11WNMVendorSpecificReportEntry OBJECT-TYPE
   SYNTAX Dot11WNMVendorSpecificReportEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "An entry in the dot11WNMVendorSpecificReportTable Indexed by
      dot11WNMVendorSpecificRprtIndex."
   INDEX { dot11WNMVendorSpecificRprtIndex }
   ::= { dot11WNMVendorSpecificReportTable 1 }

Dot11WNMVendorSpecificReportEntry ::=
   SEQUENCE {
      dot11WNMVendorSpecificRprtIndex                       Unsigned32,
      dot11WNMVendorSpecificRprtRqstToken                   OCTET STRING,
      dot11WNMVendorSpecificRprtIfIndex                     InterfaceIndex,
      dot11WNMVendorSpecificRprtContent                     OCTET STRING }

dot11WNMVendorSpecificRprtIndex OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "Index for Vendor Specific Report elements in
      dot11WNMVendorSpecificReportTable, greater than 0."
   ::= { dot11WNMVendorSpecificReportEntry 1 }

dot11WNMVendorSpecificRprtRqstToken OBJECT-TYPE
   SYNTAX OCTET STRING
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a management report is completed.

      This attribute indicates the request token that was indicated in the WNM
      request that generated this measurement report. This should be an exact
      match to the original dot11WNMRqstToken attribute. Note that there may be
      multiple entries in the table that match this value since a single request
      may generate multiple WNM reports."
   ::= { dot11WNMVendorSpecificReportEntry 2 }

dot11WNMVendorSpecificRprtIfIndex OBJECT-TYPE
   SYNTAX InterfaceIndex
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "The ifIndex for this row of WNMVendorSpecific Report has been received
      on."
   ::= { dot11WNMVendorSpecificReportEntry 3 }

dot11WNMVendorSpecificRprtContent OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(0..255))
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a management report is completed.

      This attribute provides an envelope for all the vendor specific subele-
      ments that may be included in a WNM Vendor Specific request element. The
      default value is null."
   DEFVAL { ''H }
   ::= { dot11WNMVendorSpecificReportEntry 4 }

-- **********************************************************************
-- * End of dot11WNMVendorSpecificReport TABLE
-- **********************************************************************

-- **********************************************************************
-- * dot11WNMMulticastDiagnosticReport TABLE
-- **********************************************************************
dot11WNMMulticastDiagnosticReportTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11WNMMulticastDiagnosticReportEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "Group contains the current list of Multicast Diagnostic reports that have
      been received by the MLME. The report tables shall be maintained as FIFO
      to preserve freshness, thus the rows in this table can be deleted for mem-
      ory constraints or other implementation constraints determined by the ven-
      dor. New rows shall have different RprtIndex values than those deleted
      within the range limitation of the index. One easy way is to monotonically
      increase RprtIndex for new reports being written in the table."
   ::= { dot11WNMReport 2 }

dot11WNMMulticastDiagnosticReportEntry OBJECT-TYPE
   SYNTAX Dot11WNMMulticastDiagnosticReportEntry
```

```
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry in the dot11WNMMulticastDiagnosticReportTable Indexed by
        dot11WNMMulticastDiagnosticRprtIndex."
    INDEX { dot11WNMMulticastDiagnosticRprtIndex }
    ::= { dot11WNMMulticastDiagnosticReportTable 1 }

Dot11WNMMulticastDiagnosticReportEntry ::=
    SEQUENCE {
        dot11WNMMulticastDiagnosticRprtIndex                  Unsigned32,
        dot11WNMMulticastDiagnosticRprtRqstToken              OCTET STRING,
        dot11WNMMulticastDiagnosticRprtIfIndex                InterfaceIndex,
        dot11WNMMulticastDiagnosticRprtMeasurementTime        TSFType,
        dot11WNMMulticastDiagnosticRprtDuration               Unsigned32,
        dot11WNMMulticastDiagnosticRprtMcstGroup              MacAddress,
        dot11WNMMulticastDiagnosticRprtReason                 OCTET STRING,
        dot11WNMMulticastDiagnosticRprtRcvdMsduCount          Unsigned32,
        dot11WNMMulticastDiagnosticRprtFirstSeqNumber         Unsigned32,
        dot11WNMMulticastDiagnosticRprtLastSeqNumber          Unsigned32,
        dot11WNMMulticastDiagnosticRprtMcstRate               Unsigned32  }

dot11WNMMulticastDiagnosticRprtIndex OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Index for Multicast Diagnostic Report elements in
        dot11WNMMulticastDiagnosticReportTable, greater than 0."
    ::= { dot11WNMMulticastDiagnosticReportEntry 1 }

dot11WNMMulticastDiagnosticRprtRqstToken OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the request token that was indicated in the WNM
        request that generated this measurement report. This should be an exact
        match to the original dot11WNMRqstToken attribute. Note that there may be
        multiple entries in the table that match this value since a single request
        may generate multiple WNM reports."
    ::= { dot11WNMMulticastDiagnosticReportEntry 2 }

dot11WNMMulticastDiagnosticRprtIfIndex OBJECT-TYPE
    SYNTAX InterfaceIndex
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The ifIndex for this row of WNMMulticastDiagnostic Report has been
        received on."
    ::= { dot11WNMMulticastDiagnosticReportEntry 3 }

dot11WNMMulticastDiagnosticRprtMeasurementTime OBJECT-TYPE
    SYNTAX TSFType
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the value of the STA TSF timer at the time the
        measurement started. For a triggered Multicast Diagnostics report, this is
        the TSF value at the reporting STA when the trigger condition was met.
        When the reason for sending the report is Performance Measurement and the
        Multicast Received MSDU Count is nonzero, the Measurement Time field is
        set to the value of the STA TSF timer at the time of the first multicast
        MSDU received during the measurement interval."
    ::= { dot11WNMMulticastDiagnosticReportEntry 4 }

dot11WNMMulticastDiagnosticRprtDuration OBJECT-TYPE
    SYNTAX Unsigned32
    UNITS "TUs"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the period over which the Multicast Diagnostic
        Report was generated, expressed in units of TUs."
    ::= { dot11WNMMulticastDiagnosticReportEntry 5 }

dot11WNMMulticastDiagnosticRprtMcstGroup OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        Multicast Group address indicates the MAC address of the multicast group
        for this report element."
    ::= { dot11WNMMulticastDiagnosticReportEntry 6 }

dot11WNMMulticastDiagnosticRprtReason OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(1))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the reason why the measuring STA sent the Multi-
        cast Diagnostics report. b0 (least significant bit) indicates Inactivity
        Timeout Trigger. b1 indicates the measurement result from the completed
        measurement. These are defined further in 8.4.2.24.12."
    ::= { dot11WNMMulticastDiagnosticReportEntry 7 }

dot11WNMMulticastDiagnosticRprtRcvdMsduCount OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the total number of multicast MSDUs with the
        indicated Multicast MAC Address that were received during the Measurement
        Duration. For a triggered multicast diagnostics measurement this is the
        total number of MSDUs received between the acceptance of the multicast
        diagnostics measurement request and the occurrence of the trigger condi-
        tion for MSDUs with the indicated Multicast MAC Address."
    ::= { dot11WNMMulticastDiagnosticReportEntry 8 }

dot11WNMMulticastDiagnosticRprtFirstSeqNumber OBJECT-TYPE
```

```
    SYNTAX Unsigned32 (0..65535)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the twelve least significant bits of the First
        Sequence Number field. When the LSB of the first octet of the Multicast
        MAC address field in the multicast diagnostic request is set to 1, the
        twelve LSBs of the First Sequence Number field contain the sequence number
        of the first frame received with destination address equal to the value in
        the Multicast MAC address field during the measurement period. When the
        LSB of the first octet of the Multicast MAC address field in the multicast
        diagnostic request is set to 0, the twelve LSBs of the First Sequence Num-
        ber field contain the sequence number of the first group addressed frame,
        that does not have the broadcast MAC address as its destination, received
        during the measurement period. The four most significant bits of the First
        Sequence Number field are set to 0. This field is set to 0 if the Multi-
        cast Received MSDU Count is 0."
    ::= { dot11WNMMulticastDiagnosticReportEntry 9 }

dot11WNMMulticastDiagnosticRprtLastSeqNumber OBJECT-TYPE
    SYNTAX Unsigned32 (0..65535)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the twelve least significant bits of the Last
        Sequence Number field. When the LSB of the first octet of the Multicast
        MAC address field in the multicast diagnostic request is set to 1, the
        twelve LSBs of the Last Sequence Number field contain the sequence number
        of the last frame received with destination address equal to the value in
        the Multicast MAC address field during the measurement period. When the
        LSB of the first octet of the Multicast MAC address field in the multicast
        diagnostic request is 0, the twelve LSBs of the Last Sequence Number field
        contain the sequence number of the last group addressed frame, that does
        not have the broadcast MAC address as its destination, received during the
        measurement period. The four most significant bits of the Last Sequence
        Number field are set to 0. This field is set to 0 if the Multicast
        Received MSDU Count is 0."
    ::= { dot11WNMMulticastDiagnosticReportEntry 10 }

dot11WNMMulticastDiagnosticRprtMcstRate OBJECT-TYPE
    SYNTAX Unsigned32 (0..65535)
    UNITS "0.5Mb/s"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the highest data rate, in 0.5 Mb/s units, at
        which the STA has received a group addressed frame with a valid FCS during
        the measurement period.The Multicast Rate field is encoded with the MSB
        set to 1 to indicate that the data rate is in the basic rate set, and set
        to 0 to indicate that the data rate is not in the basic rate set. The
        remaining 15 bit value is multiplied by 0.5 Mb/s to indicate the data
        rate. The Multicast Rate field is set to 0 by the STA to indicate that it
        has not received a group addressed frame with a valid FCS during the mea-
        surement period."
    ::= { dot11WNMMulticastDiagnosticReportEntry 11 }
```

```
-- **********************************************************************
-- * End of dot11WNMMulticastDiagnosticReport TABLE
-- **********************************************************************

-- **********************************************************************
-- * dot11WNMLocationCivicReport TABLE
-- **********************************************************************
dot11WNMLocationCivicReportTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11WNMLocationCivicReportEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "Group contains the current list of Location Civic reports that have been
      received by the MLME. The report tables shall be maintained as FIFO to
      preserve freshness, thus the rows in this table can be deleted for memory
      constraints or other implementation constraints determined by the vendor.
      New rows shall have different RprtIndex values than those deleted within
      the range limitation of the index. One easy way is to monotonically
      increase RprtIndex for new reports being written in the table."
   ::= { dot11WNMReport 3 }

dot11WNMLocationCivicReportEntry OBJECT-TYPE
   SYNTAX Dot11WNMLocationCivicReportEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "An entry in the dot11WNMLocationCivicReportTable Indexed by
      dot11WNMLocationCivicRprtIndex."
   INDEX { dot11WNMLocationCivicRprtIndex }
   ::= { dot11WNMLocationCivicReportTable 1 }

Dot11WNMLocationCivicReportEntry ::=
   SEQUENCE {
      dot11WNMLocationCivicRprtIndex                        Unsigned32,
      dot11WNMLocationCivicRprtRqstToken                    OCTET STRING,
      dot11WNMLocationCivicRprtIfIndex                      InterfaceIndex,
      dot11WNMLocationCivicRprtCivicLocation                OCTET STRING }

dot11WNMLocationCivicRprtIndex OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "Index for Location Civic Report elements in
      dot11WNMLocationCivicReportTable, greater than 0."
   ::= { dot11WNMLocationCivicReportEntry 1 }

dot11WNMLocationCivicRprtRqstToken OBJECT-TYPE
   SYNTAX OCTET STRING
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a management report is completed.

      This attribute indicates the request token that was indicated in the WNM
      request that generated this measurement report. This should be an exact
      match to the original dot11WNMRqstToken attribute. Note that there may be
      multiple entries in the table that match this value since a single request
      may generate multiple WNM reports."
   ::= { dot11WNMLocationCivicReportEntry 2 }

dot11WNMLocationCivicRprtIfIndex OBJECT-TYPE
```

```
    SYNTAX InterfaceIndex
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The ifIndex for this row of WNMLocationCivic Report has been received
        on."
    ::= { dot11WNMLocationCivicReportEntry 3 }

dot11WNMLocationCivicRprtCivicLocation OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates a variable octet field and contains a list of
        civic address elements in TLV format as defined in IETF RFC 4776-2006."
    ::= { dot11WNMLocationCivicReportEntry 4}

-- ********************************************************************
-- * End of dot11WNMLocationCivicReport TABLE
-- ********************************************************************

-- ********************************************************************
-- * dot11WNMLocationIdentifierReport TABLE
-- ********************************************************************
dot11WNMLocationIdentifierReportTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11WNMLocationIdentifierReportEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Group contains the current list of Location Identifier reports that have
        been received by the MLME. The report tables shall be maintained as FIFO
        to preserve freshness, thus the rows in this table can be deleted for mem-
        ory constraints or other implementation constraints determined by the ven-
        dor. New rows shall have different RprtIndex values than those deleted
        within the range limitation of the index. One easy way is to monotonically
        increase RprtIndex for new reports being written in the table."
    ::= { dot11WNMReport 4 }

dot11WNMLocationIdentifierReportEntry OBJECT-TYPE
    SYNTAX Dot11WNMLocationIdentifierReportEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry in the dot11WNMLocationIdentifierReportTable Indexed by
        dot11WNMLocationIdentifierRprtIndex."
    INDEX { dot11WNMLocationIdentifierRprtIndex }
    ::= { dot11WNMLocationIdentifierReportTable 1 }

Dot11WNMLocationIdentifierReportEntry ::=
    SEQUENCE {
        dot11WNMLocationIdentifierRprtIndex                Unsigned32,
        dot11WNMLocationIdentifierRprtRqstToken            OCTET STRING,
        dot11WNMLocationIdentifierRprtIfIndex              InterfaceIndex,
        dot11WNMLocationIdentifierRprtExpirationTSF        TSFType,
        dot11WNMLocationIdentifierRprtPublicIdUri          OCTET STRING }

dot11WNMLocationIdentifierRprtIndex OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
```

```
        "Index for Location Identifier Report elements in
        dot11WNMLocationIdentifierReportTable, greater than 0."
    ::= { dot11WNMLocationIdentifierReportEntry 1 }

dot11WNMLocationIdentifierRprtRqstToken OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the request token that was indicated in the WNM
        request that generated this measurement report. This should be an exact
        match to the original dot11WNMRqstToken attribute. Note that there may be
        multiple entries in the table that match this value since a single request
        may generate multiple WNM reports."
    ::= { dot11WNMLocationIdentifierReportEntry 2 }

dot11WNMLocationIdentifierRprtIfIndex OBJECT-TYPE
    SYNTAX InterfaceIndex
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The ifIndex for this row of WNMLocationIdentifier Report has been
        received on."
    ::= { dot11WNMLocationIdentifierReportEntry 3 }

dot11WNMLocationIdentifierRprtExpirationTSF OBJECT-TYPE
    SYNTAX TSFType
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the value of the STA TSF timer when the Public
        Identifier URI field value is no longer valid. The Expiration TSF field
        set to 0 indicates the Public Identifier URI does not expire."
    ::= { dot11WNMLocationIdentifierReportEntry 4 }

dot11WNMLocationIdentifierRprtPublicIdUri OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates a value in URI format that points to a location
        object. It can be used to return the location value for the requesting
        STA. The format of the location value returned when the URI is derefer-
        enced is dependent on the provider of the URI and is beyond the scope of
        this document. The Public Identifier URI confirms the validity of the
        location estimate to an external agent when a STA forwards a location
        estimate to that agent. The protocol used to query the infrastructure for
        a location report based on the Public Identifier URI is beyond the scope
        of this standard."
    ::= { dot11WNMLocationIdentifierReportEntry 5}

-- **********************************************************************
-- * End of dot11WNMLocationIdentifierReport TABLE
-- **********************************************************************
-- *******************************************************************
-- * dot11WNMEventTransitReport TABLE
-- *******************************************************************
dot11WNMEventTransitReportTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11WNMEventTransitReportEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "Group contains the current list of Transition Event reports that have
      been received by the MLME. The report tables shall be maintained as FIFO
      to preserve freshness, thus the rows in this table can be deleted for mem-
      ory constraints or other implementation constraints determined by the ven-
      dor. New rows shall have different RprtIndex values than those deleted
      within the range limitation of the index. One easy way is to monotonically
      increase RprtIndex for new reports being written in the table."
   ::= { dot11WNMReport 5 }

dot11WNMEventTransitReportEntry OBJECT-TYPE
   SYNTAX Dot11WNMEventTransitReportEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "An entry in the dot11WNMEventTransitReportTable Indexed by
      dot11WNMEventTransitRprtIndex."
   INDEX { dot11WNMEventTransitRprtIndex }
   ::= { dot11WNMEventTransitReportTable 1 }

Dot11WNMEventTransitReportEntry ::=
   SEQUENCE {
      dot11WNMEventTransitRprtIndex                         Unsigned32,
      dot11WNMEventTransitRprtRqstToken                     OCTET STRING,
      dot11WNMEventTransitRprtIfIndex                       InterfaceIndex,
      dot11WNMEventTransitRprtEventStatus                   INTEGER,
      dot11WNMEventTransitRprtEventTSF                      TSFType,
      dot11WNMEventTransitRprtUTCOffset                     OCTET STRING,
      dot11WNMEventTransitRprtTimeError                     OCTET STRING,
      dot11WNMEventTransitRprtSourceBssid                   MacAddress,
      dot11WNMEventTransitRprtTargetBssid                   MacAddress,
      dot11WNMEventTransitRprtTransitTime                   Unsigned32,
      dot11WNMEventTransitRprtTransitReason                 INTEGER,
      dot11WNMEventTransitRprtTransitResult                 Unsigned32,
      dot11WNMEventTransitRprtSourceRCPI                    Unsigned32,
      dot11WNMEventTransitRprtSourceRSNI                    Unsigned32,
      dot11WNMEventTransitRprtTargetRCPI                    Unsigned32,
      dot11WNMEventTransitRprtTargetRSNI                    Unsigned32 }

dot11WNMEventTransitRprtIndex OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "Index for Transition Event Report elements in
      dot11WNMEventTransitReportTable, greater than 0."
   ::= { dot11WNMEventTransitReportEntry 1 }

dot11WNMEventTransitRprtRqstToken OBJECT-TYPE
   SYNTAX OCTET STRING
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a management report is completed.

      This attribute indicates the request token that was indicated in the WNM
```

```
        request that generated this measurement report. This should be an exact
        match to the original dot11WNMRqstToken attribute. Note that there may be
        multiple entries in the table that match this value since a single request
        may generate multiple WNM reports."
    ::= { dot11WNMEventTransitReportEntry 2 }

dot11WNMEventTransitRprtIfIndex OBJECT-TYPE
    SYNTAX InterfaceIndex
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The ifIndex for this row of WNMEventTransit Report has been received on."
    ::= { dot11WNMEventTransitReportEntry 3 }

dot11WNMEventTransitRprtEventStatus OBJECT-TYPE
    SYNTAX INTEGER {
        successful(0),
        requestFailed(1),
        requestRefused(2),
        requestIncapable(3),
        detectedFrequentTransition(4)
        }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute contains the status value included in the Event Report."
    ::= { dot11WNMEventTransitReportEntry 4 }

dot11WNMEventTransitRprtEventTSF OBJECT-TYPE
    SYNTAX TSFType
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute contains the value of the Event timestamp field."
    ::= { dot11WNMEventTransitReportEntry 5 }

dot11WNMEventTransitRprtUTCOffset OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(10))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the UTC Offset Time Value optionally included in
        the Event Report."
    ::= { dot11WNMEventTransitReportEntry 6 }

dot11WNMEventTransitRprtTimeError OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(5))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute contains the value of the Event Time Error field optionally
        included in the Event Report."
    ::= { dot11WNMEventTransitReportEntry 7 }

dot11WNMEventTransitRprtSourceBssid OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the source BSSID for the reported transition
        event."
    ::= { dot11WNMEventTransitReportEntry 8 }

dot11WNMEventTransitRprtTargetBssid OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the target BSSID for the reported transition
        event."
    ::= { dot11WNMEventTransitReportEntry 9 }

dot11WNMEventTransitRprtTransitTime OBJECT-TYPE
    SYNTAX Unsigned32 (0..65535)
    UNITS "TUs"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the transition time for the reported transition
        event in TUs. The Transition time is defined as the time difference
        between the starting time and the ending time of a transition between APs,
        even if the transition results in remaining on the same AP. Start and end
        times for a transition event are defined in 10.23.2.2"
    ::= { dot11WNMEventTransitReportEntry 10 }

dot11WNMEventTransitRprtTransitReason OBJECT-TYPE
    SYNTAX INTEGER {
        unspecified(0),
        excessiveFrameLossRatesPoorConditions(1),
        excessiveDelayForCurrentTrafficStreams(2),
        insufficientQosCapacityForCurrentTrafficStreams(3),
        firstAssociationToEss(4),
        loadBalancing(5),
        betterApFound(6),
        deauthenticatedDisassociatedFromPreviousAp(7),
        apFailedIeee8021XEapAuthentication(8),
        apFailed4wayHandshake(9),
        receivedTooManyReplayCounterFailures(10),
        receivedTooManyDataMICFailures(11),
        exceededMaxNumberOfRetransmissions(12),
        receivedTooManyBroadcastDisassociations(13),
        receivedTooManyBroadcastDeauthentications(14),
        previousTransitionFailed(15),
        lowRSSI(16)
        }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the reason for the reported BSS Transition event.
        The format for this list of reasons is further detailed in 8.4.2.70.2."
    ::= { dot11WNMEventTransitReportEntry 11 }

dot11WNMEventTransitRprtTransitResult OBJECT-TYPE
    SYNTAX Unsigned32(0..65535)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the result of the attempted transition and is set
        to one of the status codes specified in Table 8-37 in 8.4.1.9."
    ::= { dot11WNMEventTransitReportEntry 12 }

dot11WNMEventTransitRprtSourceRCPI OBJECT-TYPE
    SYNTAX Unsigned32(0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the received channel power of the most recently
        measured frame from the Source BSSID before the STA reassociates to the
        Target BSSID. The Source RCPI is a logarithmic function of the received
        signal power, as defined in the RCPI measurement subclause for the PHY
        Type."
    ::= { dot11WNMEventTransitReportEntry 13 }

dot11WNMEventTransitRprtSourceRSNI OBJECT-TYPE
    SYNTAX Unsigned32(0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the received signal-to-noise indication of the
        most recently measured frame from the Source BSSID before the STA reasso-
        ciates to the Target BSSID. The Source RSNI is a logarithmic function of
        the signal-to-noise ratio, as defined in 8.4.2.43."
    ::= { dot11WNMEventTransitReportEntry 14 }

dot11WNMEventTransitRprtTargetRCPI OBJECT-TYPE
    SYNTAX Unsigned32(0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the received channel power of the first measured
        frame just after STA reassociates to the Target BSSID. If association with
        target BSSID failed, the Target RCPI field indicates the received channel
        power of the most recently measured frame from the Target BSSID. The Tar-
        get RCPI is a logarithmic function of the received signal power, as
        defined in the RCPI measurement subclause for the PHY Type."
    ::= { dot11WNMEventTransitReportEntry 15 }
```

```
dot11WNMEventTransitRprtTargetRSNI OBJECT-TYPE
    SYNTAX Unsigned32(0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the received signal-to-noise indication of the
        first measured frame just after STA reassociates to the Target BSSID. If
        association with target BSSID failed, the Target RCPI field indicates the
        received signal-to-noise indication of the most recently measured frame
        from the Target BSSID. The Target RSNI is a logarithmic function of the
        signal-to-noise ratio, as defined in 8.4.2.43."
    ::= { dot11WNMEventTransitReportEntry 16 }

-- **********************************************************************
-- * End of dot11WNMEventTransitReport TABLE
-- **********************************************************************

-- **********************************************************************
-- * dot11WNMEventRsnaReport TABLE
-- **********************************************************************
dot11WNMEventRsnaReportTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11WNMEventRsnaReportEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Group contains the current list of RSNA Event reports that have been
        received by the MLME. The report tables shall be maintained as FIFO to
        preserve freshness, thus the rows in this table can be deleted for memory
        constraints or other implementation constraints determined by the vendor.
        New rows shall have different RprtIndex values than those deleted within
        the range limitation of the index. One easy way is to monotonically
        increase RprtIndex for new reports being written in the table."
    ::= { dot11WNMReport 6 }

dot11WNMEventRsnaReportEntry OBJECT-TYPE
    SYNTAX Dot11WNMEventRsnaReportEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry in the dot11WNMEventRsnaReportTable Indexed by
        dot11WNMEventRsnaRprtIndex."
    INDEX { dot11WNMEventRsnaRprtIndex }
    ::= { dot11WNMEventRsnaReportTable 1 }

Dot11WNMEventRsnaReportEntry ::=
    SEQUENCE {
        dot11WNMEventRsnaRprtIndex                          Unsigned32,
        dot11WNMEventRsnaRprtRqstToken                      OCTET STRING,
        dot11WNMEventRsnaRprtIfIndex                        InterfaceIndex,
        dot11WNMEventRsnaRprtEventStatus                    INTEGER,
        dot11WNMEventRsnaRprtEventTSF                       TSFType,
        dot11WNMEventRsnaRprtUTCOffset                      OCTET STRING,
        dot11WNMEventRsnaRprtTimeError                      OCTET STRING,
        dot11WNMEventRsnaRprtTargetBssid                    MacAddress,
        dot11WNMEventRsnaRprtAuthType                       OCTET STRING,
        dot11WNMEventRsnaRprtEapMethod                      OCTET STRING,
        dot11WNMEventRsnaRprtResult                         Unsigned32,
        dot11WNMEventRsnaRprtRsnElement                     OCTET STRING }

dot11WNMEventRsnaRprtIndex OBJECT-TYPE
```

```
    SYNTAX Unsigned32
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Index for RSNA Event Report elements in dot11WNMEventRsnaReportTable,
        greater than 0."
    ::= { dot11WNMEventRsnaReportEntry 1 }

dot11WNMEventRsnaRprtRqstToken OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the request token that was indicated in the WNM
        request that generated this measurement report. This should be an exact
        match to the original dot11WNMRqstToken attribute. Note that there may be
        multiple entries in the table that match this value since a single request
        may generate multiple WNM reports."
    ::= { dot11WNMEventRsnaReportEntry 2 }

dot11WNMEventRsnaRprtIfIndex OBJECT-TYPE
    SYNTAX InterfaceIndex
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The ifIndex for this row of WNMEventRsna Report has been received on."
    ::= { dot11WNMEventRsnaReportEntry 3 }

dot11WNMEventRsnaRprtEventStatus OBJECT-TYPE
    SYNTAX INTEGER {
        successful(0),
        requestFailed(1),
        requestRefused(2),
        requestIncapable(3),
        detectedFrequentTransition(4)
        }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute contains the status value included in the Event Report."
    ::= { dot11WNMEventRsnaReportEntry 4 }

dot11WNMEventRsnaRprtEventTSF OBJECT-TYPE
    SYNTAX TSFType
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute contains the value of the Event timestamp field."
    ::= { dot11WNMEventRsnaReportEntry 5 }

dot11WNMEventRsnaRprtUTCOffset OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(10))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
```

```
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the UTC Offset Time Value optionally included in
        the Event Report."
    ::= { dot11WNMEventRsnaReportEntry 6 }

dot11WNMEventRsnaRprtTimeError OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(5))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute contains the value of the Event Time Error field optionally
        included in the Event Report."
    ::= { dot11WNMEventRsnaReportEntry 7 }

dot11WNMEventRsnaRprtTargetBssid OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the BSSID of the AP accepting the authorization
        attempt."
    ::= { dot11WNMEventRsnaReportEntry 8 }

dot11WNMEventRsnaRprtAuthType OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(4))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the AKM suite, as defined in Table 8-101 in
        8.4.2.27.3. The first three octets indicate the OUI. The last octet indi-
        cates the suite type."
    ::= { dot11WNMEventRsnaReportEntry 9 }

dot11WNMEventRsnaRprtEapMethod OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(1..8))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates a value that identifies the EAP Method. When the
        Authentication Type field is set to the value of either 00-0F-AC:1
        (Authentication negotiated over IEEE 802.1X or using PMKSA caching as
        defined in 11.5.9.3) or 00-0F-AC:3 (AKM suite selector for Fast BSS Tran-
        sition as defined in 11.6.1.7), the EAP Method field contains the IANA
        assigned EAP type defined at http://www.iana.org/assignments/eap-numbers.
        The EAP type contains either the legacy type (1 octet) or the expanded
        type (1 octet type = 254, 3-octet Vendor ID, 4-octet Vendor-Type). The EAP
        Method field is set to 0 otherwise."
    ::= { dot11WNMEventRsnaReportEntry 10 }

dot11WNMEventRsnaRprtResult OBJECT-TYPE
```

```
    SYNTAX Unsigned32(0..65535)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the result of the RSNA event and is set to one of
        the status codes specified in Table 8-37 in 8.4.1.9."
    ::= { dot11WNMEventRsnaReportEntry 11 }

dot11WNMEventRsnaRprtRsnElement OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute contains the entire contents of the negotiated RSNE at the
        time of the authentication attempt. The maximum length of the RSNE field
        is less than the maximum length of an RSNE, as defined in 8.4.2.27. If the
        length of the RSNE included here exceeds the maximum length of the RSNE
        field, the RSNE shall be truncated to the maximum length allowed for the
        RSNE field."
    ::= { dot11WNMEventRsnaReportEntry 12 }

-- **********************************************************************
-- * End of dot11WNMEventRsnaReport TABLE
-- **********************************************************************

-- **********************************************************************
-- * dot11WNMEventPeerReport TABLE
-- **********************************************************************
dot11WNMEventPeerReportTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11WNMEventPeerReportEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Group contains the current list of Peer-to-Peer Event reports that have
        been received by the MLME. The report tables shall be maintained as FIFO
        to preserve freshness, thus the rows in this table can be deleted for mem-
        ory constraints or other implementation constraints determined by the ven-
        dor. New rows shall have different RprtIndex values than those deleted
        within the range limitation of the index. One easy way is to monotonically
        increase RprtIndex for new reports being written in the table."
    ::= { dot11WNMReport 7 }

dot11WNMEventPeerReportEntry OBJECT-TYPE
    SYNTAX Dot11WNMEventPeerReportEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry in the dot11WNMEventPeerReportTable Indexed by
        dot11WNMEventPeerRprtIndex."
    INDEX { dot11WNMEventPeerRprtIndex }
    ::= { dot11WNMEventPeerReportTable 1 }

Dot11WNMEventPeerReportEntry ::=
    SEQUENCE {
        dot11WNMEventPeerRprtIndex                              Unsigned32,
        dot11WNMEventPeerRprtRqstToken                          OCTET STRING,
        dot11WNMEventPeerRprtIfIndex                            InterfaceIndex,
        dot11WNMEventPeerRprtEventStatus                        INTEGER,
        dot11WNMEventPeerRprtEventTSF                            TSFType,
        dot11WNMEventPeerRprtUTCOffset                           OCTET STRING,
        dot11WNMEventPeerRprtTimeError                           OCTET STRING,
        dot11WNMEventPeerRprtPeerMacAddress                      MacAddress,
        dot11WNMEventPeerRprtOperatingClass                      Unsigned32,
        dot11WNMEventPeerRprtChanNumber                          Unsigned32,
        dot11WNMEventPeerRprtStaTxPower                          Integer32,
        dot11WNMEventPeerRprtConnTime                            Unsigned32,
        dot11WNMEventPeerRprtPeerStatus                          INTEGER }

dot11WNMEventPeerRprtIndex OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Index for Peer-to-Peer Event Report elements in
        dot11WNMEventPeerReportTable, greater than 0."
    ::= { dot11WNMEventPeerReportEntry 1 }

dot11WNMEventPeerRprtRqstToken OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the request token that was indicated in the WNM
        request that generated this measurement report. This should be an exact
        match to the original dot11WNMRqstToken attribute. Note that there may be
        multiple entries in the table that match this value since a single request
        may generate multiple WNM reports."
    ::= { dot11WNMEventPeerReportEntry 2 }

dot11WNMEventPeerRprtIfIndex OBJECT-TYPE
    SYNTAX InterfaceIndex
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The ifIndex for this row of WNMEventPeer Report has been received on."
    ::= { dot11WNMEventPeerReportEntry 3 }

dot11WNMEventPeerRprtEventStatus OBJECT-TYPE
    SYNTAX INTEGER {
        successful(0),
        requestFailed(1),
        requestRefused(2),
        requestIncapable(3),
        detectedFrequentTransition(4)
        }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute contains the status value included in the Event Report."
    ::= { dot11WNMEventPeerReportEntry 4 }

dot11WNMEventPeerRprtEventTSF OBJECT-TYPE
    SYNTAX TSFType
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
```

```
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute contains the value of the Event timestamp field."
    ::= { dot11WNMEventPeerReportEntry 5 }

dot11WNMEventPeerRprtUTCOffset OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(10))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the UTC Offset Time Value optionally included in
        the Event Report."
    ::= { dot11WNMEventPeerReportEntry 6 }

dot11WNMEventPeerRprtTimeError OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(5))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute contains the value of the Event Time Error field optionally
        included in the Event Report."
    ::= { dot11WNMEventPeerReportEntry 7 }

dot11WNMEventPeerRprtPeerMacAddress OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the MAC address of the peer STA or IBSS BSSID. If
        this event is for a Peer-to-Peer link in an infrastructure BSS, this field
        contains the MAC address of the peer STA. If this event is for a Peer-to-
        Peer link in an IBSS, this field contains the BSSID of the IBSS."
    ::= { dot11WNMEventPeerReportEntry 8 }

dot11WNMEventPeerRprtOperatingClass OBJECT-TYPE
    SYNTAX Unsigned32(1..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the channel set for this Peer-to-Peer Event
        report. Country, Operating Class and Channel Number together specify the
        channel frequency and spacing for this measurement request. Valid values
        of Operating Class as shown in Annex E."
    ::= { dot11WNMEventPeerReportEntry 9 }

dot11WNMEventPeerRprtChanNumber OBJECT-TYPE
    SYNTAX Unsigned32 (1..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
```

```
      It is written by the SME when a management report is completed.

      This attribute indicates the current operating channel for this Peer-to-
      Peer Event report. The Channel Number is only defined within the indicated
      Operating Class as shown in Annex E."
   ::= { dot11WNMEventPeerReportEntry 10 }

dot11WNMEventPeerRprtStaTxPower OBJECT-TYPE
   SYNTAX Integer32 (-128..127)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a management report is completed.

      This attribute indicates the STA transmit power used for the Peer-to-Peer
      link. The STA Tx Power field indicates the target transmit power at the
      antenna in dBm with a tolerance of +/-5dB for the lowest basic rate of the
      reporting STA."
   ::= { dot11WNMEventPeerReportEntry 11 }

dot11WNMEventPeerRprtConnTime OBJECT-TYPE
   SYNTAX Unsigned32 (0..16777215)
   UNITS "seconds"
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a management report is completed.

      This attribute indicates a value representing the connection time for the
      reported Peer-to-Peer event. If the Peer Status is 0, this field indicates
      the duration of the Direct Link. If the Peer Status is 1, this field indi-
      cates the time difference from the time the Direct Link was established to
      the time at which the reporting STA generated the event report. If the
      Peer Status is 2, this field indicates the duration of the IBSS member-
      ship. If the Peer Status is 3, this field indicates the time difference
      from the time the STA joined the IBSS to the time at which the reporting
      STA generated the event report. See 10.23.2.4."
   ::= { dot11WNMEventPeerReportEntry 12 }

dot11WNMEventPeerRprtPeerStatus OBJECT-TYPE
   SYNTAX INTEGER {
      directLinkTerminated(0),
      directLinkActive(1),
      ibssMembershipTerminated(2),
      ibssMembershipActive(3)
      }
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a management report is completed.

      This attribute indicates the peer link connection status."
   ::= { dot11WNMEventPeerReportEntry 13 }

-- **********************************************************************
-- * End of dot11WNMEventPeerReport TABLE
-- **********************************************************************

-- **********************************************************************
-- * dot11WNMEventWNMLogReport TABLE
-- **********************************************************************
```

```
dot11WNMEventWNMLogReportTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11WNMEventWNMLogReportEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "Group contains the current list of WNMLog Event reports that have been
      received by the MLME. The report tables shall be maintained as FIFO to
      preserve freshness, thus the rows in this table can be deleted for memory
      constraints or other implementation constraints determined by the vendor.
      New rows shall have different RprtIndex values than those deleted within
      the range limitation of the index. One easy way is to monotonically
      increase RprtIndex for new reports being written in the table."
   ::= { dot11WNMReport 8 }

dot11WNMEventWNMLogReportEntry OBJECT-TYPE
   SYNTAX Dot11WNMEventWNMLogReportEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "An entry in the dot11WNMEventWNMLogReportTable Indexed by
      dot11WNMEventWNMLogRprtIndex."
   INDEX { dot11WNMEventWNMLogRprtIndex }
   ::= { dot11WNMEventWNMLogReportTable 1 }

Dot11WNMEventWNMLogReportEntry ::=
   SEQUENCE {
      dot11WNMEventWNMLogRprtIndex                     Unsigned32,
      dot11WNMEventWNMLogRprtRqstToken                 OCTET STRING,
      dot11WNMEventWNMLogRprtIfIndex                   InterfaceIndex,
      dot11WNMEventWNMLogRprtEventStatus               INTEGER,
      dot11WNMEventWNMLogRprtEventTSF                  TSFType,
      dot11WNMEventWNMLogRprtUTCOffset                 OCTET STRING,
      dot11WNMEventWNMLogRprtTimeError                 OCTET STRING,
      dot11WNMEventWNMLogRprtContent                   OCTET STRING }

dot11WNMEventWNMLogRprtIndex OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "Index for WNMLog Event Report elements in dot11WNMEventWNMLogReportTable,
      greater than 0."
   ::= { dot11WNMEventWNMLogReportEntry 1 }

dot11WNMEventWNMLogRprtRqstToken OBJECT-TYPE
   SYNTAX OCTET STRING
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a management report is completed.

      This attribute indicates the request token that was indicated in the WNM
      request that generated this measurement report. This should be an exact
      match to the original dot11WNMRqstToken attribute. Note that there may be
      multiple entries in the table that match this value since a single request
      may generate multiple WNM reports."
   ::= { dot11WNMEventWNMLogReportEntry 2 }

dot11WNMEventWNMLogRprtIfIndex OBJECT-TYPE
   SYNTAX InterfaceIndex
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
```

```
      "The ifIndex for this row of WNMEventWNMLog Report has been received on."
   ::= { dot11WNMEventWNMLogReportEntry 3 }

dot11WNMEventWNMLogRprtEventStatus OBJECT-TYPE
   SYNTAX INTEGER {
      successful(0),
      requestFailed(1),
      requestRefused(2),
      requestIncapable(3),
      detectedFrequentTransition(4)
      }
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a management report is completed.

      This attribute contains the status value included in the Event Report."
   ::= { dot11WNMEventWNMLogReportEntry 4 }

dot11WNMEventWNMLogRprtEventTSF OBJECT-TYPE
   SYNTAX TSFType
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a management report is completed.

      This attribute contains the value of the Event timestamp field."
   ::= { dot11WNMEventWNMLogReportEntry 5 }

dot11WNMEventWNMLogRprtUTCOffset OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(10))
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a management report is completed.

      This attribute indicates the UTC Offset Time Value optionally included in
      the Event Report."
   ::= { dot11WNMEventWNMLogReportEntry 6 }

dot11WNMEventWNMLogRprtTimeError OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(5))
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a management report is completed.

      This attribute contains the value of the Event Time Error field optionally
      included in the Event Report."
   ::= { dot11WNMEventWNMLogReportEntry 7 }

dot11WNMEventWNMLogRprtContent OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(0..2284))
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a management report is completed.

      This attribute contains the entire syslog message, consisting of the PRI,
      HEADER, and MSG portion of a WNM Log message as described in IETF RFC
      3164-2001. The TAG field of the MSG portion of the message is a 17 octet
      string containing the ASCII representation of the STA MAC address using
      hexadecimal notation with colons between octets. The octet containing the
      individual/group bit occurs last, and that bit is in the least significant
      position within that octet. See 10.23.2.5."
   ::= { dot11WNMEventWNMLogReportEntry 8 }

-- ********************************************************************
-- * End of dot11WNMEventWNMLogReport TABLE
-- ********************************************************************

-- ********************************************************************
-- * dot11WNMDiagMfrInfoReport TABLE
-- ********************************************************************
dot11WNMDiagMfrInfoReportTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11WNMDiagMfrInfoReportEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "Group contains the current list of Manufacturer Information STA reports
      that have been received by the MLME. The report tables shall be maintained
      as FIFO to preserve freshness, thus the rows in this table can be deleted
      for memory constraints or other implementation constraints determined by
      the vendor. New rows shall have different RprtIndex values than those
      deleted within the range limitation of the index. One easy way is to mono-
      tonically increase RprtIndex for new reports being written in the table."
   ::= { dot11WNMReport 9 }

dot11WNMDiagMfrInfoReportEntry OBJECT-TYPE
   SYNTAX Dot11WNMDiagMfrInfoReportEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "An entry in the dot11WNMDiagMfrInfoReportTable Indexed by
      dot11WNMDiagMfrInfoRprtIndex."
   INDEX { dot11WNMDiagMfrInfoRprtIndex }
   ::= { dot11WNMDiagMfrInfoReportTable 1 }

Dot11WNMDiagMfrInfoReportEntry ::=
   SEQUENCE {
      dot11WNMDiagMfrInfoRprtIndex                          Unsigned32,
      dot11WNMDiagMfrInfoRprtRqstToken                      OCTET STRING,
      dot11WNMDiagMfrInfoRprtIfIndex                        InterfaceIndex,
      dot11WNMDiagMfrInfoRprtEventStatus                    INTEGER,
      dot11WNMDiagMfrInfoRprtMfrOi                          OCTET STRING,
      dot11WNMDiagMfrInfoRprtMfrIdString                    OCTET STRING,
      dot11WNMDiagMfrInfoRprtMfrModelString                 OCTET STRING,
      dot11WNMDiagMfrInfoRprtMfrSerialNumberString          OCTET STRING,
      dot11WNMDiagMfrInfoRprtMfrFirmwareVersion             OCTET STRING,
      dot11WNMDiagMfrInfoRprtMfrAntennaType                 OCTET STRING,
      dot11WNMDiagMfrInfoRprtCollocRadioType                INTEGER,
      dot11WNMDiagMfrInfoRprtDeviceType                     INTEGER,
      dot11WNMDiagMfrInfoRprtCertificateID                  OCTET STRING}

dot11WNMDiagMfrInfoRprtIndex OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "Index for Manufacturer Information STA Report elements in
      dot11WNMDiagMfrInfoReportTable, greater than 0."
   ::= { dot11WNMDiagMfrInfoReportEntry 1 }
```

```
dot11WNMDiagMfrInfoRprtRqstToken OBJECT-TYPE
   SYNTAX OCTET STRING
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a management report is completed.

      This attribute indicates the request token that was indicated in the WNM
      request that generated this measurement report. This should be an exact
      match to the original dot11WNMRqstToken attribute. Note that there may be
      multiple entries in the table that match this value since a single request
      may generate multiple WNM reports."
   ::= { dot11WNMDiagMfrInfoReportEntry 2 }

dot11WNMDiagMfrInfoRprtIfIndex OBJECT-TYPE
   SYNTAX InterfaceIndex
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "The ifIndex for this row of WNMDiagMfrInfo Report has been received on."
   ::= { dot11WNMDiagMfrInfoReportEntry 3 }

dot11WNMDiagMfrInfoRprtEventStatus OBJECT-TYPE
   SYNTAX INTEGER {
      successful(0),
      requestFailed(1),
      requestRefused(2),
      requestIncapable(3),
      detectedFrequentTransition(4)
      }
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a management report is completed.

      This attribute contains the status value included in the Event Report."
   ::= { dot11WNMDiagMfrInfoReportEntry 4}

dot11WNMDiagMfrInfoRprtMfrOi OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(0..5))
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a management report is completed.

      This attribute indicates the Manufacturer OI for the reported Manufacturer
      Information STA Diagnostic. The OUI attribute contains an organizationally
      unique identifier, the first 24-bits of the network connected device that
      indicate the specific vendor for that device."
   ::= { dot11WNMDiagMfrInfoReportEntry 5 }

dot11WNMDiagMfrInfoRprtMfrIdString OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(0..255))
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a management report is completed.

      This attribute indicates the Manufacturer ID string for the reported Man-
      ufacturer Information STA Diagnostic. The ID attribute contains an ASCII
```

```
        string indicating the manufacturer identifier of the wireless network
        adaptor. This string is not null terminated."
    ::= { dot11WNMDiagMfrInfoReportEntry 6 }

dot11WNMDiagMfrInfoRprtMfrModelString OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(0..255))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the Manufacturer model string for the reported
        Manufacturer Information STA Diagnostic. The model attribute contains an
        ASCII string indicating the model of the wireless network adaptor. This
        string is not null terminated."
    ::= { dot11WNMDiagMfrInfoReportEntry 7 }

dot11WNMDiagMfrInfoRprtMfrSerialNumberString OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(0..255))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the Manufacturer serial number string for the
        reported Manufacturer Information STA Diagnostic. The serial number attri-
        bute contains an ASCII string indicating the serial number of the wireless
        network adaptor. This string is not null terminated."
    ::= { dot11WNMDiagMfrInfoReportEntry 8 }

dot11WNMDiagMfrInfoRprtMfrFirmwareVersion OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(0..255))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the Manufacturer firmware version string for the
        reported Manufacturer Information STA Diagnostic. The firmware version
        attribute contains an ASCII string identifying the version of firmware
        currently installed on the wireless network adaptor. This string is not
        null terminated."
    ::= { dot11WNMDiagMfrInfoReportEntry 9 }

dot11WNMDiagMfrInfoRprtMfrAntennaType OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(0..255))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the Manufacturer antenna type string for the
        reported Manufacturer Information STA Diagnostic. The first octet of this
        string indicates the antenna count, and the second octet indicates the
        antenna gain. The antenna gain indicates the peak gain in dBi of the
        antenna connected to the wireless network adaptor. The remaining octets
        contain an ASCII string indicating the type of antenna connected to the
        wireless network adaptor."
    ::= { dot11WNMDiagMfrInfoReportEntry 10 }
```

```
dot11WNMDiagMfrInfoRprtCollocRadioType OBJECT-TYPE
    SYNTAX INTEGER {
        reserved(0),
        cellular(1),
        cordless(2),
        gps(3),
        ieee80211(4),
        ieee80215(5),
        ieee80216(6),
        ieee80220(7),
        ieee80222(8),
        digitalAudioBroadcasting(9),
        digitalVideoBroadcasting(10)
        }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute contains the type of the collocated radio."
    ::= { dot11WNMDiagMfrInfoReportEntry 11 }

dot11WNMDiagMfrInfoRprtDeviceType OBJECT-TYPE
    SYNTAX INTEGER {
        reserved(0),
        referenceDesign(1),
        accessPointWirelessRouterSoho(2),
        enterpriseAccessPoint(3),
        broadbandGateway(4),
        digitalStillCamera(5),
        portableVideoCamera(6),
        networkedWebCamera(7),
        digitalAudioStationary(8),
        digitalAudioPortable(9),
        setTopBoxMediaServer(10),
        tvMonitorDigitalPictureFrame(11),
        gameConsoleGameAdaptor(12),
        gamingDevice(13),
        mediaServerMediaAdaptor(14),
        networkStorageDevice(15),
        externalCard(16),
        internalCard(17),
        ultraMobilPc(18),
        notebookComputer(19),
        personalDigitalAssistant(20),
        printerPrintServer(21),
        phoneDualMode(22),
        phoneSingleMode(23),
        smartphoneDualMode(24),
        smartphoneSingleMode(25),
        otherDevices(221)
        }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the type of device in which the IEEE 802.11 STA
        resides."
    ::= { dot11WNMDiagMfrInfoReportEntry 12 }

dot11WNMDiagMfrInfoRprtCertificateID OBJECT-TYPE
```

```
    SYNTAX OCTET STRING (SIZE(0..251))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the Certificate ID for the
        reported Manufacturer Information STA Diagnostic."
    ::= { dot11WNMDiagMfrInfoReportEntry 13 }

-- ********************************************************************
-- * End of dot11WNMDiagMfrInfoReport TABLE
-- ********************************************************************

-- ********************************************************************
-- * dot11WNMDiagConfigProfReport TABLE
-- ********************************************************************
dot11WNMDiagConfigProfReportTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11WNMDiagConfigProfReportEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Group contains the current list of Configuration Profile reports that
        have been received by the MLME. The report tables shall be maintained as
        FIFO to preserve freshness, thus the rows in this table can be deleted for
        memory constraints or other implementation constraints determined by the
        vendor. New rows shall have different RprtIndex values than those deleted
        within the range limitation of the index. One easy way is to monotonically
        increase RprtIndex for new reports being written in the table."
    ::= { dot11WNMReport 10 }

dot11WNMDiagConfigProfReportEntry OBJECT-TYPE
    SYNTAX Dot11WNMDiagConfigProfReportEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry in the dot11WNMDiagConfigProfReportTable Indexed by
        dot11WNMDiagConfigProfRprtIndex."
    INDEX { dot11WNMDiagConfigProfRprtIndex }
    ::= { dot11WNMDiagConfigProfReportTable 1 }

Dot11WNMDiagConfigProfReportEntry ::=
    SEQUENCE {
        dot11WNMDiagConfigProfRprtIndex                       Unsigned32,
        dot11WNMDiagConfigProfRprtRqstToken                   OCTET STRING,
        dot11WNMDiagConfigProfRprtIfIndex                     InterfaceIndex,
        dot11WNMDiagConfigProfRprtEventStatus                 INTEGER,
        dot11WNMDiagConfigProfRprtProfileId                   Unsigned32,
        dot11WNMDiagConfigProfRprtSupportedOperatingClasses   OCTET STRING,
        dot11WNMDiagConfigProfRprtTxPowerMode                 INTEGER,
        dot11WNMDiagConfigProfRprtTxPowerLevels               OCTET STRING,
        dot11WNMDiagConfigProfRprtCipherSuite                 OCTET STRING,
        dot11WNMDiagConfigProfRprtAkmSuite                    OCTET STRING,
        dot11WNMDiagConfigProfRprtEapType                     Unsigned32,
        dot11WNMDiagConfigProfRprtEapVendorID                 OCTET STRING,
        dot11WNMDiagConfigProfRprtEapVendorType               OCTET STRING,
        dot11WNMDiagConfigProfRprtCredentialType              INTEGER,
        dot11WNMDiagConfigProfRprtSSID                        OCTET STRING,
        dot11WNMDiagConfigProfRprtPowerSaveMode               INTEGER }

dot11WNMDiagConfigProfRprtIndex OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Index for Configuration Profile Report elements in
        dot11WNMDiagConfigProfReportTable, greater than 0."
    ::= { dot11WNMDiagConfigProfReportEntry 1 }

dot11WNMDiagConfigProfRprtRqstToken OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the request token that was indicated in the WNM
        request that generated this measurement report. This should be an exact
        match to the original dot11WNMRqstToken attribute. Note that there may be
        multiple entries in the table that match this value since a single request
        may generate multiple WNM reports."
    ::= { dot11WNMDiagConfigProfReportEntry 2 }

dot11WNMDiagConfigProfRprtIfIndex OBJECT-TYPE
    SYNTAX InterfaceIndex
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The ifIndex for this row of WNMDiagConfigProf Report has been received
        on."
    ::= { dot11WNMDiagConfigProfReportEntry 3 }

dot11WNMDiagConfigProfRprtEventStatus OBJECT-TYPE
    SYNTAX INTEGER {
        successful(0),
        requestFailed(1),
        requestRefused(2),
        requestIncapable(3),
        detectedFrequentTransition(4)
        }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute contains the status value included in the Event Report."
    ::= { dot11WNMDiagConfigProfReportEntry 4}

dot11WNMDiagConfigProfRprtProfileId OBJECT-TYPE
    SYNTAX Unsigned32 (0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates a unique identifier for referencing a configura-
        tion profile available on a device. The value of the identifier can be any
        arbitrary value, as long as it is uniquely associated to a single config-
        uration profile on the device sending the identifier."
    ::= { dot11WNMDiagConfigProfReportEntry 5 }

dot11WNMDiagConfigProfRprtSupportedOperatingClasses OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(0..255))
    MAX-ACCESS read-only
```

```
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a management report is completed.

      This attribute indicates the current Operating Class followed by a list of
      each Supported Operating Class, as defined in 8.4.2.56. Each octet con-
      tains an integer representing a operating class. Operating Classes are
      defined in Annex E. The default value is null."
   DEFVAL { ''H }
   ::= { dot11WNMDiagConfigProfReportEntry 6 }

dot11WNMDiagConfigProfRprtTxPowerMode OBJECT-TYPE
   SYNTAX INTEGER {
      fixedPowerMode(0),
      automaticPowerMode(1)
      }
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a management report is completed.

      This attribute indicates the power mode of the STA."
   ::= { dot11WNMDiagConfigProfReportEntry 7 }

dot11WNMDiagConfigProfRprtTxPowerLevels OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(1..255))
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a management report is completed.

      This attribute lists the power levels for the STA. Each octet contains an
      integer representing a power level encoded as a 2's complement value in
      dBm, rounded to the nearest integer. If the Power Mode is automatic, the
      list contains only the minimum and the maximum power levels for the STA.
      If the Power Mode is fixed, the list contains one or more fixed power
      level settings available at this STA, arranged in increasing numerical
      order."
   ::= { dot11WNMDiagConfigProfReportEntry 8 }

dot11WNMDiagConfigProfRprtCipherSuite OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(4))
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a management report is completed.

      This attribute indicates the cipher suite, as defined in Table 8-99. The
      first three octets indicate the OUI. The last octet indicates the suite
      type."
   ::= { dot11WNMDiagConfigProfReportEntry 9 }

dot11WNMDiagConfigProfRprtAkmSuite OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(4))
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a management report is completed.
      This attribute indicates the AKM suite, as defined in Table 8-101 in
      8.4.2.27.3. The first three octets indicate the OUI. The last octet indi-
      cates the suite type."
   ::= { dot11WNMDiagConfigProfReportEntry 10 }

dot11WNMDiagConfigProfRprtEapType OBJECT-TYPE
   SYNTAX Unsigned32 (0..255)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a management report is completed.

      This attribute indicates the single EAP method used by the STA. Valid EAP
      Type numbers are assigned by IANA and are defined at http://www.iana.org/
      assignments/eap-numbers."
   ::= { dot11WNMDiagConfigProfReportEntry 11 }

dot11WNMDiagConfigProfRprtEapVendorID OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(0..3))
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a management report is completed.

      This attribute indicates the EAP Vendor ID number for the EAP method used
      by the STA. The EAP Vendor ID field is included when the EAP Type field is
      set to 254, and is excluded otherwise."
   ::= { dot11WNMDiagConfigProfReportEntry 12 }

dot11WNMDiagConfigProfRprtEapVendorType OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(0..4))
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a management report is completed.

      This attribute indicates the EAP Vendor Type number for the EAP method
      used by the STA. The EAP Vendor Type field is included when the EAP Type
      field is set to 254, and is excluded otherwise."
   ::= { dot11WNMDiagConfigProfReportEntry 13 }

dot11WNMDiagConfigProfRprtCredentialType OBJECT-TYPE
   SYNTAX INTEGER {
      none(0),
      preSharedKey(1),
      userNamePassword(2),
      x509Certificate(3),
      otherCertificate(4),
      oneTimePassword(5),
      token(6)
      }
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a management report is completed.

      This attribute indicates the type of IEEE 802.1X credentials used by the
      STA for this authentication diagnostic."
   ::= { dot11WNMDiagConfigProfReportEntry 14 }
```

```
dot11WNMDiagConfigProfRprtSSID OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(1..32))
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a management report is completed.

      This attribute indicates the SSID for the diagnostic report, as defined in
      8.4.2.2."
   ::= { dot11WNMDiagConfigProfReportEntry 15 }

dot11WNMDiagConfigProfRprtPowerSaveMode OBJECT-TYPE
   SYNTAX INTEGER {
      unknownMode(0),
      none(1),
      psDtims1Mode(2),
      psDtims0Mode(3),
      uapsdMode(4),
      sapsdMode(5),
      upsmpMode(6),
      spsmpMode(7),
      smpsMode(8),
      wnmSleepMode(9),
      fmsMode(10),
      timBroadcastMode(11),
      tfsMode(12),
      tdlsPeerUapsdMode(13),
      tdlsPeerPsmMode(14)
      }
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a management report is completed.

      This attribute indicates the power save mode in use by the STA, as defined
      in Table 8-147."
   ::= { dot11WNMDiagConfigProfReportEntry 16 }

-- **********************************************************************
-- * End of dot11WNMDiagConfigProfReport TABLE
-- **********************************************************************

-- **********************************************************************
-- * dot11WNMDiagAssocReport TABLE
-- **********************************************************************
dot11WNMDiagAssocReportTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11WNMDiagAssocReportEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "Group contains the current list of Association Diagnostic reports that
      have been received by the MLME. The report tables shall be maintained as
      FIFO to preserve freshness, thus the rows in this table can be deleted for
      memory constraints or other implementation constraints determined by the
      vendor. New rows shall have different RprtIndex values than those deleted
      within the range limitation of the index. One easy way is to monotonically
      increase RprtIndex for new reports being written in the table."
   ::= { dot11WNMReport 11 }

dot11WNMDiagAssocReportEntry OBJECT-TYPE
   SYNTAX Dot11WNMDiagAssocReportEntry
   MAX-ACCESS not-accessible
```

```
    STATUS current
    DESCRIPTION
        "An entry in the dot11WNMDiagAssocReportTable Indexed by
        dot11WNMDiagAssocRprtIndex."
    INDEX { dot11WNMDiagAssocRprtIndex }
    ::= { dot11WNMDiagAssocReportTable 1 }

Dot11WNMDiagAssocReportEntry ::=
    SEQUENCE {
        dot11WNMDiagAssocRprtIndex                          Unsigned32,
        dot11WNMDiagAssocRprtRqstToken                      OCTET STRING,
        dot11WNMDiagAssocRprtIfIndex                        InterfaceIndex,
        dot11WNMDiagAssocRprtEventStatus                    INTEGER,
        dot11WNMDiagAssocRprtBssid                          MacAddress,
        dot11WNMDiagAssocRprtOperatingClass                 Unsigned32,
        dot11WNMDiagAssocRprtChannelNumber                  Unsigned32,
        dot11WNMDiagAssocRprtStatusCode                     Unsigned32 }

dot11WNMDiagAssocRprtIndex OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Index for Association Diagnostic Report elements in
        dot11WNMDiagAssocReportTable, greater than 0."
    ::= { dot11WNMDiagAssocReportEntry 1 }

dot11WNMDiagAssocRprtRqstToken OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the request token that was indicated in the WNM
        request that generated this measurement report. This should be an exact
        match to the original dot11WNMRqstToken attribute. Note that there may be
        multiple entries in the table that match this value since a single request
        may generate multiple WNM reports."
    ::= { dot11WNMDiagAssocReportEntry 2 }

dot11WNMDiagAssocRprtIfIndex OBJECT-TYPE
    SYNTAX InterfaceIndex
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The ifIndex for this row of WNMDiagAssoc Report has been received on."
    ::= { dot11WNMDiagAssocReportEntry 3 }

dot11WNMDiagAssocRprtEventStatus OBJECT-TYPE
    SYNTAX INTEGER {
        successful(0),
        requestFailed(1),
        requestRefused(2),
        requestIncapable(3),
        detectedFrequentTransition(4)
        }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.
```

```
      This attribute contains the status value included in the Event Report."
   ::= { dot11WNMDiagAssocReportEntry 4 }

dot11WNMDiagAssocRprtBssid OBJECT-TYPE
   SYNTAX MacAddress
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a management report is completed.

      This attribute indicates the BSSID for the target AP for this Association
      Diagnostic Report."
   ::= { dot11WNMDiagAssocReportEntry 5 }

dot11WNMDiagAssocRprtOperatingClass OBJECT-TYPE
   SYNTAX Unsigned32(1..255)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a management report is completed.

      This attribute indicates the channel set for the target AP for this Asso-
      ciation Diagnostic Report. Country, Operating Class and Channel Number
      together specify the channel frequency and spacing for this measurement
      request. Valid values of Operating Class as shown in Annex E."
   ::= { dot11WNMDiagAssocReportEntry 6 }

dot11WNMDiagAssocRprtChannelNumber OBJECT-TYPE
   SYNTAX Unsigned32 (1..255)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a management report is completed.

      This attribute indicates the operating channel of the target AP for this
      Association Diagnostic Report. The Channel Number is only defined within
      the indicated Operating Class as sown in Annex E."
   ::= { dot11WNMDiagAssocReportEntry 7 }

dot11WNMDiagAssocRprtStatusCode OBJECT-TYPE
   SYNTAX Unsigned32(0..65535)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a management report is completed.

      This attribute indicates the result of the association diagnostic and is
      set to one of the status codes specified in Table 8-37 in 8.4.1.9."
   ::= { dot11WNMDiagAssocReportEntry 8 }

-- ********************************************************************
-- * End of dot11WNMDiagAssocReport TABLE
-- ********************************************************************

-- ********************************************************************
-- * dot11WNMDiag8021xAuthReport TABLE
-- ********************************************************************
dot11WNMDiag8021xAuthReportTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11WNMDiag8021xAuthReportEntry
   MAX-ACCESS not-accessible
```

```
    STATUS current
    DESCRIPTION
        "Group contains the current list of IEEE 802.1X Authentication Diagnostic
        reports that have been received by the MLME. The report tables shall be
        maintained as FIFO to preserve freshness, thus the rows in this table can
        be deleted for memory constraints or other implementation constraints
        determined by the vendor. New rows shall have different RprtIndex values
        than those deleted within the range limitation of the index. One easy way
        is to monotonically increase RprtIndex for new reports being written in
        the table."
    ::= { dot11WNMReport 12 }

dot11WNMDiag8021xAuthReportEntry OBJECT-TYPE
    SYNTAX Dot11WNMDiag8021xAuthReportEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry in the dot11WNMDiag8021xAuthReportTable Indexed by
        dot11WNMDiag8021xAuthRprtIndex."
    INDEX { dot11WNMDiag8021xAuthRprtIndex }
    ::= { dot11WNMDiag8021xAuthReportTable 1 }

Dot11WNMDiag8021xAuthReportEntry ::=
    SEQUENCE {
        dot11WNMDiag8021xAuthRprtIndex                          Unsigned32,
        dot11WNMDiag8021xAuthRprtRqstToken                      OCTET STRING,
        dot11WNMDiag8021xAuthRprtIfIndex                        InterfaceIndex,
        dot11WNMDiag8021xAuthRprtEventStatus                    INTEGER,
        dot11WNMDiag8021xAuthRprtBssid                          MacAddress,
        dot11WNMDiag8021xAuthRprtOperatingClass                 Unsigned32,
        dot11WNMDiag8021xAuthRprtChannelNumber                  Unsigned32,
        dot11WNMDiag8021xAuthRprtEapType                        Unsigned32,
        dot11WNMDiag8021xAuthRprtEapVendorID                    OCTET STRING,
        dot11WNMDiag8021xAuthRprtEapVendorType                  OCTET STRING,
        dot11WNMDiag8021xAuthRprtCredentialType                 INTEGER,
        dot11WNMDiag8021xAuthRprtStatusCode                     Unsigned32 }

dot11WNMDiag8021xAuthRprtIndex OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Index for IEEE 802.1X Authentication Diagnostic Report elements in
        dot11WNMDiag8021xAuthReportTable, greater than 0."
    ::= { dot11WNMDiag8021xAuthReportEntry 1 }

dot11WNMDiag8021xAuthRprtRqstToken OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the request token that was indicated in the WNM
        request that generated this measurement report. This should be an exact
        match to the original dot11WNMRqstToken attribute. Note that there may be
        multiple entries in the table that match this value since a single request
        may generate multiple WNM reports."
    ::= { dot11WNMDiag8021xAuthReportEntry 2 }

dot11WNMDiag8021xAuthRprtIfIndex OBJECT-TYPE
    SYNTAX InterfaceIndex
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The ifIndex for this row of WNMDiag8021xAuth Report has been received
        on."
    ::= { dot11WNMDiag8021xAuthReportEntry 3 }

dot11WNMDiag8021xAuthRprtEventStatus OBJECT-TYPE
    SYNTAX INTEGER {
        successful(0),
        requestFailed(1),
        requestRefused(2),
        requestIncapable(3),
        detectedFrequentTransition(4)
        }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute contains the status value included in the Event Report."
    ::= { dot11WNMDiag8021xAuthReportEntry 4 }

dot11WNMDiag8021xAuthRprtBssid OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the BSSID for the target AP for this Authentica-
        tion Diagnostic Report."
    ::= { dot11WNMDiag8021xAuthReportEntry 5 }

dot11WNMDiag8021xAuthRprtOperatingClass OBJECT-TYPE
    SYNTAX Unsigned32(1..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the channel set for the target AP for this
        Authentication Diagnostic Report. Country, Operating Class and Channel
        Number together specify the channel frequency and spacing for this mea-
        surement request. Valid values of Operating Class as shown in Annex E."
    ::= { dot11WNMDiag8021xAuthReportEntry 6 }

dot11WNMDiag8021xAuthRprtChannelNumber OBJECT-TYPE
    SYNTAX Unsigned32 (1..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the operating channel of the target AP for this
        Authentication Diagnostic Report. The Channel Number is only defined
        within the indicated Operating Class as shown in Annex E."
    ::= { dot11WNMDiag8021xAuthReportEntry 7 }

dot11WNMDiag8021xAuthRprtEapType OBJECT-TYPE
    SYNTAX Unsigned32 (0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the single EAP method used by the STA. Valid EAP
        Type numbers are assigned by IANA and are defined at http://www.iana.org/
        assignments/eap-numbers."
    ::= { dot11WNMDiag8021xAuthReportEntry 8 }

dot11WNMDiag8021xAuthRprtEapVendorID OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(0..3))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the EAP Vendor ID number for the EAP method used
        by the STA. The EAP Vendor ID field is included when the EAP Type field is
        set to 254, and is excluded otherwise."
    ::= { dot11WNMDiag8021xAuthReportEntry 9 }

dot11WNMDiag8021xAuthRprtEapVendorType OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(0..4))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the EAP Vendor Type number for the EAP method
        used by the STA. The EAP Vendor Type field is included when the EAP Type
        field is set to 254, and is excluded otherwise."
    ::= { dot11WNMDiag8021xAuthReportEntry 10 }

dot11WNMDiag8021xAuthRprtCredentialType OBJECT-TYPE
    SYNTAX INTEGER {
        none(0),
        preSharedKey(1),
        userNamePassword(2),
        x509Certificate(3),
        otherCertificate(4),
        oneTimePassword(5),
        token(6)
        }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the type of IEEE 802.1X credentials used by the
        STA for this authentication diagnostic."
    ::= { dot11WNMDiag8021xAuthReportEntry 11 }

dot11WNMDiag8021xAuthRprtStatusCode OBJECT-TYPE
    SYNTAX Unsigned32 (0..65535)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.
```

```
      This attribute indicates the result of the authentication diagnostic and
      is set to one of the status codes specified in Table 8-37."
   ::= { dot11WNMDiag8021xAuthReportEntry 12 }

-- ********************************************************************
-- * End of dot11WNMDiag8021xAuthReport TABLE
-- ********************************************************************

-- ********************************************************************
-- * dot11WNMLocConfigReport TABLE
-- ********************************************************************
dot11WNMLocConfigReportTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11WNMLocConfigReportEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "Group contains the current list of Location Configuration reports that
      have been received by the MLME. The report tables shall be maintained as
      FIFO to preserve freshness, thus the rows in this table can be deleted for
      memory constraints or other implementation constraints determined by the
      vendor. New rows shall have different RprtIndex values than those deleted
      within the range limitation of the index. One easy way is to monotonically
      increase RprtIndex for new reports being written in the table."
   ::= { dot11WNMReport 13 }

dot11WNMLocConfigReportEntry OBJECT-TYPE
   SYNTAX Dot11WNMLocConfigReportEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "An entry in the dot11WNMLocConfigReportTable Indexed by
      dot11WNMLocConfigRprtIndex."
   INDEX { dot11WNMLocConfigRprtIndex }
   ::= { dot11WNMLocConfigReportTable 1 }

Dot11WNMLocConfigReportEntry ::=
   SEQUENCE {
      dot11WNMLocConfigRprtIndex                             Unsigned32,
      dot11WNMLocConfigRprtRqstToken                         OCTET STRING,
      dot11WNMLocConfigRprtIfIndex                           InterfaceIndex,
      dot11WNMLocConfigRprtLocIndParams                      OCTET STRING,
      dot11WNMLocConfigRprtLocIndChanList                    OCTET STRING,
      dot11WNMLocConfigRprtLocIndBcastRate                   Unsigned32,
      dot11WNMLocConfigRprtLocIndOptions                     OCTET STRING,
      dot11WNMLocConfigRprtStatusConfigSubelemId             INTEGER,
      dot11WNMLocConfigRprtStatusResult                      INTEGER,
      dot11WNMLocConfigRprtVendorSpecificRprtContent         OCTET STRING }

dot11WNMLocConfigRprtIndex OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "Index for Location Configuration Report elements in
      dot11WNMLocConfigReportTable, greater than 0."
   ::= { dot11WNMLocConfigReportEntry 1 }

dot11WNMLocConfigRprtRqstToken OBJECT-TYPE
   SYNTAX OCTET STRING
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
```

Copyright © 2012 IEEE. All rights reserved.

```
       It is written by the SME when a management report is completed.

       This attribute indicates the request token that was indicated in the WNM
       request that generated this measurement report. This should be an exact
       match to the original dot11WNMRqstToken attribute. Note that there may be
       multiple entries in the table that match this value since a single request
       may generate multiple WNM reports."
    ::= { dot11WNMLocConfigReportEntry 2 }

dot11WNMLocConfigRprtIfIndex OBJECT-TYPE
    SYNTAX InterfaceIndex
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
       "The ifIndex for this row of WNMLocConfig Report has been received on."
    ::= { dot11WNMLocConfigReportEntry 3 }

dot11WNMLocConfigRprtLocIndParams OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(16))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
       "This is a status variable.
       It is written by the SME when a management report is completed.

       This attribute indicates STA Location reporting characteristics. The for-
       mat of these Location Indication Parameters are detailed in 8.4.2.73.2."
    ::= { dot11WNMLocConfigReportEntry 4 }

dot11WNMLocConfigRprtLocIndChanList OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(0..254))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
       "This is a status variable.
       It is written by the SME when a management report is completed.

       This attribute lists location indication reporting channel information for
       this Location Configuration Report. The default value is null. Each pair
       of octets indicates a different Operating Class and channel number for
       this request. The detailed format for this list of channels as described
       in 8.4.2.73.3."
    DEFVAL { ''H }
    ::= { dot11WNMLocConfigReportEntry 5 }

dot11WNMLocConfigRprtLocIndBcastRate OBJECT-TYPE
    SYNTAX Unsigned32 (0..65535)
    UNITS "0.5Mb/s"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
       "This is a status variable.
       It is written by the SME when a management report is completed.

       This attribute indicates the data rate, in 0.5Mb/s units, at which the STA
       broadcasts its Location Track Notification frames."
    ::= { dot11WNMLocConfigReportEntry 6 }

dot11WNMLocConfigRprtLocIndOptions OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(0..255))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
       "This is a status variable.
```

```
        It is written by the SME when a management report is completed.

        This attribute indicates the location track indication options used; see
        8.4.2.73.9."
    ::= { dot11WNMLocConfigReportEntry 7}

dot11WNMLocConfigRprtStatusConfigSubelemId OBJECT-TYPE
    SYNTAX INTEGER {
        multipleSubelemIds(0),
        locationIndicationParams(1),
        locationIndicationChannels(2),
        locationStatus(3),
        radioInformation(4),
        motion(5),
        locationIndicationBcastDataRate(6),
        timeOfDeparture(7),
        vendorSpecific(8)
        }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute is set to a specific Location Parameters subelement ID
        transmitted in a Location Configuration Request frame. If the following
        StatusResult attribute field value applies to more than one subelement
        then the Config subelement ID is set to 0. If the Status field value
        applies to one subelement, then a Location Status subelement may be
        included in the Location Configuration Response for each configuration
        subelement that has a non-Success Status value."
    ::= { dot11WNMLocConfigReportEntry 8 }

dot11WNMLocConfigRprtStatusResult OBJECT-TYPE
    SYNTAX INTEGER {
        successful(0),
        requestFailed(1),
        requestRefused(2),
        requestIncapable(3),
        detectedFrequentTransition(4)
        }

    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute contains the resulting status of the Location Configuration
        Request frame for the indicated Location Parameter subelement ID, as
        listed in Table 8-137, Event Report Status."
    ::= { dot11WNMLocConfigReportEntry 9 }

dot11WNMLocConfigRprtVendorSpecificRprtContent OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(0..255))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute provides an envelope for all the vendor specific subele-
        ments that may be included in Location Configuration Report element. The
        default value is null."
    DEFVAL { ''H }
    ::= { dot11WNMLocConfigReportEntry 10 }

-- **********************************************************************
-- * End of dot11WNMLocConfigReport TABLE
-- **********************************************************************

-- **********************************************************************
-- * dot11WNMBssTransitReport TABLE
-- **********************************************************************
dot11WNMBssTransitReportTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11WNMBssTransitReportEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Group contains the current list of BSS Transition Management reports that
        have been received by the MLME. The report tables shall be maintained as
        FIFO to preserve freshness, thus the rows in this table can be deleted for
        memory constraints or other implementation constraints determined by the
        vendor. New rows shall have different RprtIndex values than those deleted
        within the range limitation of the index. One easy way is to monotonically
        increase RprtIndex for new reports being written in the table."
    ::= { dot11WNMReport 14 }

dot11WNMBssTransitReportEntry OBJECT-TYPE
    SYNTAX Dot11WNMBssTransitReportEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry in the dot11WNMBssTransitReportTable Indexed by
        dot11WNMBssTransitRprtIndex."
    INDEX { dot11WNMBssTransitRprtIndex }
    ::= { dot11WNMBssTransitReportTable 1 }

Dot11WNMBssTransitReportEntry ::=
    SEQUENCE {
        dot11WNMBssTransitRprtIndex                         Unsigned32,
        dot11WNMBssTransitRprtRqstToken                     OCTET STRING,
        dot11WNMBssTransitRprtIfIndex                       InterfaceIndex,
        dot11WNMBssTransitRprtStatusCode                    INTEGER,
        dot11WNMBssTransitRprtBSSTerminationDelay           Unsigned32,
        dot11WNMBssTransitRprtTargetBssid                   MacAddress,
        dot11WNMBssTransitRprtCandidateList                 OCTET STRING }

dot11WNMBssTransitRprtIndex OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Index for BSS Transition Management Report elements in
        dot11WNMBssTransitReportTable, greater than 0."
    ::= { dot11WNMBssTransitReportEntry 1 }

dot11WNMBssTransitRprtRqstToken OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the request token that was indicated in the WNM
        request that generated this measurement report. This should be an exact
        match to the original dot11WNMRqstToken attribute. Note that there may be
```

```
      multiple entries in the table that match this value since a single request
      may generate multiple WNM reports."
   ::= { dot11WNMBssTransitReportEntry 2 }

dot11WNMBssTransitRprtIfIndex OBJECT-TYPE
   SYNTAX InterfaceIndex
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "The ifIndex for this row of WNMBssTransit Report has been received on."
   ::= { dot11WNMBssTransitReportEntry 3 }

dot11WNMBssTransitRprtStatusCode OBJECT-TYPE
   SYNTAX INTEGER {
      accept(0),
      rejectUnspecified(1),
      rejectInsufficientBeacons(2),
      rejectInsufficientCapacity(3),
      rejectBssTerminationUndesired(4),
      rejectBssTerminationDelayRequest(5),
      rejectBssTransitionCandidateListProvided(6),
      rejectNoSuitableBssTransitionCandidates(7),
      rejectLeavingEss(8)
      }
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a management report is completed.

      This attribute indicates the status of this BSS Transition report."
   ::= { dot11WNMBssTransitReportEntry 4 }

dot11WNMBssTransitRprtBSSTerminationDelay OBJECT-TYPE
   SYNTAX Unsigned32 (1..255)
   UNITS "minutes"
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a management report is completed.

      This attribute indicates the number of minutes that the responding STA
      requests the BSS to delay termination. This attribute is included only if
      the Status Code field value is set to 5."
   ::= { dot11WNMBssTransitReportEntry 5 }

dot11WNMBssTransitRprtTargetBssid OBJECT-TYPE
   SYNTAX MacAddress
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a management report is completed.

      This attribute indicates the target BSSID for this BSS Transition Report."
   ::= { dot11WNMBssTransitReportEntry 6 }

dot11WNMBssTransitRprtCandidateList OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(0..2304))
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when a management report is completed.

      This attribute lists one or more Neighbor Report elements which are BSS
      transition candidates for this request.  The Neighbore Report elements are
      described in 8.4.2.39."
::= { dot11WNMBssTransitReportEntry 7 }


-- ********************************************************************
-- * End of dot11WNMBssTransitReport TABLE
-- ********************************************************************

-- ********************************************************************
-- * dot11WNMColocInterfReport TABLE
-- ********************************************************************
dot11WNMColocInterfReportTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11WNMColocInterfReportEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "Group contains the current list of Collocated Interference reports that
      have been received by the MLME. The report tables shall be maintained as
      FIFO to preserve freshness, thus the rows in this table can be deleted for
      memory constraints or other implementation constraints determined by the
      vendor. New rows shall have different RprtIndex values than those deleted
      within the range limitation of the index. One easy way is to monotonically
      increase RprtIndex for new reports being written in the table."
   ::= { dot11WNMReport 16 }

dot11WNMColocInterfReportEntry OBJECT-TYPE
   SYNTAX Dot11WNMColocInterfReportEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "An entry in the dot11WNMColocInterfReportTable Indexed by
      dot11WNMColocInterfRprtIndex."
   INDEX { dot11WNMColocInterfRprtIndex }
   ::= { dot11WNMColocInterfReportTable 1 }

Dot11WNMColocInterfReportEntry ::=
   SEQUENCE {
      dot11WNMColocInterfRprtIndex                           Unsigned32,
      dot11WNMColocInterfRprtRqstToken                       OCTET STRING,
      dot11WNMColocInterfRprtIfIndex                         InterfaceIndex,
      dot11WNMColocInterfRprtPeriod                          Unsigned32,
      dot11WNMColocInterfRprtInterfLevel                     Integer32,
      dot11WNMColocInterfRprtInterfAccuracy                  Unsigned32,
      dot11WNMColocInterfRprtInterfIndex                     Unsigned32,
      dot11WNMColocInterfRprtInterfInterval                  Integer32,
      dot11WNMColocInterfRprtInterfBurstLength               Integer32,
      dot11WNMColocInterfRprtInterfStartTime                 Integer32,
      dot11WNMColocInterfRprtInterfCenterFreq                Integer32,
      dot11WNMColocInterfRprtInterfBandwidth                 Unsigned32 }

dot11WNMColocInterfRprtIndex OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "Index for Collocated Interference Report elements in
      dot11WNMColocInterfReportTable, greater than 0."
   ::= { dot11WNMColocInterfReportEntry 1 }

dot11WNMColocInterfRprtRqstToken OBJECT-TYPE
```

```
    SYNTAX OCTET STRING
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the request token that was indicated in the WNM
        request that generated this measurement report. This should be an exact
        match to the original dot11WNMRqstToken attribute. Note that there may be
        multiple entries in the table that match this value since a single request
        may generate multiple WNM reports."
    ::= { dot11WNMColocInterfReportEntry 2 }

dot11WNMColocInterfRprtIfIndex OBJECT-TYPE
    SYNTAX InterfaceIndex
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        The ifIndex for this row of WNMColocInterf Report has been received on."
    ::= { dot11WNMColocInterfReportEntry 3 }

dot11WNMColocInterfRprtPeriod OBJECT-TYPE
    SYNTAX Unsigned32(0..255)
    UNITS "100 TU"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates how often the STA periodically reports the collo-
        cated interference. The field is in units of 100 TUs. If the Report Period
        field is set to 0, then the reporting is not periodic, and a report is
        generated when the STA detects a change in the collocated interference.
        See 10.23.8 for further details."
    ::= { dot11WNMColocInterfReportEntry 4 }

dot11WNMColocInterfRprtInterfLevel OBJECT-TYPE
    SYNTAX Integer32(-128..127)
    UNITS "dBm"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute contains a signed integer indicating the maximum level of
        the collocated interference power in units of dBm over all receive chains
        averaged over a 4 s period during an interference period and across inter-
        ference bandwidth. When the interference level is unknown, the field is
        set to +127 dBm. When the interference level is equal or greater than 126
        dBm, the field is set to +126 dBm. If no collocated interference is pres-
        ent the field is set to -128 dBm. When the interference level is equal or
        lower than -127 dBm, the field is set to -127 dBm. The interference level
        is referenced to the antenna connector (see definition in Clause 3) used
        for reception, like RCPI."
    ::= { dot11WNMColocInterfReportEntry 5 }

dot11WNMColocInterfRprtInterfAccuracy OBJECT-TYPE
    SYNTAX Unsigned32(0..15)
    UNITS "dB"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates an unsigned integer indicating the expected accu-
        racy of the estimate of interference in dB with 95% confidence interval.
        If the Interference Level field is X (dBm) and the expected accuracy field
        is Y (dB), the actual interference level is in the range of [X - Y, X +Y]
        with the probability of 95%. If the accuracy is unknown then the Expected
        Accuracy field is set to 15."
    ::= { dot11WNMColocInterfReportEntry 6 }

dot11WNMColocInterfRprtInterfIndex OBJECT-TYPE
    SYNTAX Unsigned32(0..15)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the interference index that is unique for each
        type of interference source. The field set to 0 indicates that no collo-
        cated interference is present. See 10.23.8 for further details."
    ::= { dot11WNMColocInterfReportEntry 7 }

dot11WNMColocInterfRprtInterfInterval OBJECT-TYPE
    SYNTAX Integer32
    UNITS "microseconds"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the interval between two successive periods of
        interference in microseconds. When the interval between two successive
        periods of interference is variable the field is set to 2E32-1. When the
        interval between two successive periods of interference is equal or
        greater than 2E32-2 the field is set to 2E32-2. If no collocated interfer-
        ence is present the field is set to 0."
    ::= { dot11WNMColocInterfReportEntry 8 }

dot11WNMColocInterfRprtInterfBurstLength OBJECT-TYPE
    SYNTAX Integer32
    UNITS "microseconds"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the duration of each period of interference in
        microseconds. When the duration of each period of interference is variable
        the field is set to 2E32-1). When the duration of each period of interfer-
        ence is equal or greater than 2E32-2, the field is set to 2E32-2. If no
        collocated interference is present the field is set to 0."
    ::= { dot11WNMColocInterfReportEntry 9 }

dot11WNMColocInterfRprtInterfStartTime OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute contains the least significant 4 octets (i.e. B0-B31) of
        the TSF timer at the start of the interference burst. When either the
        Interference Interval or the Interference Burst Length fields are set to
        2E32-1, this field indicates the average duty cycle. The average duty
        cycle value is defined as Round-to-Integer ((2E32-2)[average interference
        burst length (microsecond)]/[average interference interval (microsec-
        ond)]). When the interference is nonperiodic the Interference Start Time
        field is set to 0. If no collocated interference is present the field is
        set to 0."
    ::= { dot11WNMColocInterfReportEntry 10 }

dot11WNMColocInterfRprtInterfCenterFreq OBJECT-TYPE
    SYNTAX Integer32
    UNITS "5 kHz"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates center frequency of interference in units of 5
        kHz. When center frequency is unknown, the center frequency of the STA's
        operating channel is reported. If no collocated interference is present
        the field is set to 0."
    ::= { dot11WNMColocInterfReportEntry 11 }

dot11WNMColocInterfRprtInterfBandwidth OBJECT-TYPE
    SYNTAX Unsigned32 (0..65535)
    UNITS "5 kHz"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when a management report is completed.

        This attribute indicates the bandwidth at the -3dB roll-off point of the
        interference signal in 5 kHz. When bandwidth of the interference signal is
        unknown, the field is set to 65 535. When bandwidth of the interference
        signal is equal or greater than 65 534 the field is set to 65 534. If no
        collocated interference is present the field is set to 0."
    ::= { dot11WNMColocInterfReportEntry 12 }

-- ********************************************************************
-- * End of dot11WNMColocInterfReport TABLE
-- ********************************************************************

-- ********************************************************************
-- * END of Wireless Network Management Interface MIB
-- ********************************************************************

-- **********************************************************************
-- * END of IEEE 802.11 RM and WNM Interface MIB
-- **********************************************************************
-- **********************************************************************
-- * dot11MeshSTAConfig TABLE
-- **********************************************************************

dot11MeshSTAConfigTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11MeshSTAConfigEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "Mesh Station Configuration attributes. In tabular form to allow for mul-
      tiple instances on an agent."
   ::= { dot11smt 23 }

dot11MeshSTAConfigEntry  OBJECT-TYPE
   SYNTAX Dot11MeshSTAConfigEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "An entry in the dot11MeshStationConfigTable. It is possible for there to
      be multiple IEEE 802.11 interfaces on one agent, each with its unique MAC
      address. The relationship between an IEEE 802.11 interface and an inter-
      face in the context of the Internet-standard MIB is one-to-one. As such,
      the value of an ifIndex object instance can be directly used to identify
      corresponding instances of the objects defined herein.
      ifIndex - Each IEEE 802.11 interface is represented by an ifEntry. Inter-
      face tables in this MIB module are indexed by ifIndex."
   INDEX { ifIndex }
   ::= { dot11MeshSTAConfigTable 1 }

Dot11MeshSTAConfigEntry ::=
   SEQUENCE {
      dot11MeshID                                       OCTET STRING,
      dot11MeshNumberOfPeerings                         Unsigned32,
      dot11MeshAcceptingAdditionalPeerings              TruthValue,
      dot11MeshConnectedToMeshGate                      TruthValue,
      dot11MeshSecurityActivated                        TruthValue,
      dot11MeshActiveAuthenticationProtocol             INTEGER,
      dot11MeshMaxRetries                               Unsigned32,
      dot11MeshRetryTimeout                             Unsigned32,
      dot11MeshConfirmTimeout                           Unsigned32,
      dot11MeshHoldingTimeout                           Unsigned32,
      dot11MeshConfigGroupUpdateCount                   Unsigned32,
      dot11MeshActivePathSelectionProtocol              INTEGER,
      dot11MeshActivePathSelectionMetric                INTEGER,
      dot11MeshForwarding                               TruthValue,
      dot11MeshTTL                                      Unsigned32,
      dot11MeshGateAnnouncements                        TruthValue,
      dot11MeshGateAnnouncementInterval                 Unsigned32,
      dot11MeshActiveCongestionControlMode              INTEGER,
      dot11MeshActiveSynchronizationMethod              INTEGER,
      dot11MeshNbrOffsetMaxNeighbor                     Unsigned32,
      dot11MBCAActivated                                TruthValue,
      dot11MeshBeaconTimingReportInterval               Unsigned32,
      dot11MeshBeaconTimingReportMaxNum                 Unsigned32,
      dot11MeshDelayedBeaconTxInterval                  Unsigned32,
      dot11MeshDelayedBeaconTxMaxDelay                  Unsigned32,
      dot11MeshDelayedBeaconTxMinDelay                  Unsigned32,
      dot11MeshAverageBeaconFrameDuration               Unsigned32,
      dot11MeshSTAMissingAckRetryLimit                  Unsigned32,
      dot11MeshAwakeWindowDuration                      Unsigned32,
      dot11MCCAImplemented                              TruthValue,
      dot11MCCAActivated                                TruthValue,
      dot11MAFlimit                                     Unsigned32,
      dot11MCCAScanDuration                             Unsigned32,
      dot11MCCAAdvertPeriodMax                               Unsigned32,
      dot11MCCAMinTrackStates                                Unsigned32,
      dot11MCCAMaxTrackStates                                Unsigned32,
      dot11MCCAOPtimeout                                     Unsigned32,
      dot11MCCACWmin                                         Unsigned32,
      dot11MCCACWmax                                         Unsigned32,
      dot11MCCAAIFSN                                         Unsigned32
      }


dot11MeshID OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(0..32))
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect for the next MLME-START.request.

      This attribute reflects the Mesh ID configured in this entity."
   ::= { dot11MeshSTAConfigEntry 1 }

dot11MeshNumberOfPeerings OBJECT-TYPE
   SYNTAX Unsigned32 (0..255)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute indicates the number of mesh peering currently maintained
      by the STA. This value is reflected in the Number of Peerings subfield in
      the Mesh Formation Info field in the Mesh Configuration element."
   ::= { dot11MeshSTAConfigEntry 2 }

dot11MeshAcceptingAdditionalPeerings OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute indicates whether the station is willing to accept addi-
      tional peerings. This value is reflected in the Accepting Additional Mesh
      Peerings subfield in the Mesh Capability field in the Mesh Configuration
      element."
   ::= { dot11MeshSTAConfigEntry 3 }

dot11MeshConnectedToMeshGate OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute indicates whether the station has a mesh path to a mesh
      gate. This value is reflected in the Connected to Mesh Gate subfield in
      the Mesh Formation Info field in the Mesh Configuration element."
   ::= { dot11MeshSTAConfigEntry 4 }
```

```
dot11MeshSecurityActivated OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect for the next MLME-START.request.

      This attribute specifies whether the station is security enabled."
   ::= { dot11MeshSTAConfigEntry 5 }

dot11MeshActiveAuthenticationProtocol OBJECT-TYPE
   SYNTAX INTEGER {
      null (0),
      sae (1),
      ieee8021x (2),
      vendorSpecific (255) }
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect for the next MLME-START.request.

      This attribute specifies the active authentication protocol."
   DEFVAL { null }
   ::= { dot11MeshSTAConfigEntry 6 }

dot11MeshMaxRetries OBJECT-TYPE
   SYNTAX Unsigned32 (0..16)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute specifies the maximum number of Mesh Peering Open retries
      that can be sent to establish a new mesh peering instance in a mesh BSS."
   DEFVAL { 2 }
   ::= { dot11MeshSTAConfigEntry 7 }

dot11MeshRetryTimeout OBJECT-TYPE
   SYNTAX Unsigned32 (1..255)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute specifies the initial retry timeout, in millisecond units,
      used by the Mesh Peering Open message."
   DEFVAL { 40 }
   ::= { dot11MeshSTAConfigEntry 8 }

dot11MeshConfirmTimeout OBJECT-TYPE
   SYNTAX Unsigned32 (1..255)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
```

```
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute specifies the initial retry timeout, in millisecond units,
        used by the Mesh Peering Open message."
    DEFVAL { 40 }
    ::= { dot11MeshSTAConfigEntry 9 }

dot11MeshHoldingTimeout OBJECT-TYPE
    SYNTAX Unsigned32 (1..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute specifies the confirm timeout, in millisecond units, used
        by the mesh peering management to close a mesh peering."
    DEFVAL { 40 }
    ::= { dot11MeshSTAConfigEntry 10 }

dot11MeshConfigGroupUpdateCount OBJECT-TYPE
    SYNTAX Unsigned32 (1..4294967295)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute specifies how many times the Mesh Group Key Inform frame
        will be retried per mesh group key handshake attempt."
    DEFVAL { 3 }
    ::= { dot11MeshSTAConfigEntry 11 }

dot11MeshActivePathSelectionProtocol OBJECT-TYPE
    SYNTAX INTEGER { hwmp (1), vendorSpecific (255) }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect for the next MLME-START.request.

        This attribute specifies the active path selection protocol."
    DEFVAL { hwmp }
    ::= { dot11MeshSTAConfigEntry 12 }

dot11MeshActivePathSelectionMetric OBJECT-TYPE
    SYNTAX INTEGER { airtimeLinkMetric (1), vendorSpecific (255) }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect for the next MLME-START.request.

        This attribute specifies the active path selection metric."
    DEFVAL { airtimeLinkMetric }
    ::= { dot11MeshSTAConfigEntry 13 }

dot11MeshForwarding OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute specifies the ability of a mesh STA to forward MSDUs."
    DEFVAL { true }
    ::= { dot11MeshSTAConfigEntry 14 }

dot11MeshTTL OBJECT-TYPE
    SYNTAX Unsigned32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute specifies the value of Mesh TTL subfield set at a source
        mesh STA."
    DEFVAL { 31 }
    ::= { dot11MeshSTAConfigEntry 15 }

dot11MeshGateAnnouncements OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute specifies whether the mesh STA activates mesh gate
        announcements."
    DEFVAL { false }
    ::= { dot11MeshSTAConfigEntry 16 }

dot11MeshGateAnnouncementInterval OBJECT-TYPE
    SYNTAX Unsigned32 (1..65535)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute specifies the gate announcement interval. The gate
        announcement interval is the number of seconds between the transmission of
        two gate announcements."
    DEFVAL { 10 }
    ::= { dot11MeshSTAConfigEntry 17 }

dot11MeshActiveCongestionControlMode OBJECT-TYPE
    SYNTAX INTEGER {
        null (0),
        congestionControlSignaling (1),
        vendorSpecific (255) }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
```

```
      Changes take effect for the next MLME-START.request.

      This attribute specifies the active congestion control protocol."
   DEFVAL { null }
   ::= { dot11MeshSTAConfigEntry 18 }

dot11MeshActiveSynchronizationMethod OBJECT-TYPE
   SYNTAX INTEGER {
      neighborOffsetSynchronization (1),
      vendorSpecific (255) }
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect for the next MLME-START.request.

      This attribute specifies the MBSS’s active synchronization method."
   DEFVAL { neighborOffsetSynchronization }
   ::= { dot11MeshSTAConfigEntry 19 }

dot11MeshNbrOffsetMaxNeighbor OBJECT-TYPE
   SYNTAX Unsigned32 (1..255)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      This attribute specifies the maximum number of neighbor STAs with which
      the mesh STA maintains synchronization using the neighbor offset synchro-
      nization method."
   DEFVAL { 50 }
   ::= { dot11MeshSTAConfigEntry 20 }

dot11MBCAActivated OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute specifies whether the station activates mesh beacon colli-
      sion avoidance mechanisms."
   DEFVAL { false }
   ::= { dot11MeshSTAConfigEntry 21 }

dot11MeshBeaconTimingReportInterval OBJECT-TYPE
   SYNTAX Unsigned32 (1..255)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.

      This attribute specifies when the Beacon Timing element is present in Bea-
      con frames. The Beacon Timing element is present when the DTIM Count value
      in the Beacon frame is zero or equal to an integer multiple of the set
      value."
   DEFVAL { 4 }
   ::= { dot11MeshSTAConfigEntry 22 }
```

```
dot11MeshBeaconTimingReportMaxNum OBJECT-TYPE
   SYNTAX Unsigned32 (0..50)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute specifies the maximum number of the Beacon Timing Informa-
      tion field contained in a Beacon Timing element in the transmitting Beacon
      frames."
   DEFVAL { 16 }
   ::= { dot11MeshSTAConfigEntry 23 }

dot11MeshDelayedBeaconTxInterval OBJECT-TYPE
   SYNTAX Unsigned32 (0..255)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute specifies the interval of the delayed beacon transmission
      for the purpose of MBCA. The value is expressed in units of Beacon Inter-
      val. The value 0 indicates that the delayed beacon transmission is dis-
      abled."
   DEFVAL { 0 }
   ::= { dot11MeshSTAConfigEntry 24 }

dot11MeshDelayedBeaconTxMaxDelay OBJECT-TYPE
   SYNTAX Unsigned32 (0..65535)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute specifies the maximum delay time from a TBTT of delayed
      beacon transmissions for the purpose of MBCA. The value is expressed in
      units of microseconds."
   DEFVAL { 2048 }
   ::= { dot11MeshSTAConfigEntry 25 }

dot11MeshDelayedBeaconTxMinDelay OBJECT-TYPE
   SYNTAX Unsigned32 (0..4023)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute specifies the minimum delay time from a TBTT of delayed
      beacon transmissions for the purpose of MBCA. The value is expressed in
      units of microseconds."
   DEFVAL { 0 }
   ::= { dot11MeshSTAConfigEntry 26 }

dot11MeshAverageBeaconFrameDuration OBJECT-TYPE
   SYNTAX Unsigned32 (0..16383)
   MAX-ACCESS read-write
```

```
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute specifies the average duration of the last 16 Beacon frames
        of other mesh STAs received by this mesh STA. The value is expressed in
        units of microseconds."
    ::= { dot11MeshSTAConfigEntry 27 }

dot11MeshSTAMissingAckRetryLimit OBJECT-TYPE
    SYNTAX Unsigned32 (0..100)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute specifies the number of times the mesh STA may retry a
        frame for which it does not receive an ACK for a STA in power save mode
        after the mesh STA does not receive an ACK to a directed MPDU sent with
        the EOSP set to 1."
    ::= { dot11MeshSTAConfigEntry 28 }

dot11MeshAwakeWindowDuration OBJECT-TYPE
    SYNTAX Unsigned32 (0..65535)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute specifies the duration of the mesh Awake Window in TUs.
        This value is reflected in the value of the Mesh Awake Window element."
    ::= { dot11MeshSTAConfigEntry 29 }

dot11MCCAImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute specifies whether the MCCA is implemented in this station."
    ::= { dot11MeshSTAConfigEntry 30 }

dot11MCCAActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute specifies whether the station is MCCA enabled."
    DEFVAL { false }
    ::= { dot11MeshSTAConfigEntry 31 }

dot11MAFlimit OBJECT-TYPE
```

```
    SYNTAX Unsigned32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute specifies the maximum MCCA access fraction allowed at the
        mesh STA. This number expresses a multiple of (1/255)."
    DEFVAL { 128 }
    ::= { dot11MeshSTAConfigEntry 32 }

dot11MCCAScanDuration OBJECT-TYPE
    SYNTAX Unsigned32 (1..65535)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute specifies the duration in TUs after the activation of MCCA
        that the mesh STA shall not initiate or accept MCCAOP Setup Requests."
    DEFVAL { 3200 } -- (2 ** 5) * 100
    ::= { dot11MeshSTAConfigEntry 33 }

dot11MCCAAdvertPeriodMax OBJECT-TYPE
    SYNTAX Unsigned32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute specifies the maximum interval that a mesh STA with
        dot11MCCAActivated equal to true waits for an MCCAOP advertisement. It is
        expressed in number of DTIM intervals."
    DEFVAL { 1 }
    ::= { dot11MeshSTAConfigEntry 34 }


dot11MCCAMinTrackStates OBJECT-TYPE
    SYNTAX Unsigned32 (83..65535)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute specifies the smallest number of MCCAOP reservations that
        the MAC entity is able to track."
DEFVAL { 83 }
::= { dot11MeshSTAConfigEntry 35 }


dot11MCCAMaxTrackStates OBJECT-TYPE
    SYNTAX Unsigned32 (83..65535)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.
        The lower bound is given by the current value of dot11MCCAMinTrackStates.

        This attribute specifies the maximum number of MCCAOP reservations that
        the MAC entity is able to track."
    DEFVAL { 83 }
    ::= { dot11MeshSTAConfigEntry 36 }

dot11MCCAOPtimeout OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute specifies the timeout value for an MCCAOP teardown. It is
        expressed in TU."
    DEFVAL { 10000 }
    ::= { dot11MeshSTAConfigEntry 37 }

dot11MCCACWmin OBJECT-TYPE
    SYNTAX Unsigned32 (0..65535)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute specifies the value of the minimum size of the window that
        shall be used by the mesh STA during the MCCAOP for which it is the MCCAOP
        owner for generating a random number for the backoff."
    DEFVAL { 0 }
    ::= { dot11MeshSTAConfigEntry 38 }

dot11MCCACWmax OBJECT-TYPE
    SYNTAX Unsigned32 (0..65535)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute specifies the value of the maximum size of the window that
        shall be used by the mesh STA during the MCCAOP for which it is the MCCAOP
        owner for generating a random number for the backoff. The value of this
        attribute shall be such that it could always be expressed in the form of
        2**X - 1, where X is an integer."
    DEFVAL { 31 }
    ::= { dot11MeshSTAConfigEntry 39 }

dot11MCCAAIFSN OBJECT-TYPE
    SYNTAX Unsigned32 (0..15)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.
        This attribute specifies the number of slots, after a SIFS duration, that
        the mesh STA shall sense the medium idle either before transmitting or
        executing a backoff during an MCCAOP for which it is the MCCAOP owner."
    DEFVAL { 1 }
    ::= { dot11MeshSTAConfigEntry 40 }

-- **********************************************************************
-- * End of dot11MeshSTAConfig TABLE
-- **********************************************************************

-- **********************************************************************
-- * dot11MeshHWMPConfig TABLE
-- **********************************************************************

dot11MeshHWMPConfigTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11MeshHWMPConfigEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Mesh Station HWMP Configuration attributes. In tabular form to allow for
        multiple instances on an agent."
    ::= { dot11smt 24 }

dot11MeshHWMPConfigEntry  OBJECT-TYPE
    SYNTAX Dot11MeshHWMPConfigEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry in the dot11MeshHWMPConfigTable. It is possible for there to be
        multiple IEEE 802.11 interfaces on one agent, each with its unique MAC
        address. The relationship between an IEEE 802.11 interface and an inter-
        face in the context of the Internet-standard MIB is one-to-one. As such,
        the value of an ifIndex object instance can be directly used to identify
        corresponding instances of the objects defined herein.
        ifIndex - Each IEEE 802.11 interface is represented by an ifEntry. Inter-
        face tables in this MIB module are indexed by ifIndex."
    INDEX { ifIndex }
    ::= { dot11MeshHWMPConfigTable 1 }

Dot11MeshHWMPConfigEntry ::=
    SEQUENCE {
        dot11MeshHWMPmaxPREQretries                            Unsigned32,
        dot11MeshHWMPnetDiameter                               Unsigned32,
        dot11MeshHWMPnetDiameterTraversalTime                  Unsigned32,
        dot11MeshHWMPpreqMinInterval                           Unsigned32,
        dot11MeshHWMPperrMinInterval                           Unsigned32,
        dot11MeshHWMPactivePathToRootTimeout                   Unsigned32,
        dot11MeshHWMPactivePathTimeout                         Unsigned32,
        dot11MeshHWMProotMode                                  INTEGER,
        dot11MeshHWMProotInterval                              Unsigned32,
        dot11MeshHWMPrannInterval                              Unsigned32,
        dot11MeshHWMPtargetOnly                                INTEGER,
        dot11MeshHWMPmaintenanceInterval                       Unsigned32,
        dot11MeshHWMPconfirmationInterval                      Unsigned32
        }

dot11MeshHWMPmaxPREQretries OBJECT-TYPE
    SYNTAX Unsigned32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.
```

```
      This attribute specifies the number of Action frames containing a PREQ
      that an originator mesh STA can send to a particular path target for a
      specific path discovery."
   DEFVAL { 3 }
   ::= { dot11MeshHWMPConfigEntry 1}

dot11MeshHWMPnetDiameter OBJECT-TYPE
   SYNTAX Unsigned32 (1..255)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute specifies the estimate of the maximum number of hops that
      it takes for an HWMP element to propagate across the mesh BSS."
   DEFVAL { 31 }
   ::= { dot11MeshHWMPConfigEntry 2}

dot11MeshHWMPnetDiameterTraversalTime OBJECT-TYPE
   SYNTAX Unsigned32 (1..65535)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute specifies the estimate of the interval of time (in TUs)
      that it takes for an HWMP element to propagate across the mesh BSS."
   DEFVAL { 500 }
   ::= { dot11MeshHWMPConfigEntry 3}

dot11MeshHWMPpreqMinInterval OBJECT-TYPE
   SYNTAX Unsigned32 (1..65535)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute specifies the minimum interval of time (in TUs) during
      which a mesh STA can send only one Action frame containing a PREQ ele-
      ment."
   DEFVAL { 100 }
   ::= { dot11MeshHWMPConfigEntry 4}

dot11MeshHWMPperrMinInterval OBJECT-TYPE
   SYNTAX Unsigned32 (1..65535)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute specifies the minimum interval of time (in TUs) during
      which a mesh STA can send only one Action frame containing a PERR ele-
      ment."
   DEFVAL { 100 }
   ::= { dot11MeshHWMPConfigEntry 5}
```

```
dot11MeshHWMPactivePathToRootTimeout OBJECT-TYPE
   SYNTAX Unsigned32 (1..65535)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This object shall specify the time (in TUs) for which mesh STAs receiving
      a proactive PREQ shall consider the forwarding information to the root
      mesh STA to be valid; it needs to be greater than
      dot11MeshHWMProotInterval."
   DEFVAL { 5000 }
   ::= { dot11MeshHWMPConfigEntry 6}

dot11MeshHWMPactivePathTimeout OBJECT-TYPE
   SYNTAX Unsigned32 (1..65535)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute specifies the time (in TUs) for which mesh STAs receiving a
      PREQ to individual target(s) shall consider the forwarding information to
      be valid."
   DEFVAL { 5000 }
   ::= { dot11MeshHWMPConfigEntry 7}

dot11MeshHWMProotMode OBJECT-TYPE
   SYNTAX INTEGER {
      noRoot(0),
      proactivePREQnoPREP(2),
      proactivePREQwithPREP(3),
      rann(4) }
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute controls the configuration of a mesh STA as root mesh STA.
      A mesh STA is configured as a root mesh STA if dot11MeshHWMProotMode is
      set to 2, 3 or 4. Different values correspond to different modes of the
      root mesh STA. The mesh STA is not a root mesh STA when the attribute is
      set to 0."
   DEFVAL { noRoot }
   ::= { dot11MeshHWMPConfigEntry 8}

dot11MeshHWMProotInterval OBJECT-TYPE
   SYNTAX Unsigned32 (1..65535)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute specifies the minimum interval of time (in TUs) during
      which a root mesh STA can send only one Action frame containing a proac-
```

```
        tive PREQ element."
    DEFVAL { 2000 }
    ::= { dot11MeshHWMPConfigEntry 9}

dot11MeshHWMPrannInterval OBJECT-TYPE
    SYNTAX Unsigned32 (1..65535)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute specifies the minimum interval of time (in TUs) during
        which a mesh STA can send only one Action frame containing a RANN ele-
        ment."
    DEFVAL { 2000 }
    ::= { dot11MeshHWMPConfigEntry 10}

dot11MeshHWMPtargetOnly OBJECT-TYPE
    SYNTAX INTEGER { intermediateMSTA(0), targetOnly(1) }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute, when set to intermediateMSTA (0), allows intermediate mesh
        STAs to respond with a PREP to a PREQ if they have valid forwarding infor-
        mation to the requested target. When set to targetOnly (1), only the tar-
        get mesh STA is allowed to respond with a PREP to a PREQ."
    DEFVAL { targetOnly }
    ::= { dot11MeshHWMPConfigEntry 11}

dot11MeshHWMPmaintenanceInterval OBJECT-TYPE
    SYNTAX Unsigned32 (1..65535)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute specifies the minimum interval of time (in TUs) during
        which a mesh STA can send only one Action frame containing a PREQ element
        for path maintenance."
    DEFVAL { 2000 }
    ::= { dot11MeshHWMPConfigEntry 12}

dot11MeshHWMPconfirmationInterval OBJECT-TYPE
    SYNTAX Unsigned32 (1..65535)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute specifies the minimum interval of time (in TUs) during
        which a mesh STA can send only one Action frame containing a PREQ element
        for root path confirmation."
    DEFVAL { 2000 }
    ::= { dot11MeshHWMPConfigEntry 13}
```

```
-- **********************************************************************
-- * End of dot11MeshHWMPConfig TABLE
-- **********************************************************************

-- ********************************************************************
-- * dot11RSNAConfigPasswordValue TABLE
-- ********************************************************************

dot11RSNAConfigPasswordValueTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11RSNAConfigPasswordValueEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "When SAE authentication is the selected AKM suite,
        this table is used to locate the binary representation
        of a shared, secret, and potentially low-entropy word,
        phrase, code, or key that will be used as the
        authentication credential between a TA/RA pair.

        This table is logically write-only. Reading this table
        returns unsuccessful status or null or zero."
    ::= { dot11smt 25 }

dot11RSNAConfigPasswordValueEntry OBJECT-TYPE
    SYNTAX Dot11RSNAConfigPasswordValueEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry (conceptual row) in the Password Value Table"
    INDEX { dot11RSNAConfigPasswordValueIndex }
    ::= { dot11RSNAConfigPasswordValueTable 1 }

Dot11RSNAConfigPasswordValueEntry ::=
    SEQUENCE {
        dot11RSNAConfigPasswordValueIndex                     Unsigned32,
        dot11RSNAConfigPasswordCredential                     OCTET STRING,
        dot11RSNAConfigPasswordPeerMac                        MacAddress }

dot11RSNAConfigPasswordValueIndex OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "The auxiliary variable used to identify instances of the columnar
        objects in the Password Value table."
    ::= { dot11RSNAConfigPasswordValueEntry 1 }

dot11RSNAConfigPasswordCredential OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This variable is a binary representation of a shared,
        secret, and potentially low-entropy word, phrase, code
        or key used as an authentication credential.

        Any character-based word or phrase shall be converted
        into a canonical binary representation according to
        11.3.3 before populating the Password Credential."
    ::= { dot11RSNAConfigPasswordValueEntry 2 }

dot11RSNAConfigPasswordPeerMac OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This variable represents the MAC address of the peer
        that is to be authenticated. A wildcard BSSID is
        permitted when passwords are shared among peers."
    ::= { dot11RSNAConfigPasswordValueEntry 3 }

-- ********************************************************************
-- * End of dot11RSNAConfigPasswordValue TABLE
-- ********************************************************************

-- ********************************************************************
-- * dot11RSNAConfigDLCGroup TABLE
-- ********************************************************************

dot11RSNAConfigDLCGroupTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11RSNAConfigDLCGroupEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "This table gives a prioritized list of domain parameter set
        Identifiers for discrete logarithm cryptography (DLC) groups."
    ::= { dot11smt 26 }

dot11RSNAConfigDLCGroupEntry OBJECT-TYPE
    SYNTAX Dot11RSNAConfigDLCGroupEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry (conceptual row) in the DLC Group Table."
    INDEX { dot11RSNAConfigDLCGroupIndex }
    ::= { dot11RSNAConfigDLCGroupTable 1 }

Dot11RSNAConfigDLCGroupEntry ::=
    SEQUENCE {
        dot11RSNAConfigDLCGroupIndex                        Unsigned32,
        dot11RSNAConfigDLCGroupIdentifier                   Unsigned32 }

dot11RSNAConfigDLCGroupIndex OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "The variable used to identify instances of the columnar
        objects in the DLC Group Table. Entries are sorted
        based on the Group Index according to the priority
        of the Group Identifier relative to other objects.

        More preferred Group Identifiers will have a lower
        index in the Group Entry."
    ::= { dot11RSNAConfigDLCGroupEntry 1 }

dot11RSNAConfigDLCGroupIdentifier OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
```

```
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This variable uniquely identifies a domain parameter
      set for a group in the IANA registry `Group Description'
      attributes for RFC 2409 (IKE)."
   ::= { dot11RSNAConfigDLCGroupEntry 2 }

-- **********************************************************************
-- *    End of dot11RSNAConfigDLCGroup TABLE
-- **********************************************************************

-- **********************************************************************
-- * MAC Attribute Templates
-- **********************************************************************

-- **********************************************************************
-- * dot11Operation TABLE
-- **********************************************************************

dot11OperationTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11OperationEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "Group contains MAC attributes pertaining to the operation of the MAC.
      This has been implemented as a table in order to allow for multiple
      instantiations on an agent."
   ::= { dot11mac 1 }

dot11OperationEntry OBJECT-TYPE
   SYNTAX Dot11OperationEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "An entry in the dot11OperationEntry Table.

      ifIndex - Each IEEE 802.11 interface is represented by an ifEntry. Inter-
      face tables in this MIB module are indexed by ifIndex."
   INDEX { ifIndex }
   ::= { dot11OperationTable 1 }

Dot11OperationEntry ::=
   SEQUENCE {
      dot11MACAddress                                       MacAddress,
      dot11RTSThreshold                                     Unsigned32,
      dot11ShortRetryLimit                                  Unsigned32,
      dot11LongRetryLimit                                   Unsigned32,
      dot11FragmentationThreshold                           Unsigned32,
      dot11MaxTransmitMSDULifetime                          Unsigned32,
      dot11MaxReceiveLifetime                               Unsigned32,
      dot11ManufacturerID                                   DisplayString,
      dot11ProductID                                        DisplayString,
      dot11CAPLimit                                         Unsigned32,
      dot11HCCWmin                                          Unsigned32,
      dot11HCCWmax                                          Unsigned32,
      dot11HCCAIFSN                                         Unsigned32,
      dot11ADDBAResponseTimeout                             Unsigned32,
      dot11ADDTSResponseTimeout                             Unsigned32,
      dot11ChannelUtilizationBeaconInterval                 Unsigned32,
      dot11ScheduleTimeout                                  Unsigned32,
        dot11DLSResponseTimeout                              Unsigned32,
        dot11QAPMissingAckRetryLimit                         Unsigned32,
        dot11EDCAAveragingPeriod                             Unsigned32,
        dot11HTProtection                                    INTEGER,
        dot11RIFSMode                                        TruthValue,
        dot11PSMPControlledAccess                            TruthValue,
        dot11ServiceIntervalGranularity                      Unsigned32,
        dot11DualCTSProtection                               TruthValue,
        dot11LSIGTXOPFullProtectionActivated                 TruthValue,
        dot11NonGFEntitiesPresent                            TruthValue,
        dot11PCOActivated                                    TruthValue,
        dot11PCOFortyMaxDuration                             Unsigned32,
        dot11PCOTwentyMaxDuration                            Unsigned32,
        dot11PCOFortyMinDuration                             Unsigned32,
        dot11PCOTwentyMinDuration                            Unsigned32,
        dot11FortyMHzIntolerant                              TruthValue,
        dot11BSSWidthTriggerScanInterval                     Unsigned32,
        dot11BSSWidthChannelTransitionDelayFactor            Unsigned32,
        dot11OBSSScanPassiveDwell                            Unsigned32,
        dot11OBSSScanActiveDwell                             Unsigned32,
        dot11OBSSScanPassiveTotalPerChannel                  Unsigned32,
        dot11OBSSScanActiveTotalPerChannel                   Unsigned32,
        dot112040BSSCoexistenceManagementSupport             TruthValue,
        dot11OBSSScanActivityThreshold                       Unsigned32 }

dot11MACAddress OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        Unique MAC Address assigned to the STA."
    ::= { dot11OperationEntry 1 }

dot11RTSThreshold OBJECT-TYPE
    SYNTAX Unsigned32 (0..65536)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute indicates the number of octets in a PSDU, below which an
        RTS/CTS handshake is not performed, except as RTS/CTS is used as a cross
        modulation protection mechanism as defined in 9.23. An RTS/CTS handshake
        is performed at the beginning of any frame exchange sequence where the
        PSDU is of type Data or Management, the PSDU has an individual address in
        the Address1 field, and the length of the PSDU is greater than this
        threshold. Setting this attribute to be larger than the maximum PSDU size
        has the effect of turning off the RTS/CTS handshake for frames of Data or
        Management type transmitted by this STA. Setting this attribute to 0 has
        the effect of turning on the RTS/CTS handshake for all frames of Data or
        Management type transmitted by this STA."
    DEFVAL { 65536 }
    ::= { dot11OperationEntry 2 }

dot11ShortRetryLimit OBJECT-TYPE
    SYNTAX Unsigned32 (1..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
```

```
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute indicates the maximum number of transmission attempts of a
        frame, the length of which is less than or equal to dot11RTSThreshold,
        that is made before a failure condition is indicated."
    DEFVAL { 7 }
    ::= { dot11OperationEntry 3 }

dot11LongRetryLimit OBJECT-TYPE
    SYNTAX Unsigned32 (1..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute indicates the maximum number of transmission attempts of a
        frame, the length of which is greater than dot11RTSThreshold, that is made
        before a failure condition is indicated."
    DEFVAL { 4 }
    ::= { dot11OperationEntry 4 }

dot11FragmentationThreshold OBJECT-TYPE
    SYNTAX Unsigned32 (256..8000)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute specifies the current maximum size, in octets, of the MPDU
        that may be delivered to the security encapsulation. This maximum size
        does not apply when an MSDU is transmitted using an HT-immediate or HT-
        delayed Block Ack agreement, or when an MSDU or MMPDU is carried in an A-
        MPDU. Fields added to the frame by security encapsulation are not counted
        against the limit specified by this attribute. Except as described above,
        an MSDU or MMPDU is fragmented when the resulting frame has an individual
        address in the Address1 field, and the length of the frame is larger than
        this threshold, excluding security encapsulation fields. The default value
        for this attribute is the lesser of 8000 or the aMPDUMaxLength or the aPS-
        DUMaxLength of the attached PHY and the value never exceeds the lesser of
        8000 or the aMPDUMaxLength or the aPSDUMaxLength of the attached PHY."
    ::= { dot11OperationEntry 5 }

dot11MaxTransmitMSDULifetime OBJECT-TYPE
    SYNTAX Unsigned32 (1..4294967295)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        The MaxTransmitMSDULifetime is the elapsed time in TU, after the initial
        transmission of an MSDU, after which further attempts to transmit the MSDU
        are terminated."
    DEFVAL { 512 }
    ::= { dot11OperationEntry 6 }

dot11MaxReceiveLifetime OBJECT-TYPE
```

```
    SYNTAX Unsigned32 (1..4294967295)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        The MaxReceiveLifetime is the elapsed time in TU, after the initial recep-
        tion of a fragmented MMPDU or MSDU, after which further attempts to reas-
        semble the MMPDU or MSDU are terminated."
    DEFVAL { 512 }
    ::= { dot11OperationEntry 7 }

dot11ManufacturerID OBJECT-TYPE
    SYNTAX DisplayString (SIZE(0..128))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        The ManufacturerID includes, at a minimum, the name of the manufacturer.
        It may include additional information at the manufacturer's discretion.
        The default value of this attribute is null."
    ::= { dot11OperationEntry 8 }

dot11ProductID OBJECT-TYPE
    SYNTAX DisplayString (SIZE(0..128))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        The ProductID includes, at a minimum, an identifier that is unique to the
        manufacturer. It may include additional information at the manufacturer's
        discretion. The default value of this attribute is null."
    ::= { dot11OperationEntry 9 }

dot11CAPLimit OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute specifies the maximum number of TUs a Controlled access
        phase(CAP) can last."
    ::= { dot11OperationEntry 10 }

dot11HCCWmin OBJECT-TYPE
    SYNTAX Unsigned32 -- (0..aCWmin)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute specifies the value of the minimum size of the window that
        is used by the HC for generating a random number for the backoff. The
```

```
       value of this attribute is such that it could always be expressed in the
       form of 2**X - 1, where X is an integer."
    DEFVAL { 0 }
    ::= { dot11OperationEntry 11 }

dot11HCCWmax OBJECT-TYPE
    SYNTAX Unsigned32 -- (0..aCWmax)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
       "This is a control variable.
       It is written by an external management entity.
       Changes take effect as soon as practical in the implementation.

       This attribute specifies the value of the maximum size of the window that
       is used by the HC for generating a random number for the backoff. The
       value of this attribute is such that it could always be expressed in the
       form of 2**X - 1, where X is an integer."
    DEFVAL { 0 }
    ::= { dot11OperationEntry 12 }

dot11HCCAIFSN OBJECT-TYPE
    SYNTAX Unsigned32 (1..15)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
       "This is a control variable.
       It is written by an external management entity.
       Changes take effect as soon as practical in the implementation.

       This attribute specifies the number of slots, after a SIFS duration, that
       the HC senses the medium idle either before transmitting or executing a
       backoff."
    DEFVAL { 1 }
    ::= { dot11OperationEntry 13 }

dot11ADDBAResponseTimeout OBJECT-TYPE
    SYNTAX Unsigned32 (1..65535)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
       "This is a control variable.
       It is written by an external management entity.
       Changes take effect as soon as practical in the implementation.

       This attribute specifies the timeout in seconds for an ADDBA Response
       frame that is a response to an ADDBA Request frame."
    DEFVAL { 1 }
    ::= { dot11OperationEntry 14 }

dot11ADDTSResponseTimeout OBJECT-TYPE
    SYNTAX Unsigned32 (1..65535)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
       "This is a control variable.
       It is written by an external management entity.
       Changes take effect as soon as practical in the implementation.

       This attribute specifies the maximum number of seconds an ADDTS request is
       to be responded."
    DEFVAL { 1 }
    ::= { dot11OperationEntry 15 }
```

```
dot11ChannelUtilizationBeaconInterval OBJECT-TYPE
   SYNTAX Unsigned32 (1..100)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute indicates the number of beacon intervals over which the
      channel busy time should be averaged."
   DEFVAL { 50 }
   ::= { dot11OperationEntry 16 }

dot11ScheduleTimeout OBJECT-TYPE
   SYNTAX Unsigned32 (1..100)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute indicates the duration in TUs after which a STA could go
      into power save mode."
   DEFVAL { 10 }
   ::= { dot11OperationEntry 17 }

dot11DLSResponseTimeout OBJECT-TYPE
   SYNTAX Unsigned32 (1..65535)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute specifies the maximum number of seconds a direct link
      request is to be responded."
   DEFVAL { 10 }
   ::= { dot11OperationEntry 18 }

dot11QAPMissingAckRetryLimit OBJECT-TYPE
   SYNTAX Unsigned32 (1..100)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute indicates the number of times the AP may retry a frame for
      which it does not receive an ACK for a STA in power save mode after
      receiving a PS-Poll and sending an individually addressed response or
      after the AP does not receive an ACK to a directed MPDU sent with the EOSP
      equal to 1."
   ::= { dot11OperationEntry 19 }

dot11EDCAAveragingPeriod OBJECT-TYPE
   SYNTAX Unsigned32 (1..100)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute indicates the number of seconds over which the
      admitted_time and used_time are computed."
   DEFVAL { 5 }
   ::= { dot11OperationEntry 20 }

dot11HTProtection OBJECT-TYPE
   SYNTAX INTEGER {
      htNoProtection (0),
      htNonmemberProtection(1),
      ht20MHzProtection(2),
      htNonHTmixed(3) }
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute indicates the level of protection that needs to be provided
      to the transmissions in an IEEE 802.11 network with HT STAs."
   DEFVAL { htNoProtection }
   ::= { dot11OperationEntry 21 }

dot11RIFSMode OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute, when true, indicates that use of RIFS is allowed in the
      BSS."
   DEFVAL { false }
   ::= { dot11OperationEntry 22 }

dot11PSMPControlledAccess OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute, when true indicates that the AP accepts associations only
      from STAs for which dot11PSMPOptionImplemented is true."
   DEFVAL { false }
   ::= { dot11OperationEntry 23 }

dot11ServiceIntervalGranularity OBJECT-TYPE
   SYNTAX Unsigned32 (0..7)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute indicates the SI granularity to be used for scheduled PSMP.
      The value of the granularity is given by
      (dot11ServiceIntervalGranularity+1)*5 ms."
   DEFVAL { 0 }
   ::= { dot11OperationEntry 24 }

dot11DualCTSProtection OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute, when true indicates that the AP uses dual CTS protection
      to protect the non-STBC frame and STBC frame transmissions."
   DEFVAL { false }
   ::= { dot11OperationEntry 25 }

dot11LSIGTXOPFullProtectionActivated OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute, when true, indicates that the LSIG TXOP protection may be
      used by STAs that have the attribute
      dot11LSigTxopProtectionOptionImplemented equal to true."
   DEFVAL { false }
   ::= { dot11OperationEntry 26 }

dot11NonGFEntitiesPresent OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME when it determines that the presence of green-
      field stations in the BSS has changed.

      This attribute, when true, indicates that STA that are not HT-greenfield
      Capable are present in the BSS."
   DEFVAL { false }
   ::= { dot11OperationEntry 27 }

dot11PCOActivated OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute, when true, indicates that the PCO is activated."
   DEFVAL { false }
   ::= { dot11OperationEntry 28 }

dot11PCOFortyMaxDuration OBJECT-TYPE
   SYNTAX Unsigned32 (1..200)
   MAX-ACCESS read-write
```

```
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        The attribute indicates the maximum duration of 40 MHz phase in TU under
        PCO operation. The value of this attribute shall be equal to or larger
        than dot11PCOFortyMinDuration."
    DEFVAL { 30 }
    ::= { dot11OperationEntry 29 }

dot11PCOTwentyMaxDuration OBJECT-TYPE
    SYNTAX Unsigned32 (1..200)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        The attribute indicates the maximum duration of 20 MHz phase in TU under
        PCO operation. The value of this attribute shall be equal to or larger
        than dot11PCOTwentyMinDuration."
    DEFVAL { 30 }
    ::= { dot11OperationEntry 30 }

dot11PCOFortyMinDuration OBJECT-TYPE
    SYNTAX Unsigned32 (1..200)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        The attribute indicates the minimum duration of 40 MHz phase in TU under
        PCO operation."
    DEFVAL { 20 }
    ::= { dot11OperationEntry 31 }

dot11PCOTwentyMinDuration OBJECT-TYPE
    SYNTAX Unsigned32 (1..200)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        The attribute indicates the minimum duration of 20 MHz phase in TU under
        PCO operation."
    DEFVAL { 20 }
    ::= { dot11OperationEntry 32 }

dot11FortyMHzIntolerant OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.
```

```
      This attribute, when true, indicates that the STA requests that 40 MHz
      mask PPDUs are not transmitted within range of the STA."
   DEFVAL { false }
   ::= { dot11OperationEntry 33 }

dot11BSSWidthTriggerScanInterval OBJECT-TYPE
   SYNTAX Unsigned32 (10..900)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute indicates the maximum interval in seconds between scan
      operations to be performed to detect BSS channel width trigger events."
   DEFVAL { 300 }
   ::= { dot11OperationEntry 34 }

dot11BSSWidthChannelTransitionDelayFactor OBJECT-TYPE
   SYNTAX Unsigned32 (5..100)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute indicates the minimum ratio between the delay time in per-
      forming a switch from 20 MHz BSS operation to 20/40 MHz BSS operation and
      the maximum interval between OBSS scan operations."
   DEFVAL { 5 }
   ::= { dot11OperationEntry 35 }

dot11OBSSScanPassiveDwell OBJECT-TYPE
   SYNTAX Unsigned32 (5..1000)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute indicates the minimum amount of time in TU that the STA
      continuously scans each channel when performing a passive OBSS scan oper-
      ation."
   DEFVAL { 20 }
   ::= { dot11OperationEntry 36 }

dot11OBSSScanActiveDwell OBJECT-TYPE
   SYNTAX Unsigned32 (10..1000)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute indicates the minimum amount of time in TU that the STA
      continuously scans each channel when performing an active OBSS scan oper-
      ation."
   DEFVAL { 10 }
   ::= { dot11OperationEntry 37 }
```

```
dot11OBSSScanPassiveTotalPerChannel OBJECT-TYPE
   SYNTAX Unsigned32 (200..10000)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute indicates the minimum total amount of time in TU that the
      STA scans each channel when performing a passive OBSS scan operation."
   DEFVAL { 200 }
   ::= { dot11OperationEntry 38 }

dot11OBSSScanActiveTotalPerChannel OBJECT-TYPE
   SYNTAX Unsigned32 (20..10000)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute indicates the minimum total amount of time in TU that the
      STA scans each channel when performing an active OBSS scan operation."
   DEFVAL { 20 }
   ::= { dot11OperationEntry 39 }

dot112040BSSCoexistenceManagementSupport OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      This attribute, when true, indicates that the STA supports the transmis-
      sion and reception of the 20/40 BSS Coexistence Management frame."
   DEFVAL { false }
   ::= { dot11OperationEntry 40 }

dot11OBSSScanActivityThreshold OBJECT-TYPE
   SYNTAX Unsigned32 (0..100)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute indicates in hundredths of percent, the maximum total time
      that a STA may be active on the medium during a period of
      dot11BSSWidthChannelTransitionDelayFactor *
      dot11BSSWidthTriggerScanInterval seconds without being obligated to per-
      form OBSS Scan operations. The default value of this attribute is 25,
      which equates to 0.25%."
   DEFVAL { 25 }
   ::= { dot11OperationEntry 41}

-- **********************************************************************
-- *    End of dot11Operation TABLE
-- **********************************************************************

-- **********************************************************************
-- *    dot11Counters TABLE
-- **********************************************************************

dot11CountersTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11CountersEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "Group containing attributes that are MAC counters. Implemented as a table
      to allow for multiple instantiations on an agent."
   ::= { dot11mac 2 }

dot11CountersEntry OBJECT-TYPE
   SYNTAX Dot11CountersEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "An entry in the dot11CountersEntry Table.

      ifIndex - Each IEEE 802.11 interface is represented by an ifEntry. Inter-
      face tables in this MIB module are indexed by ifIndex."
   INDEX { ifIndex }
   ::= { dot11CountersTable 1 }

Dot11CountersEntry ::=
   SEQUENCE {
      dot11TransmittedFragmentCount                    Counter32,
      dot11GroupTransmittedFrameCount                  Counter32,
      dot11FailedCount                                 Counter32,
      dot11RetryCount                                  Counter32,
      dot11MultipleRetryCount                          Counter32,
      dot11FrameDuplicateCount                         Counter32,
      dot11RTSSuccessCount                             Counter32,
      dot11RTSFailureCount                             Counter32,
      dot11ACKFailureCount                             Counter32,
      dot11ReceivedFragmentCount                       Counter32,
      dot11GroupReceivedFrameCount                     Counter32,
      dot11FCSErrorCount                               Counter32,
      dot11TransmittedFrameCount                       Counter32,
      dot11WEPUndecryptableCount                       Counter32,
      dot11QosDiscardedFragmentCount                   Counter32,
      dot11AssociatedStationCount                      Counter32,
      dot11QosCFPollsReceivedCount                     Counter32,
      dot11QosCFPollsUnusedCount                       Counter32,
      dot11QosCFPollsUnusableCount                     Counter32,
      dot11QosCFPollsLostCount                         Counter32,
      dot11TransmittedAMSDUCount                       Counter32,
      dot11FailedAMSDUCount                            Counter32,
      dot11RetryAMSDUCount                             Counter32,
      dot11MultipleRetryAMSDUCount                     Counter32,
      dot11TransmittedOctetsInAMSDUCount               Counter64,
      dot11AMSDUAckFailureCount                        Counter32,
      dot11ReceivedAMSDUCount                          Counter32,
      dot11ReceivedOctetsInAMSDUCount                  Counter64,
      dot11TransmittedAMPDUCount                       Counter32,
      dot11TransmittedMPDUsInAMPDUCount                Counter32,
      dot11TransmittedOctetsInAMPDUCount               Counter64,
      dot11AMPDUReceivedCount                          Counter32,
      dot11MPDUInReceivedAMPDUCount                    Counter32,
      dot11ReceivedOctetsInAMPDUCount                  Counter64,
      dot11AMPDUDelimiterCRCErrorCount                 Counter32,
      dot11ImplicitBARFailureCount                     Counter32,
      dot11ExplicitBARFailureCount                     Counter32,
      dot11ChannelWidthSwitchCount                     Counter32,
        dot11TwentyMHzFrameTransmittedCount               Counter32,
        dot11FortyMHzFrameTransmittedCount                Counter32,
        dot11TwentyMHzFrameReceivedCount                  Counter32,
        dot11FortyMHzFrameReceivedCount                   Counter32,
        dot11PSMPUTTGrantDuration                         Counter32,
        dot11PSMPUTTUsedDuration                          Counter32,
        dot11GrantedRDGUsedCount                          Counter32,
        dot11GrantedRDGUnusedCount                        Counter32,
        dot11TransmittedFramesInGrantedRDGCount           Counter32,
        dot11TransmittedOctetsInGrantedRDGCount           Counter64,
        dot11BeamformingFrameCount                        Counter32,
        dot11DualCTSSuccessCount                          Counter32,
        dot11DualCTSFailureCount                          Counter32,
        dot11STBCCTSSuccessCount                          Counter32,
        dot11STBCCTSFailureCount                          Counter32,
        dot11nonSTBCCTSSuccessCount                       Counter32,
        dot11nonSTBCCTSFailureCount                       Counter32,
        dot11RTSLSIGSuccessCount                          Counter32,
        dot11RTSLSIGFailureCount                          Counter32,
        dot11PBACErrors                                   Counter32,
        dot11DeniedAssociationCounterDueToBSSLoad         Counter32
    }

dot11TransmittedFragmentCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when a fragment is successfully transmitted.

        This counter is incremented for an acknowledged MPDU with an individual
        address in the address 1 field or an MPDU with a group address in the
        address 1 field of type Data or Management."
    ::= { dot11CountersEntry 1 }

dot11GroupTransmittedFrameCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when a group addressed frame is transmitted.

        This counter is incremented only when the group bit is set in the destina-
        tion MAC address of a successfully transmitted MSDU. When operating as a
        STA in an ESS, where these frames are directed to the AP, this implies
        having received an acknowledgment to all associated MPDUs."
    ::= { dot11CountersEntry 2 }

dot11FailedCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when the condition described below occurs.

        This counter increments when an MSDU is not transmitted successfully due
        to the number of transmit attempts exceeding either the
        dot11ShortRetryLimit or dot11LongRetryLimit."
    ::= { dot11CountersEntry 3 }

dot11RetryCount OBJECT-TYPE
```

```
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the MAC when the condition described below occurs.

      This counter increments when an MSDU is successfully transmitted after one
      or more retransmissions."
   ::= { dot11CountersEntry 4 }

dot11MultipleRetryCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the MAC when the condition described below occurs.

      This counter increments when an MSDU is successfully transmitted after
      more than one retransmission."
   ::= { dot11CountersEntry 5 }

dot11FrameDuplicateCount  OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the MAC when the condition described below occurs.

      This counter increments when a frame is received that the Sequence Control
      field indicates is a duplicate."
   ::= { dot11CountersEntry 6 }

dot11RTSSuccessCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the MAC when a CTS is received in response to an RTS.

      This counter increments when a CTS is received in response to an RTS."
   ::= {  dot11CountersEntry 7 }

dot11RTSFailureCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the MAC when the condition described below occurs.

      This counter increments when a CTS is not received in response to an RTS."
   ::= { dot11CountersEntry 8 }

dot11ACKFailureCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the MAC when the condition described below occurs.
      This counter increments when an ACK is not received when expected."
   ::= {  dot11CountersEntry 9 }

dot11ReceivedFragmentCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the MAC when a fragment is successfully received.

      This counter is incremented for each successfully received MPDU of type
      Data or Management."
   ::= { dot11CountersEntry 10 }

dot11GroupReceivedFrameCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the MAC when a group addressed frame is received.

      This counter increments when a MSDU is received with the group bit set in
      the destination MAC address."
   ::= { dot11CountersEntry 11 }

dot11FCSErrorCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the MAC when the condition described below occurs.

      This counter increments when an FCS error is detected in a received MPDU."
   ::= { dot11CountersEntry 12 }

dot11TransmittedFrameCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the MAC when a frame is successfully transmitted.

      This counter increments for each successfully transmitted MSDU."
   ::= { dot11CountersEntry 13 }

dot11WEPUndecryptableCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the MAC when the condition described below occurs.

      This counter increments when a frame is received with the Protected Frame
      subfield of the Frame Control field equal to one and the WEPOn value for
      the key mapped to the transmitter's MAC address indicates that the frame
      should not have been encrypted or that frame is discarded due to the
      receiving STA not implementing the privacy option."
   ::= { dot11CountersEntry 14 }
```

```
dot11QosDiscardedFragmentCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the MAC when the condition described below occurs.

      This counter increments for each QoS Data MPDU that has been discarded."
   ::= { dot11CountersEntry 15 }

dot11AssociatedStationCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the MAC when a STA associates or disassociates.

      This counter, only available at AP, increments when a station associates
      or reassociates. This counter decrements when a station disassociates."
   ::= { dot11CountersEntry 16 }

dot11QosCFPollsReceivedCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the MAC when a CF-Poll is received.

      This counter increments for each QoS (+)CF-Poll that has been received."
   ::= { dot11CountersEntry 17 }

dot11QosCFPollsUnusedCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the MAC when a CF-Poll is unused.

      This counter increments for each QoS (+)CF-Poll that has been received but
      not used."
   ::= { dot11CountersEntry 18 }

dot11QosCFPollsUnusableCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the MAC when the condition described below occurs.

      This counter increments for each QoS (+)CF-Poll that has been received but
      could not be used due to the TXOP size being smaller than the time that is
      required for one frame exchange sequence."
   ::= { dot11CountersEntry 19 }

dot11QosCFPollsLostCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
```

```
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when the condition described below occurs.

        This counter increments for each QoS (+)CF-Poll that has been issued where
        there was no response from the QoS STA."
    ::= { dot11CountersEntry 20 }

dot11TransmittedAMSDUCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when a transmitted A-MSDU is acknowledged.

        This counter shall be incremented for an acknowledged A-MSDU frame with an
        individual address in the address 1 field or an A-MSDU frame with a group
        address in the address 1 field."
    ::= { dot11CountersEntry 21 }

dot11FailedAMSDUCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when the condition described below occurs.

        This counter shall be incremented when an A-MSDU is not transmitted suc-
        cessfully due to the number of transmit attempts exceeding either the
        dot11ShortRetryLimit or dot11LongRetryLimit."
    ::= { dot11CountersEntry 22 }

dot11RetryAMSDUCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when a transmitted A-MSDU is acknowledged only
        after one or more retransmissions.

        This counter shall be incremented when an A-MSDU is successfully transmit-
        ted after one or more retransmissions."
    ::= { dot11CountersEntry 23 }

dot11MultipleRetryAMSDUCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when a transmitted A-MSDU is acknowledged only
        after more than one retransmission.

        This counter shall be incremented when an A-MSDU is successfully transmit-
        ted after more than one retransmission."
    ::= { dot11CountersEntry 24 }

dot11TransmittedOctetsInAMSDUCount OBJECT-TYPE
    SYNTAX Counter64
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when an A-MSDU is transmitted.

        This counter shall be incremented by the number of octets in the framebody
        of an A-MSDU frame when an A-MSDU frame is successfully transmitted."
    ::= { dot11CountersEntry 25 }

dot11AMSDUAckFailureCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when the condition described below occurs.

        This counter shall be incremented when an acknowledgment to an A-MSDU is
        not received when expected. This acknowledgment can be in an ACK or the
        BlockAck frame."
    ::= { dot11CountersEntry 26 }

dot11ReceivedAMSDUCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when an A-MSDU is received.

        This counter shall be incremented for a received A-MSDU frame with the
        station's MAC address in the address 1 field or an A-MSDU frame with a
        group address in the address 1 field."
    ::= { dot11CountersEntry 27 }

dot11ReceivedOctetsInAMSDUCount OBJECT-TYPE
    SYNTAX Counter64
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when an A-MSDU is received.

        This counter shall be incremented by the number of octets in the framebody
        of an A-MSDU frame when an A-MSDU frame is received."
    ::= { dot11CountersEntry 28 }

dot11TransmittedAMPDUCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when an A-MPDU is transmitted.

        This counter shall be incremented when an A-MPDU is transmitted."
    ::= { dot11CountersEntry 29 }

dot11TransmittedMPDUsInAMPDUCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when an A-MPDU is transmitted.
```

```
      This counter shall increment by the number of MPDUs in the A-MPDU when an
      A-MPDU is transmitted."
   ::= { dot11CountersEntry 30 }

dot11TransmittedOctetsInAMPDUCount OBJECT-TYPE
   SYNTAX Counter64
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the MAC when an A-MPDU is transmitted.

      This counter shall be incremented by the number of octets in the A-MPDU
      frame when an A-MPDU frame is transmitted."
   ::= { dot11CountersEntry 31 }

dot11AMPDUReceivedCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the MAC when an A-MPDU is received.

      This counter shall be incremented when the MAC receives an A-MPDU from the
      PHY."
   ::= { dot11CountersEntry 32 }

dot11MPDUInReceivedAMPDUCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the MAC when an A-MPDU is received.

      This counter shall be incremented by the number of MPDUs received in the
      A-MPDU when an A-MPDU is received."
   ::= { dot11CountersEntry 33 }

dot11ReceivedOctetsInAMPDUCount OBJECT-TYPE
   SYNTAX Counter64
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the MAC when an A-MPDU is received.

      This counter shall be incremented by the number of octets in the A-MPDU
      frame when an A-MPDU frame is received."
   ::= { dot11CountersEntry 34 }

dot11AMPDUDelimiterCRCErrorCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the MAC when the condition described below occurs.

      This counter shall be incremented when an MPDU delimiter has a CRC error
      when this is the first CRC error in the received A-MPDU or when the previ-
      ous delimiter has been decoded correctly."
```

```
    ::= { dot11CountersEntry 35 }

dot11ImplicitBARFailureCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when the condition described below occurs.

        This counter shall be incremented when the expected BlockAck is not
        received in response to an Implicit BlockAckReq frame."
    ::= { dot11CountersEntry 36 }

dot11ExplicitBARFailureCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when the condition described below occurs.

        This counter shall be incremented when the expected BlockAck is not
        received in response to an Explicit BlockAckReq."
    ::= { dot11CountersEntry 37 }

dot11ChannelWidthSwitchCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when the bandwidth is switched.

        This counter shall be increment when the bandwidth used is switched from
        20 to 40 or vice-versa."
    ::= { dot11CountersEntry 38 }

dot11TwentyMHzFrameTransmittedCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when a frame is transmitted only on the primary
        channel.

        This counter shall be incremented when a frame is transmitted only on the
        primary channel."
    ::= { dot11CountersEntry 39 }

dot11FortyMHzFrameTransmittedCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when a frame is transmitted on both control and
        secondary channels.

        This counter shall be incremented when a frame is transmitted on both con-
        trol and secondary channels."
    ::= { dot11CountersEntry 40 }
```

```
dot11TwentyMHzFrameReceivedCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the MAC when a frame is received only on the primary
      channel.

      This counter shall be incremented when a frame is received only on the
      primary channel."
   ::= { dot11CountersEntry 41 }

dot11FortyMHzFrameReceivedCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the MAC when a frame is received on both the control and
      secondary channels.

      This counter shall be incremented when a frame is received on both the
      control and secondary channels."
   ::= { dot11CountersEntry 42 }

dot11PSMPUTTGrantDuration OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the MAC when a PSMP-UTT is granted.

      This counter contains the cumulative duration of PSMP-UTT granted to the
      STA, in units of 4 microseconds."
   ::= { dot11CountersEntry 43 }

dot11PSMPUTTUsedDuration OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the MAC when a PSMP-UTT is used.

      This counter contains the cumulative duration of transmission by the STA
      during its allocated PSMP-UTT, in units of 4 microseconds"
   ::= { dot11CountersEntry 44 }

dot11GrantedRDGUsedCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the MAC when an RDG is used.

      This counter at the RD initiator shall be incremented when an allocated
      RDG is used by the station, apart from transmitting a response frame such
      as ACK or BlockAck frames."
   ::= { dot11CountersEntry 45 }

dot11GrantedRDGUnusedCount OBJECT-TYPE
```

```
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when an RDG is not used.

        This counter at the initiator shall be incremented when an allocated RDG
        is not used by the station, apart from transmitting a response frame such
        as ACK or BlockAck frames."
    ::= { dot11CountersEntry 46 }

dot11TransmittedFramesInGrantedRDGCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when an RDG is used.

        This counter at the initiator shall be incremented for every frame, other
        than response frames such as ACK or BlockAck frames, transmitted by the
        station during a granted RDG."
    ::= { dot11CountersEntry 47 }

dot11TransmittedOctetsInGrantedRDGCount OBJECT-TYPE
    SYNTAX Counter64
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when an RDG is used.

        This counter at the initiator shall be incremented by the number of octets
        in the framebody of a frame, other than response frames such as ACK or
        BlockAck frames, transmitted by the station during a granted RDG."
    ::= { dot11CountersEntry 48 }

dot11BeamformingFrameCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when a frame with beamforming parameters is sent.

        This counter shall be incremented when the transmitter sends a frame with
        new/updated beamforming parameters."
    ::= { dot11CountersEntry 49 }

dot11DualCTSSuccessCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when a dual CTS is sent.

        This counter shall be incremented when AP sends a dual CTS in response to
        a STA initiating TXOP in extended range."
    ::= { dot11CountersEntry 50 }

dot11DualCTSFailureCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when a dual CTS is not sent.

        This counter shall be incremented when AP fails to send a dual CTS in
        response to a STA initiating TXOP in extended range."
    ::= { dot11CountersEntry 51 }

dot11STBCCTSSuccessCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when no collision is detected.

        This counter shall be incremented when AP does not detect a collision PIFS
        after sending a CTS to self STBC frame in extended range."
    ::= { dot11CountersEntry 52 }

dot11STBCCTSFailureCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when a collision is detected.

        This counter shall be incremented when AP detects a collision PIFS after
        sending a CTS to self STBC frame in extended range."
    ::= { dot11CountersEntry 53 }

dot11nonSTBCCTSSuccessCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when no collision is detected.

        This counter shall be incremented when AP does not detect a collision PIFS
        after sending a CTS to self that is an non-STBC frame in extended range."
    ::= { dot11CountersEntry 54 }

dot11nonSTBCCTSFailureCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when a collision is detected.

        This counter shall be incremented when AP detects a collision PIFS after
        sending a CTS to self that is an non-STBC frame in extended range."
    ::= { dot11CountersEntry 55 }

dot11RTSLSIGSuccessCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when the condition described below occurs.

        This counter shall be incremented when the duration/ID field is set
        according to the rules of EPP in the received CTS following a transmission
        of RTS in EPP mode."
    ::= { dot11CountersEntry 56 }

dot11RTSLSIGFailureCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when the condition described below occurs.

        This counter shall be incremented when the duration/ID field is not set
        according to the rules of EPP in the received CTS following a transmission
        of RTS in EPP mode."
    ::= { dot11CountersEntry 57 }

dot11PBACErrors OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when the condition described below occurs.

        This variable indicates the number of errors encountered in the PBAC pro-
        cedures."
    ::= { dot11CountersEntry 58}

dot11DeniedAssociationCounterDueToBSSLoad OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the SME when the condition described below occurs.

        This counter, available at a WNM AP, shall increment when an association
        or reassociation request is denied because the AP has insufficient band-
        width to handle the additional STA."
    ::= { dot11CountersEntry 59}

-- **********************************************************************
-- * End of dot11Counters TABLE
-- **********************************************************************

-- **********************************************************************
-- * dot11GroupAddresses  TABLE
-- **********************************************************************

dot11GroupAddressesTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11GroupAddressesEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "A conceptual table containing a set of MAC addresses identifying the mul-
        ticast-group addresses for which this STA receives frames. The default
        value of this attribute is null."
    ::= { dot11mac 3 }

dot11GroupAddressesEntry OBJECT-TYPE
```

```
    SYNTAX Dot11GroupAddressesEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An Entry (conceptual row) in the Group Addresses Table.

        ifIndex - Each IEEE 802.11 interface is represented by an ifEntry. Inter-
        face tables in this MIB module are indexed by ifIndex."
    INDEX { ifIndex, dot11GroupAddressesIndex }
    ::= { dot11GroupAddressesTable  1 }

Dot11GroupAddressesEntry ::=
    SEQUENCE {
        dot11GroupAddressesIndex    InterfaceIndex,
        dot11Address                                            MacAddress,
        dot11GroupAddressesStatus                               RowStatus }

dot11GroupAddressesIndex OBJECT-TYPE
    SYNTAX InterfaceIndex
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "The auxiliary variable used to identify instances of the columnar objects
        in the Group Addresses Table."
    ::= { dot11GroupAddressesEntry 1 }

dot11Address OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        MAC address identifying multicast-group addresses from which this STA
        receives frames."
    ::= { dot11GroupAddressesEntry 2 }

dot11GroupAddressesStatus OBJECT-TYPE
    SYNTAX RowStatus
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "The status column used for creating, modifying, and deleting instances of
        the columnar objects in the Group Addresses Table."
    DEFVAL { active }
    ::= { dot11GroupAddressesEntry 3 }

-- **********************************************************************
-- *    End of dot11GroupAddresses TABLE
-- **********************************************************************

-- **********************************************************************
-- *    SMT EDCA Config TABLE
-- **********************************************************************

dot11EDCATable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11EDCAEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Conceptual table for EDCA default parameter values at a non-AP STA. This
        table contains the four entries of the EDCA parameters corresponding to
        four possible ACs. Index 1 corresponds to AC_BK, index 2 to AC_BE, index 3
        to AC_VI, and index 4 to AC_VO."
    REFERENCE
        "IEEE 802.11-2012, 9.2.4.2"
    ::= { dot11mac 4 }

dot11EDCAEntry OBJECT-TYPE
    SYNTAX Dot11EDCAEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An Entry (conceptual row) in the EDCA Table.

        ifIndex - Each IEEE 802.11 interface is represented by an ifEntry. Inter-
        face tables in this MIB module are indexed by ifIndex."
    INDEX { ifIndex, dot11EDCATableIndex }
    ::= { dot11EDCATable  1 }

Dot11EDCAEntry ::=
    SEQUENCE {
        dot11EDCATableIndex                              Unsigned32,
        dot11EDCATableCWmin                              Unsigned32,
        dot11EDCATableCWmax                              Unsigned32,
        dot11EDCATableAIFSN                              Unsigned32,
        dot11EDCATableTXOPLimit                          Unsigned32,
        dot11EDCATableMSDULifetime                       Unsigned32,
        dot11EDCATableMandatory                          TruthValue }

dot11EDCATableIndex OBJECT-TYPE
    SYNTAX Unsigned32 (1..4)
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "The auxiliary variable used to identify instances of the columnar objects
        in the EDCA Table. The value of this variable is
        1, if the value of the AC is AC_BK.
        2, if the value of the AC is AC_BE.
        3, if the value of the AC is AC_VI.
        4, if the value of the AC is AC_VO."
    ::= { dot11EDCAEntry 1 }

dot11EDCATableCWmin OBJECT-TYPE
    SYNTAX Unsigned32 (0..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by the MAC upon receiving an EDCA Parameter Set in a Beacon
        frame.
        Changes take effect as soon as practical in the implementation.

        This attribute specifies the value of the minimum size of the window that
        is used by a STA for a particular AC for generating a random number for
        the backoff. The value of this attribute is such that it could always be
        expressed in the form of 2**X - 1, where X is an integer. The default
        value for this attribute is
        aCWmin, if dot11EDCATableIndex is 1 or 2.
        (aCWmin+1)/2 - 1, if dot11EDCATableIndex is 3.
        (aCWmin+1)/4 - 1, if dot11EDCATableIndex is 4."
    ::= { dot11EDCAEntry 2 }

dot11EDCATableCWmax OBJECT-TYPE
    SYNTAX Unsigned32 (0..65535)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by the MAC upon receiving an EDCA Parameter Set in a Beacon
        frame.
        Changes take effect as soon as practical in the implementation.

        This attribute specifies the value of the maximum size of the window that
        is used by a STA for a particular AC for generating a random number for
        the backoff. The value of this attribute is such that it could always be
        expressed in the form of 2**X - 1, where X is an integer. The default
        value for this attribute is
        aCWmax, if dot11EDCATableIndex is 1 or 2.
        aCWmin, if dot11EDCATableIndex is 3.
        (aCWmin+1)/2 - 1, if dot11EDCATableIndex is 4."
    ::= { dot11EDCAEntry 3 }

dot11EDCATableAIFSN OBJECT-TYPE
    SYNTAX Unsigned32 (2..15)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by the MAC upon receiving an EDCA Parameter Set in a Beacon
        frame.
        Changes take effect as soon as practical in the implementation.

        This attribute specifies the number of slots, after a SIFS duration, that
        the STA, for a particular AC, senses the medium idle either before trans-
        mitting or executing a backoff. The default value for this attribute is
        7, if dot11EDCATableIndex is 1,
        3, if dot11EDCATableIndex is 2
        2, otherwise."
    ::= { dot11EDCAEntry 4 }

dot11EDCATableTXOPLimit OBJECT-TYPE
    SYNTAX Unsigned32 (0..65535)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by the MAC upon receiving an EDCA Parameter Set in a Beacon
        frame.
        Changes take effect as soon as practical in the implementation.

        This attribute specifies the maximum number of microseconds of an EDCA
        TXOP for a given AC. The default value for this attribute is
        1)  0 for all PHYs, if dot11EDCATableIndex is 1 or 2; this implies that
        the sender can send one MSDU in an EDCA TXOP,
        2)  3008 microseconds for Clause 18 and Clause 19 PHY and 6016 microsec-
        onds for Clause 17 PHY, if dot11EDCATableIndex is 3,
        3)  1504 microseconds for Clause 18 and Clause 19) PHY and 3264 microsec-
        onds for Clause 17 PHY, if dot11EDCATableIndex is 4."
    ::= { dot11EDCAEntry 5 }

dot11EDCATableMSDULifetime OBJECT-TYPE
    SYNTAX Unsigned32 (0..500)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by the MAC upon receiving an EDCA Parameter Set in a Beacon
        frame.
        Changes take effect as soon as practical in the implementation.
```

```
      This attribute specifies (in TUs) the maximum duration an MSDU, for a
      given AC, would be retained by the MAC before it is discarded."
   DEFVAL { 500 }
   ::= { dot11EDCAEntry 6 }

dot11EDCATableMandatory OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by the MAC upon receiving an EDCA Parameter Set in a Beacon
      frame.
      Changes take effect as soon as practical in the implementation.

      This attribute, when true, indicates that admission control is mandatory
      for the given AC. When false, this attribute indicates that the admission
      control is not mandatory for the given AC."
   DEFVAL { false }
   ::= { dot11EDCAEntry 7 }

-- **********************************************************************
-- *    End of SMT EDCA Config TABLE
-- **********************************************************************

-- **********************************************************************
-- *    SMT AP EDCA Config TABLE
-- **********************************************************************

dot11QAPEDCATable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11QAPEDCAEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "Conceptual table for EDCA default parameter values at the AP. This table
      contains the four entries of the EDCA parameters corresponding to four
      possible ACs. Index 1 corresponds to AC_BK, index 2 to AC_BE, index 3 to
      AC_VI, and index 4 to AC_VO."
      REFERENCE
      "IEEE 802.11-2012, 9.19.2"
   ::= { dot11mac 5 }

dot11QAPEDCAEntry OBJECT-TYPE
   SYNTAX Dot11QAPEDCAEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "An Entry (conceptual row) in the EDCA Table.

      ifIndex - Each IEEE 802.11 interface is represented by an ifEntry. Inter-
      face tables in this MIB module are indexed by ifIndex."
   INDEX { ifIndex, dot11QAPEDCATableIndex }
   ::= { dot11QAPEDCATable  1 }

Dot11QAPEDCAEntry ::=
   SEQUENCE {
      dot11QAPEDCATableIndex                                Unsigned32,
      dot11QAPEDCATableCWmin                                Unsigned32,
      dot11QAPEDCATableCWmax                                Unsigned32,
      dot11QAPEDCATableAIFSN                                Unsigned32,
      dot11QAPEDCATableTXOPLimit                            Unsigned32,
      dot11QAPEDCATableMSDULifetime                         Unsigned32,
      dot11QAPEDCATableMandatory                            TruthValue }
```

```
dot11QAPEDCATableIndex OBJECT-TYPE
    SYNTAX Unsigned32 (1..4)
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "The auxiliary variable used to identify instances of the columnar objects
        in the EDCA Table. The value of this variable is
        1, if the value of the AC is AC_BK.
        2, if the value of the AC is AC_BE.
        3, if the value of the AC is AC_VI.
        4, if the value of the AC is AC_VO."
    ::= { dot11QAPEDCAEntry 1 }

dot11QAPEDCATableCWmin OBJECT-TYPE
    SYNTAX Unsigned32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute specifies the value of the minimum size of the window that
        is used by an AP for a particular AC for generating a random number for
        the backoff. The value of this attribute is such that it could always be
        expressed in the form of 2**X - 1, where X is an integer. The default
        value for this attribute is
        aCWmin, if dot11QAPEDCATableIndex is 1 or 2.
        (aCWmin+1)/2 - 1, if dot11QAPEDCATableIndex is 3.
        (aCWmin+1)/4 - 1, if dot11QAPEDCATableIndex is 4."
    ::= { dot11QAPEDCAEntry 2 }

dot11QAPEDCATableCWmax OBJECT-TYPE
    SYNTAX Unsigned32 (0..65535)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute specifies the value of the maximum size of the window that
        is used by an AP for a particular AC for generating a random number for
        the backoff. The value of this attribute is such that it could always be
        expressed in the form of 2**X - 1, where X is an integer. The default
        value for this attribute is
        aCWmax, if dot11QAPEDCATableIndex is 1.
        4*(aCWmin+1) - 1, if dot11QAPEDCATableIndex is 2.
        aCWmin, if dot11QAPEDCATableIndex is 3.
        (aCWmin+1)/2 - 1, if dot11QAPEDCATableIndex is 4."
    ::= { dot11QAPEDCAEntry 3 }

dot11QAPEDCATableAIFSN OBJECT-TYPE
    SYNTAX Unsigned32 (1..15)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute specifies the number of slots, after a SIFS duration, that
        the AP, for a particular AC, senses the medium idle either before trans-
```

```
      mitting or executing a backoff. The default value for this attribute is
      7, if dot11QAPEDCATableIndex is 1,
      3, if dot11QAPEDCATableIndex is 2
      1, otherwise."
   ::= { dot11QAPEDCAEntry 4 }

dot11QAPEDCATableTXOPLimit OBJECT-TYPE
   SYNTAX Unsigned32 (0..65535)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute specifies the maximum number of microseconds of an EDCA
      TXOP for a given AC at the AP. The default value for this attribute is
      1)  0 for all PHYs, if dot11QAPEDCATableIndex is 1 or 2; this implies that
      the sender can send one MSDU in an EDCA TXOP,
      2)  3008 microseconds for Clause 18 and Clause 19 PHY and 6016 microsec-
      onds for Clause 17 PHY, if dot11QAPEDCATableIndex is 3,
      3)  1504 microseconds for Clause 18 and Clause 19 PHY and 3264 microsec-
      onds for Clause 17 PHY, if dot11QAPEDCATableIndex is 4."
   ::= { dot11QAPEDCAEntry 5 }

dot11QAPEDCATableMSDULifetime OBJECT-TYPE
   SYNTAX Unsigned32 (0..500)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute specifies (in TUs) the maximum duration an MSDU, for a
      given AC, would be retained by the MAC at the AP before it is discarded."
   DEFVAL { 500 }
   ::= { dot11QAPEDCAEntry 6 }

dot11QAPEDCATableMandatory OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute, when true, indicates that admission control is mandatory
      for the given AC. When false, this attribute indicates that the admission
      control is not mandatory for the given AC. The default value for this
      parameter is false."
   ::= { dot11QAPEDCAEntry 7 }

-- **********************************************************************
-- *    End of SMT AP EDCA Config TABLE
-- **********************************************************************

-- **********************************************************************
-- *    dot11QosCounters TABLE
-- **********************************************************************

dot11QosCountersTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11QosCountersEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "Group containing attributes that are MAC counters implemented as a table
      to allow for multiple instantiations on an agent."
   ::= { dot11mac 6 }

dot11QosCountersEntry OBJECT-TYPE
   SYNTAX Dot11QosCountersEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "An Entry (conceptual row) in the EDCA Table.

      ifIndex - Each IEEE 802.11 interface is represented by an ifEntry. Inter-
      face tables in this MIB module are indexed by ifIndex."
   INDEX { ifIndex, dot11QosCountersIndex }
   ::= { dot11QosCountersTable 1 }

Dot11QosCountersEntry ::=
   SEQUENCE {
      dot11QosCountersIndex                                   Unsigned32,
      dot11QosTransmittedFragmentCount                        Counter32,
      dot11QosFailedCount                                     Counter32,
      dot11QosRetryCount                                      Counter32,
      dot11QosMultipleRetryCount                              Counter32,
      dot11QosFrameDuplicateCount                             Counter32,
      dot11QosRTSSuccessCount                                 Counter32,
      dot11QosRTSFailureCount                                 Counter32,
      dot11QosACKFailureCount                                 Counter32,
      dot11QosReceivedFragmentCount                           Counter32,
      dot11QosTransmittedFrameCount                           Counter32,
      dot11QosDiscardedFrameCount                             Counter32,
      dot11QosMPDUsReceivedCount                              Counter32,
      dot11QosRetriesReceivedCount                            Counter32 }

dot11QosCountersIndex OBJECT-TYPE
   SYNTAX Unsigned32 (1..16)
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "The auxiliary variable used to identify instances of the columnar objects
      in the QoSCounter Table. The value of this variable is equal to TID + 1."
   ::= { dot11QosCountersEntry 1 }

dot11QosTransmittedFragmentCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the MAC when a QoS fragment is transmitted.

      This counter is incremented for an acknowledged MPDU, for a particular UP,
      with an individual address in the address 1 field or an MPDU with a group
      address in the address 1 field, either belonging to a particular TID. This
      counter has relevance only for TIDs between 0 and 7."
   ::= { dot11QosCountersEntry 2 }

dot11QosFailedCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
```

```
        "This is a status variable.
        It is written by the MAC when the condition described below occurs.

        This counter increments when an MSDU, for a particular UP, is not trans-
        mitted successfully due to the number of transmit attempts exceeding
        either the dot11ShortRetryLimit or dot11LongRetryLimit. This counter has
        relevance only for TIDs between 0 and 7."
    ::= { dot11QosCountersEntry 3 }

dot11QosRetryCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when the condition described below occurs.

        This counter increments when an MSDU, of a particular UP, is successfully
        transmitted after one or more retransmissions. This counter has relevance
        only for TIDs between 0 and 7."
    ::= { dot11QosCountersEntry 4 }

dot11QosMultipleRetryCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when the condition described below occurs.

        This counter increments when an MSDU, of a particular UP, is successfully
        transmitted after more than one retransmissions. This counter has rele-
        vance only for TIDs between 0 and 7."
    ::= { dot11QosCountersEntry 5 }

dot11QosFrameDuplicateCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when the condition described below occurs.

        This counter increments when a frame, of a particular UP, is received that
        the Sequence Control field indicates is a duplicate. This counter has rel-
        evance only for TIDs between 0 and 7."
    ::= { dot11QosCountersEntry 6 }

dot11QosRTSSuccessCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when a CTS is received in response to an RTS.

        This counter increments when a CTS is received in response to an RTS that
        has been sent for the transmission of an MPDU of a particular UP. This
        counter has relevance only for TIDs between 0 and 7."
    ::= { dot11QosCountersEntry 7 }

dot11QosRTSFailureCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when the condition described below occurs.

        This counter increments when a CTS is not received in response to an RTS
        that has been sent for the transmission of an MPDU of a particular UP.
        This counter has relevance only for TIDs between 0 and 7."
    ::= { dot11QosCountersEntry 8 }

dot11QosACKFailureCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when the condition described below occurs.

        This counter increments when an ACK is not received in response to an MPDU
        of a particular UP. This counter has relevance only for TIDs between 0 and
        7."
    ::= { dot11QosCountersEntry 9 }

dot11QosReceivedFragmentCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when a QoS fragment is received.

        This counter is incremented for each successfully received MPDU of type
        Data of a particular UP. This counter has relevance only for TIDs between
        0 and 7."
    ::= { dot11QosCountersEntry 10 }

dot11QosTransmittedFrameCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when a QoS frame is transmitted.

        This counter increments for each successfully transmitted MSDU of a par-
        ticular UP. This counter has relevance only for TIDs between 0 and 7."
    ::= { dot11QosCountersEntry 11 }

dot11QosDiscardedFrameCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when the condition described below occurs.

        This counter increments for each Discarded MSDU of a particular UP. This
        counter has relevance only for TIDs between 0 and 7."
    ::= { dot11QosCountersEntry 12 }

dot11QosMPDUsReceivedCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when a QoS MPDU is received.

        This counter increments for each received MPDU of a particular TID."
    ::= { dot11QosCountersEntry 13 }

dot11QosRetriesReceivedCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the MAC when the condition described below occurs.

        This counter increments for each received MPDU of a particular TID with
        the retry bit equal to 1."
    ::= { dot11QosCountersEntry 14 }

-- **********************************************************************
-- *    End of dot11QosCounters TABLE
-- **********************************************************************

-- **********************************************************************
-- *    Resource Type Attribute Templates
-- **********************************************************************

dot11ResourceTypeIDName OBJECT-TYPE
    SYNTAX DisplayString (SIZE(4))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Contains the name of the Resource Type ID managed object. The attribute
        is read-only and always contains the value RTID. This attribute value is
        not used as a naming attribute for any other managed object class."
    REFERENCE "IEEE Std 802.1F-1993, A.7"
    DEFVAL { "RTID" }
    ::= { dot11resAttribute 1 }

-- **********************************************************************
-- *    dot11ResourceInfo  TABLE
-- **********************************************************************

dot11ResourceInfoTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11ResourceInfoEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Provides a means of indicating, in data readable from a managed object,
        information that identifies the source of the implementation."
    REFERENCE "IEEE Std 802.1F-1993, A.7. Note that this standard has been with-
        drawn."
    ::= { dot11resAttribute 2 }

dot11ResourceInfoEntry OBJECT-TYPE
    SYNTAX Dot11ResourceInfoEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry in the dot11ResourceInfo Table.

        ifIndex - Each IEEE 802.11 interface is represented by an ifEntry. Inter-
        face tables in this MIB module are indexed by ifIndex."
    INDEX { ifIndex }
```

```
   ::= { dot11ResourceInfoTable 1 }

Dot11ResourceInfoEntry ::=
   SEQUENCE {
      dot11manufacturerOUI                               OCTET STRING,
      dot11manufacturerName                              DisplayString,
      dot11manufacturerProductName                       DisplayString,
      dot11manufacturerProductVersion                    DisplayString }

dot11manufacturerOUI OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(3))
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      Takes the value of an organizationally unique identifier."
   ::= { dot11ResourceInfoEntry 1 }

dot11manufacturerName OBJECT-TYPE
   SYNTAX DisplayString (SIZE(0..128))
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      A printable string used to identify the manufacturer of the resource. Max-
      imum string length is 128 octets."
   ::= { dot11ResourceInfoEntry 2 }

dot11manufacturerProductName OBJECT-TYPE
   SYNTAX DisplayString (SIZE(0..128))
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      A printable string used to identify the manufacturer's product name of the
      resource. Maximum string length is 128 octets."
   ::= { dot11ResourceInfoEntry 3 }

dot11manufacturerProductVersion OBJECT-TYPE
   SYNTAX DisplayString (SIZE(0..128))
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      Printable string used to identify the manufacturer's product version of
      the resource. Maximum string length is 128 octets."
   ::= { dot11ResourceInfoEntry 4 }

-- **********************************************************************
-- * End of dot11ResourceInfo  TABLE
-- **********************************************************************

-- **********************************************************************
-- * PHY Attribute Templates
-- **********************************************************************
-- **********************************************************************
-- * dot11PhyOperation  TABLE
-- **********************************************************************

dot11PhyOperationTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11PhyOperationEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "PHY level attributes concerned with operation. Implemented as a table
        indexed on ifIndex to allow for multiple instantiations on an Agent."
    ::= { dot11phy 1 }

dot11PhyOperationEntry OBJECT-TYPE
    SYNTAX Dot11PhyOperationEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry in the dot11PhyOperation Table.

        ifIndex - Each IEEE 802.11 interface is represented by an ifEntry. Inter-
        face tables in this MIB module are indexed by ifIndex."
    INDEX { ifIndex }
    ::= { dot11PhyOperationTable 1 }

Dot11PhyOperationEntry ::=
    SEQUENCE {
        dot11PHYType                                          INTEGER,
        dot11CurrentRegDomain                                 Unsigned32,
        dot11TempType                                         INTEGER }

dot11PHYType OBJECT-TYPE
    SYNTAX INTEGER {
        fhss(1),
        dsss(2),
        irbaseband(3),
        ofdm(4),
        hrdsss(5),
        erp(6),
        ht(7) }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the PHY.

        This is an 8-bit integer value that identifies the PHY type supported by
        the attached PLCP and PMD. Currently defined values and their correspond-
        ing PHY types are:

        FHSS 2.4 GHz = 01, DSSS 2.4 GHz = 02, IR Baseband = 03,
        OFDM = 04, HRDSSS = 05, ERP = 06, HT = 07"
    ::= { dot11PhyOperationEntry 1 }

dot11CurrentRegDomain OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by the SME when the device is initialized.

        The current regulatory domain this instance of the PMD is supporting. This
        object corresponds to one of the RegDomains listed in
        dot11RegDomainsSupported."
    ::= { dot11PhyOperationEntry 2 }

dot11TempType OBJECT-TYPE
    SYNTAX INTEGER { tempType1(1), tempType2(2) }
    MAX-ACCESS read-only
    STATUS deprecated
    DESCRIPTION
        "The use of dot11TempType is deprecated, because references to this vari-
        able have been removed from the normative text of IEEE Std 802.11-2012,
        and this entity may be removed in a later revision of the standard.
        This is a status variable.
        It is written by the PHY.

        There are different operating temperature requirements dependent on the
        anticipated environmental conditions. This attribute describes the current
        PHY's operating temperature range capability. Currently defined values and
        their corresponding temperature ranges are:

        Type 1 = X'01'-Commercial range of 0 to 40 degrees C,

        Type 2 = X'02'-Industrial range of -30 to 70 degrees C."
    ::= { dot11PhyOperationEntry 3 }

-- **********************************************************************
-- * End of dot11PhyOperation  TABLE
-- **********************************************************************

-- **********************************************************************
-- * dot11PhyAntenna  TABLE
-- **********************************************************************

dot11PhyAntennaTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11PhyAntennaEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Group of attributes for PhyAntenna. Implemented as a table indexed on
        ifIndex to allow for multiple instances on an agent."
    ::= { dot11phy 2}

dot11PhyAntennaEntry OBJECT-TYPE
    SYNTAX Dot11PhyAntennaEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry in the dot11PhyAntenna Table.

        ifIndex - Each IEEE 802.11 interface is represented by an ifEntry. Inter-
        face tables in this MIB module are indexed by ifIndex."
    INDEX { ifIndex }
    ::= { dot11PhyAntennaTable 1 }

Dot11PhyAntennaEntry ::=
    SEQUENCE {
        dot11CurrentTxAntenna                                   Unsigned32,
        dot11DiversitySupportImplemented                        INTEGER,
        dot11CurrentRxAntenna                                   Unsigned32,
        dot11AntennaSelectionOptionImplemented                  TruthValue,
        dot11TransmitExplicitCSIFeedbackASOptionImplemented     TruthValue,
        dot11TransmitIndicesFeedbackASOptionImplemented         TruthValue,
        dot11ExplicitCSIFeedbackASOptionImplemented             TruthValue,
        dot11TransmitIndicesComputationASOptionImplemented      TruthValue,
        dot11ReceiveAntennaSelectionOptionImplemented           TruthValue,
    dot11TransmitSoundingPPDUOptionImplemented            TruthValue,
    dot11NumberOfActiveRxAntennas                         Unsigned32 }

dot11CurrentTxAntenna OBJECT-TYPE
  SYNTAX Unsigned32 (1..255)
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "This is a control variable.
    It is written by an external management entity.
    Changes take effect as soon as practical in the implementation.

    The current antenna being used to transmit. This value is one of the val-
    ues appearing in dot11TxAntennaImplemented. This may be used by a manage-
    ment agent to control which antenna is used for transmission. "
  ::= { dot11PhyAntennaEntry 1 }

dot11DiversitySupportImplemented OBJECT-TYPE
  SYNTAX INTEGER { fixedlist(1), notsupported(2), dynamic(3) }
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "This is a capability variable.
    Its value is determined by device capabilities.

    This implementation's support for diversity, encoded as:

    X'01'-diversity is available and is performed over the fixed list of
    antennas defined in dot11DiversitySelectionRxImplemented.

    X'02'-diversity is not supported.

    X'03'-diversity is supported and control of diversity is also available,
    in which case the attribute dot11DiversitySelectionRxImplemented can be
    dynamically modified by the LME."
  ::= { dot11PhyAntennaEntry 2 }

dot11CurrentRxAntenna OBJECT-TYPE
  SYNTAX Unsigned32 (1..255)
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "This is a status variable.
    It is written by the PHY.

    The current antenna being used to receive, if the dot11 DiversitySupport
    indicates that diversity is not supported. The selected antenna is one of
    the antennae marked for receive in the dot11AntennasListTable."
  ::= { dot11PhyAntennaEntry 3 }

dot11AntennaSelectionOptionImplemented OBJECT-TYPE
  SYNTAX TruthValue
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "This is a capability variable.
    Its value is determined by device capabilities.

    This attribute, when true, indicates that ASEL is supported by the station
    implementation."
  DEFVAL { false }
  ::= { dot11PhyAntennaEntry 4 }

dot11TransmitExplicitCSIFeedbackASOptionImplemented OBJECT-TYPE
```

```
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute, when true, indicates that the transmit ASEL based on
        explicit CSI feedback is supported by the station implementation."
    DEFVAL { false }
    ::= { dot11PhyAntennaEntry 5 }

dot11TransmitIndicesFeedbackASOptionImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute, when true, indicates that the transmit ASEL based on
        antenna indices feedback is supported by the station implementation."
    DEFVAL { false }
    ::= { dot11PhyAntennaEntry 6 }

dot11ExplicitCSIFeedbackASOptionImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute, when true, indicates that the computation of CSI and feed-
        back the results to support the peer to do ASEL is supported by the sta-
        tion implementation."
    DEFVAL { false }
    ::= { dot11PhyAntennaEntry 7 }

dot11TransmitIndicesComputationASOptionImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute, when true, indicates that the transmit ASEL based on
        antenna indices selection computation and feedback the results to support
        the peer to do ASEL is supported by the station implementation."
    DEFVAL { false }
    ::= { dot11PhyAntennaEntry 8 }

dot11ReceiveAntennaSelectionOptionImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute, when true, indicates that the receive ASEL is supported by
        the station implementation."
    DEFVAL { false }
    ::= { dot11PhyAntennaEntry 9 }

dot11TransmitSoundingPPDUOptionImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute, when true, indicates that the transmission of sounding
        PPDUs is supported by the station implementation."
    DEFVAL { false }
    ::= { dot11PhyAntennaEntry 10 }

dot11NumberOfActiveRxAntennas OBJECT-TYPE
    SYNTAX Unsigned32 (1..4)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the PHY.

        This attribute indicates the number of current active antennas being used
        to receive."
    ::= { dot11PhyAntennaEntry 11 }

-- **********************************************************************
-- * End of dot11PhyAntenna  TABLE
-- **********************************************************************

-- **********************************************************************
-- * dot11PhyTxPower  TABLE
-- **********************************************************************

dot11PhyTxPowerTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11PhyTxPowerEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Group of attributes for dot11PhyTxPowerTable. Implemented as a table
        indexed on STA ID to allow for multiple instances on an Agent."
    ::= { dot11phy 3}

dot11PhyTxPowerEntry OBJECT-TYPE
    SYNTAX Dot11PhyTxPowerEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry in the dot11PhyTxPower Table.

        ifIndex - Each IEEE 802.11 interface is represented by an ifEntry. Inter-
        face tables in this MIB module are indexed by ifIndex."
    INDEX { ifIndex }
    ::= { dot11PhyTxPowerTable 1 }

Dot11PhyTxPowerEntry ::=
    SEQUENCE {
        dot11NumberSupportedPowerLevelsImplemented            Unsigned32,
        dot11TxPowerLevel1                                    Unsigned32,
        dot11TxPowerLevel2                                    Unsigned32,
        dot11TxPowerLevel3                                    Unsigned32,
        dot11TxPowerLevel4                                    Unsigned32,
        dot11TxPowerLevel5                                    Unsigned32,
        dot11TxPowerLevel6                                    Unsigned32,
        dot11TxPowerLevel7                                    Unsigned32,
        dot11TxPowerLevel8                                    Unsigned32,
        dot11CurrentTxPowerLevel                              Unsigned32 }

dot11NumberSupportedPowerLevelsImplemented OBJECT-TYPE
    SYNTAX Unsigned32 (1..8)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        The number of power levels supported by the PMD. This attribute can have a
        value of 1 to 8."
    ::= { dot11PhyTxPowerEntry 1 }

dot11TxPowerLevel1 OBJECT-TYPE
    SYNTAX Unsigned32 (0..10000)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        The transmit output power for LEVEL1 in mW. This is also the default power
        level."
    ::= { dot11PhyTxPowerEntry 2 }

dot11TxPowerLevel2 OBJECT-TYPE
    SYNTAX Unsigned32 (0..10000)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        The transmit output power for LEVEL2 in mW."
    ::= { dot11PhyTxPowerEntry 3 }

dot11TxPowerLevel3 OBJECT-TYPE
    SYNTAX Unsigned32 (0..10000)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        The transmit output power for LEVEL3 in mW."
    ::= { dot11PhyTxPowerEntry 4 }

dot11TxPowerLevel4 OBJECT-TYPE
    SYNTAX Unsigned32 (0..10000)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        The transmit output power for LEVEL4 in mW."
    ::= { dot11PhyTxPowerEntry 5 }

dot11TxPowerLevel5 OBJECT-TYPE
    SYNTAX Unsigned32 (0..10000)
```

```
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        The transmit output power for LEVEL5 in mW."
    ::= { dot11PhyTxPowerEntry 6 }

dot11TxPowerLevel6 OBJECT-TYPE
    SYNTAX Unsigned32 (0..10000)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        The transmit output power for LEVEL6 in mW."
    ::= { dot11PhyTxPowerEntry 7 }

dot11TxPowerLevel7 OBJECT-TYPE
    SYNTAX Unsigned32 (0..10000)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        The transmit output power for LEVEL7 in mW."
    ::= { dot11PhyTxPowerEntry 8 }

dot11TxPowerLevel8 OBJECT-TYPE
    SYNTAX Unsigned32 (0..10000)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        The transmit output power for LEVEL8 in mW."
    ::= { dot11PhyTxPowerEntry 9 }

dot11CurrentTxPowerLevel OBJECT-TYPE
    SYNTAX Unsigned32 (1..8)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the PHY.

        The TxPowerLevel N currently being used to transmit data. Some PHYs also
        use this value to determine the receiver sensitivity requirements for
        CCA."
    ::= { dot11PhyTxPowerEntry 10 }

-- **********************************************************************
-- * End of dot11PhyTxPower  TABLE
-- **********************************************************************

-- **********************************************************************
-- * dot11PhyFHSS  TABLE
-- **********************************************************************

dot11PhyFHSSTable OBJECT-TYPE
```

```
    SYNTAX SEQUENCE OF Dot11PhyFHSSEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Group of attributes for dot11PhyFHSSTable. Implemented as a table indexed
        on STA ID to allow for multiple instances on an Agent."
    ::= { dot11phy 4 }

dot11PhyFHSSEntry OBJECT-TYPE
    SYNTAX Dot11PhyFHSSEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry in the dot11PhyFHSS Table.

        ifIndex - Each IEEE 802.11 interface is represented by an ifEntry. Inter-
        face tables in this MIB module are indexed by ifIndex."
    INDEX { ifIndex }
    ::= { dot11PhyFHSSTable 1 }

Dot11PhyFHSSEntry ::=
    SEQUENCE {
        dot11HopTime                                          Unsigned32,
        dot11CurrentChannelNumber                             Unsigned32,
        dot11MaxDwellTime                                     Unsigned32,
        dot11CurrentDwellTime                                 Unsigned32,
        dot11CurrentSet                                       Unsigned32,
        dot11CurrentPattern                                   Unsigned32,
        dot11CurrentIndex                                     Unsigned32,
        dot11EHCCPrimeRadix                                   Unsigned32,
        dot11EHCCNumberofChannelsFamilyIndex                  Unsigned32,
        dot11EHCCCapabilityImplemented                        TruthValue,
        dot11EHCCCapabilityActivated                          TruthValue,
        dot11HopAlgorithmAdopted                              INTEGER,
        dot11RandomTableFlag                                  TruthValue,
        dot11NumberofHoppingSets                              Unsigned32,
        dot11HopModulus                                       Unsigned32,
        dot11HopOffset                                        Unsigned32 }

dot11HopTime OBJECT-TYPE
    SYNTAX Unsigned32 (224)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        The time in microseconds for the PMD to change from channel 2 to channel
        80."
    ::= { dot11PhyFHSSEntry 1 }

dot11CurrentChannelNumber OBJECT-TYPE
    SYNTAX Unsigned32 (0..200)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the PHY.

        The current channel number of the frequency output by the RF synthesizer."
    ::= { dot11PhyFHSSEntry 2 }

dot11MaxDwellTime OBJECT-TYPE
    SYNTAX Unsigned32 (1..65535)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by the SME when the device is initialized.

        The maximum time in TU that the transmitter is permitted to operate on a
        single channel."
    ::= { dot11PhyFHSSEntry 3 }

dot11CurrentDwellTime OBJECT-TYPE
    SYNTAX Unsigned32 (1..65535)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        In a non-AP STA, it is written by the MAC when it receives an FH Parameter
        Set element, and changes take effect as soon as practical in the implemen-
        tation.
        In an AP, it is written by an external management entity, and changes take
        effect for the next Beacon.

        The current time in TU that the transmitter operates on a single channel,
        as set by the MAC."
    DEFVAL { 19 }
    ::= { dot11PhyFHSSEntry 4 }

dot11CurrentSet OBJECT-TYPE
    SYNTAX Unsigned32 (1..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        The current set of patterns the PLME is using to determine the hopping
        sequence. "
    ::= { dot11PhyFHSSEntry 5 }

dot11CurrentPattern OBJECT-TYPE
    SYNTAX Unsigned32 (0..255)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        The current pattern the PLME is using to determine the hop sequence."
    ::= { dot11PhyFHSSEntry 6 }

dot11CurrentIndex OBJECT-TYPE
    SYNTAX Unsigned32 (1..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the PHY.

        The current index value the PLME is using to determine the CurrentChannel-
        Number."
    ::= { dot11PhyFHSSEntry 7 }
```

```
dot11EHCCPrimeRadix OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute indicates the value to be used as the prime radix (N) in
      the HCC and EHCC algorithms."
   ::= { dot11PhyFHSSEntry 8 }

dot11EHCCNumberofChannelsFamilyIndex OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute indicates the value to be used as the maximum for the fam-
      ily index (a) in the HCC and EHCC algorithms. The value of this field is
      not less than the prime radix minus 3 (N - 3). The valid range of allowed
      values is (N - 1), (N - 2), and (N - 3)."
   ::= { dot11PhyFHSSEntry 9 }

dot11EHCCCapabilityImplemented OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      This attribute, when true, indicates that the station implementation is
      capable of generating the HCC or EHCC algorithms for determining Hopping
      patterns. The capability is disabled, otherwise."
   DEFVAL { false }
   ::= { dot11PhyFHSSEntry 10 }

dot11EHCCCapabilityActivated OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect for the next MLME-START.request primitive or MLME-
      JOIN.request primitive.

      This attribute, when true, indicates that the capability of the station to
      operate using the HCC or EHCC algorithms for determining Hopping Patterns
      is enabled. The capability is disabled, otherwise."
   DEFVAL { false }
   ::= { dot11PhyFHSSEntry 11 }

dot11HopAlgorithmAdopted OBJECT-TYPE
   SYNTAX INTEGER { crnt(1), hopindex(2), hcc(3) }
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute, indicates which of the algorithms is used to generate the
      Hopping Patterns.
      Valid values are:

      1 - hopping patterns as defined in Clause 14
      2 - hop index method (with or without table)
      3 - HCC/EHCC method"
   ::= { dot11PhyFHSSEntry 12 }

dot11RandomTableFlag OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute, indicates that a Random Table is present when the value is
      true. When the value is false it indicates that a Random Table is not
      present and that the hop index method is to be used to determine the hop-
      ping sequence."
   DEFVAL { true }
   ::= { dot11PhyFHSSEntry 13 }

dot11NumberofHoppingSets OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by the SME when the device is initialized.

      The Number of Sets field indicates the total number of sets within the
      hopping patterns."
   ::= { dot11PhyFHSSEntry 14 }

dot11HopModulus OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by the SME when the device is initialized.

      The number of allowed channels for the hopping set. This is defined by the
      governing regulatory agency for the country code of the country in which
      this device is operating."
   ::= { dot11PhyFHSSEntry 15 }

dot11HopOffset OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the PHY.

      The next position in the hopping set."
   ::= { dot11PhyFHSSEntry 16 }
```

```
-- **********************************************************************
-- * End of dot11PhyFHSS  TABLE
-- **********************************************************************

-- **********************************************************************
-- * dot11PhyDSSSEntry  TABLE
-- **********************************************************************

dot11PhyDSSSTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11PhyDSSSEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "Entry of attributes for dot11PhyDSSSEntry. Implemented as a table indexed
      on ifIndex to allow for multiple instances on an Agent."
   ::= { dot11phy 5 }

dot11PhyDSSSEntry OBJECT-TYPE
   SYNTAX Dot11PhyDSSSEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "An entry in the dot11PhyDSSSEntry Table.

      ifIndex - Each IEEE 802.11 interface is represented by an ifEntry. Inter-
      face tables in this MIB module are indexed by ifIndex."
   INDEX { ifIndex }
   ::= { dot11PhyDSSSTable 1 }

Dot11PhyDSSSEntry ::=
   SEQUENCE {
      dot11CurrentChannel                                  Unsigned32,
      dot11CCAModeSupported                                Unsigned32,
      dot11CurrentCCAMode                                  INTEGER,
      dot11EDThreshold                                     Integer32 }

dot11CurrentChannel OBJECT-TYPE
   SYNTAX Unsigned32 (1..14)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the PHY.

      The current operating frequency channel of the DSSS PHY. Valid channel
      numbers are as defined in 16.4.6.3"
   ::= { dot11PhyDSSSEntry 1 }

dot11CCAModeSupported OBJECT-TYPE
   SYNTAX Unsigned32 (1..7)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      dot11CCAModeSupported is a bit-significant value, representing all of the
      CCA modes supported by the PHY. Valid values are:

      energy detect only (ED_ONLY) = 01,
      carrier sense only (CS_ONLY) = 02,
      carrier sense and energy detect (ED_and_CS)= 04

      This attribute is not used to indicate the CCA modes supported by a higher
      rate extension PHY. Rather, the dot11HRCCAModeImplemented attribute is
      used to indicate the CCA modes of the higher rate extension PHY."
   ::= { dot11PhyDSSSEntry 2 }

dot11CurrentCCAMode OBJECT-TYPE
   SYNTAX INTEGER {
      edonly(1),
      csonly(2),
      edandcs(4), cswithtimer(8),
      hrcsanded(16) }
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      The current CCA method in operation. Valid values are:
      energy detect only (edonly) = 01,
      carrier sense only (csonly) = 02,
      carrier sense and energy detect (edandcs)= 04
      carrier sense with timer (cswithtimer)= 08
      high rate carrier sense and energy detect (hrcsanded)=16."
   ::= { dot11PhyDSSSEntry 3 }

dot11EDThreshold OBJECT-TYPE
   SYNTAX Integer32
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      The current Energy Detect Threshold being used by the DSSS PHY."
   ::= { dot11PhyDSSSEntry 4 }

-- **********************************************************************
-- * End of dot11PhyDSSSEntry  TABLE
-- **********************************************************************

-- **********************************************************************
-- * dot11PhyIR  TABLE
-- **********************************************************************

dot11PhyIRTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11PhyIREntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "Group of attributes for dot11PhyIRTable. Implemented as a table indexed
      on ifIndex to allow for multiple instances on an Agent."
   ::= { dot11phy 6 }

dot11PhyIREntry OBJECT-TYPE
   SYNTAX Dot11PhyIREntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "An entry in the dot11PhyIR Table.

      ifIndex - Each IEEE 802.11 interface is represented by an ifEntry. Inter-
      face tables in this MIB module are indexed by ifIndex."
   INDEX { ifIndex }
   ::= { dot11PhyIRTable 1 }

Dot11PhyIREntry ::=
   SEQUENCE {
      dot11CCAWatchdogTimerMax                          Unsigned32,
      dot11CCAWatchdogCountMax                          Unsigned32,
      dot11CCAWatchdogTimerMin                          Unsigned32,
      dot11CCAWatchdogCountMin                          Unsigned32 }

dot11CCAWatchdogTimerMax OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This parameter, together with CCAWatchdogCountMax, determines when energy
      detected in the channel can be ignored."
   ::= { dot11PhyIREntry 1 }

dot11CCAWatchdogCountMax OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This parameter, together with CCAWatchdogTimerMax, determines when energy
      detected in the channel can be ignored."
   ::= { dot11PhyIREntry 2 }

dot11CCAWatchdogTimerMin OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      The minimum value to which CCAWatchdogTimerMax can be set."
   ::= { dot11PhyIREntry 3 }

dot11CCAWatchdogCountMin OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      The minimum value to which CCAWatchdogCount can be set."
   ::= { dot11PhyIREntry 4 }

-- **********************************************************************
-- * End of dot11PhyIR  TABLE
-- **********************************************************************

-- **********************************************************************
-- * dot11RegDomainsSupported  TABLE
-- ********************************************************************

dot11RegDomainsSupportedTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11RegDomainsSupportedEntry
   MAX-ACCESS not-accessible
   STATUS deprecated
          DESCRIPTION
      "Superceded by dot11OperatingClassesTable.

      There are different operational requirements dependent on the regulatory
      domain. This attribute list describes the regulatory domains the PLCP and
      PMD support in this implementation. Currently defined values and their
      corresponding Regulatory Domains are:

      FCC (USA) = X'10', DOC (Canada) = X'20', ETSI (most of Europe) = X'30',
      Spain = X'31', France = X'32', Japan = X'40', China = X'50', Other = X'00'
      "
   ::= { dot11phy 7}

dot11RegDomainsSupportedEntry OBJECT-TYPE
   SYNTAX Dot11RegDomainsSupportedEntry
   MAX-ACCESS not-accessible
   STATUS deprecated
   DESCRIPTION
      "Superceded by dot11OperatingClassesTable.
      An entry in the dot11RegDomainsSupportedTable.

      ifIndex - Each IEEE 802.11 interface is represented by an ifEntry. Inter-
      face tables in this MIB module are indexed by ifIndex."
   INDEX { ifIndex, dot11RegDomainsSupportedIndex }
   ::= { dot11RegDomainsSupportedTable 1 }

Dot11RegDomainsSupportedEntry ::=
   SEQUENCE {
      dot11RegDomainsSupportedIndex                           Unsigned32,
      dot11RegDomainsImplementedValue                         INTEGER }

dot11RegDomainsSupportedIndex OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS not-accessible
   STATUS deprecated
   DESCRIPTION
      "The auxiliary variable used to identify instances of the columnar objects
      in the RegDomainsSupport Table."
   ::= { dot11RegDomainsSupportedEntry 1 }

dot11RegDomainsImplementedValue OBJECT-TYPE
   SYNTAX INTEGER {
      other (0),
      fcc(16),
      doc(32),
      etsi(48),
      spain (49),
      france(50),
      japan (64)}
   MAX-ACCESS read-only
   STATUS deprecated
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      There are different operational requirements dependent on the regulatory
      domain. This attribute list describes the regulatory domains the PLCP and
      PMD support in this implementation. Currently defined values and their
      corresponding Regulatory Domains are:

      FCC (USA) = X'10', DOC (Canada) = X'20', ETSI (most of Europe) = X'30',
      Spain = X'31', France = X'32', Japan = X'40', China = X'50' "
   ::= { dot11RegDomainsSupportedEntry 2 }

-- **********************************************************************
-- * End of dot11RegDomainsSupported  TABLE
-- **********************************************************************

-- **********************************************************************
-- * dot11AntennasList  TABLE
-- **********************************************************************

dot11AntennasListTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11AntennasListEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "This table represents the list of antennae. An antenna can be marked to
      be capable of transmitting, receiving, and/or for participation in receive
      diversity. Each entry in this table represents a single antenna with its
      properties. The maximum number of antennae that can be contained in this
      table is 255."
   ::= { dot11phy 8 }

dot11AntennasListEntry OBJECT-TYPE
   SYNTAX Dot11AntennasListEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "An entry in the dot11AntennasListTable, representing the properties of a
      single antenna.

      ifIndex - Each IEEE 802.11 interface is represented by an ifEntry. Inter-
      face tables in this MIB module are indexed by ifIndex."
   INDEX { ifIndex, dot11AntennaListIndex }
   ::= { dot11AntennasListTable 1 }

Dot11AntennasListEntry ::=
   SEQUENCE {
      dot11AntennaListIndex                                 Unsigned32,
      dot11TxAntennaImplemented                             TruthValue,
      dot11RxAntennaImplemented                             TruthValue,
      dot11DiversitySelectionRxImplemented                  TruthValue }

dot11AntennaListIndex OBJECT-TYPE
   SYNTAX Unsigned32 (1..255)
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "The unique index of an antenna which is used to identify the columnar
      objects in the dot11AntennasList Table."
   ::= { dot11AntennasListEntry 1 }

dot11TxAntennaImplemented OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.
      When true, this object indicates that the antenna represented by
      dot11AntennaIndex can be used as a transmit antenna."
   ::= { dot11AntennasListEntry 2 }

dot11RxAntennaImplemented OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      When true, this object indicates that the antenna represented by the
      dot11AntennaIndex can be used as a receive antenna."
   ::= { dot11AntennasListEntry 3 }

dot11DiversitySelectionRxImplemented OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      When true, this object indicates that the antenna represented by
      dot11AntennaIndex can be used for receive diversity. This object is true
      only if the antenna can be used as a receive antenna, as indicated by
      dot11RxAntennaImplemented."
   ::= { dot11AntennasListEntry 4 }

-- **********************************************************************
-- * End of dot11AntennasList  TABLE
-- **********************************************************************

-- **********************************************************************
-- * dot11SupportedDataRatesTx  TABLE
-- **********************************************************************

dot11SupportedDataRatesTxTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11SupportedDataRatesTxEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "The Transmit bit rates supported by the PLCP and PMD, represented by a
      count from X'02-X'7f, corresponding to data rates in increments of
      500kbit/s from 1 Mb/s to 63.5 Mb/s subject to limitations of each individ-
      ual PHY."
   ::= { dot11phy 9 }

dot11SupportedDataRatesTxEntry OBJECT-TYPE
   SYNTAX Dot11SupportedDataRatesTxEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "An Entry (conceptual row) in the dot11SupportedDataRatesTx Table.

      ifIndex - Each IEEE 802.11 interface is represented by an ifEntry. Inter-
      face tables in this MIB module are indexed by ifIndex."
   INDEX { ifIndex, dot11SupportedDataRatesTxIndex }
   ::= { dot11SupportedDataRatesTxTable  1 }

Dot11SupportedDataRatesTxEntry ::=
   SEQUENCE {
      dot11SupportedDataRatesTxIndex                        Unsigned32,
      dot11ImplementedDataRatesTxValue                      Unsigned32 }

dot11SupportedDataRatesTxIndex OBJECT-TYPE
   SYNTAX Unsigned32 (1..255)
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "Index object that identifies which data rate to access. Range is 1..255."
   ::= { dot11SupportedDataRatesTxEntry 1 }

dot11ImplementedDataRatesTxValue OBJECT-TYPE
   SYNTAX Unsigned32 (2..127)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      The Transmit bit rates supported by the PLCP and PMD, represented by a
      count from X'02-X'7f, corresponding to data rates in increments of
      500kbit/s from 1 Mb/s to 63.5 Mb/s subject to limitations of each individ-
      ual PHY."
   ::= { dot11SupportedDataRatesTxEntry 2 }

-- **********************************************************************
-- * End of dot11SupportedDataRatesTx  TABLE
-- **********************************************************************

-- **********************************************************************
-- * dot11SupportedDataRatesRx  TABLE
-- **********************************************************************

dot11SupportedDataRatesRxTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11SupportedDataRatesRxEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "The receive bit rates supported by the PLCP and PMD, represented by a
      count from X'002-X'7f, corresponding to data rates in increments of
      500kbit/s from 1 Mb/s to 63.5 Mb/s."
   ::= { dot11phy 10 }

dot11SupportedDataRatesRxEntry OBJECT-TYPE
   SYNTAX Dot11SupportedDataRatesRxEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "An Entry (conceptual row) in the dot11SupportedDataRatesRx Table.

      ifIndex - Each IEEE 802.11 interface is represented by an ifEntry. Inter-
      face tables in this MIB module are indexed by ifIndex."
   INDEX { ifIndex, dot11SupportedDataRatesRxIndex }
   ::= { dot11SupportedDataRatesRxTable  1 }

Dot11SupportedDataRatesRxEntry ::=
   SEQUENCE {
      dot11SupportedDataRatesRxIndex                        Unsigned32,
      dot11ImplementedDataRatesRxValue                      Unsigned32 }

dot11SupportedDataRatesRxIndex OBJECT-TYPE
   SYNTAX Unsigned32 (1..255)
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
```

```
      "Index object that identifies which data rate to access. Range is 1..255."
   ::= { dot11SupportedDataRatesRxEntry 1 }

dot11ImplementedDataRatesRxValue OBJECT-TYPE
   SYNTAX Unsigned32 (2..127)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      The receive bit rates supported by the PLCP and PMD, represented by a
      count from X'02-X'7f, corresponding to data rates in increments of
      500kbit/s from 1 Mb/s to 63.5 Mb/s."
   ::= { dot11SupportedDataRatesRxEntry 2 }

-- **********************************************************************
-- * End of dot11SupportedDataRatesRx  TABLE
-- **********************************************************************

-- **********************************************************************
-- * dot11PhyOFDM TABLE
-- **********************************************************************

dot11PhyOFDMTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11PhyOFDMEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "Group of attributes for dot11PhyOFDMTable. Implemented as a table indexed
      on ifindex to allow for multiple instances on an Agent."
   ::= { dot11phy 11 }

dot11PhyOFDMEntry OBJECT-TYPE
   SYNTAX Dot11PhyOFDMEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "An entry in the dot11PhyOFDM Table.

      ifIndex - Each IEEE 802.11 interface is represented by an ifEntry. Inter-
      face tables in this MIB module are indexed by ifIndex."
   INDEX { ifIndex }
   ::= { dot11PhyOFDMTable 1 }

Dot11PhyOFDMEntry ::=
   SEQUENCE {
      dot11CurrentFrequency                                    Unsigned32,
      dot11TIThreshold                                         Integer32,
      dot11FrequencyBandsImplemented                           Unsigned32,
      dot11ChannelStartingFactor                               Unsigned32,
      dot11FiveMHzOperationImplemented                         TruthValue,
      dot11TenMHzOperationImplemented                          TruthValue,
      dot11TwentyMHzOperationImplemented                       TruthValue,
      dot11PhyOFDMChannelWidth                                 INTEGER,
      dot11OFDMCCAEDImplemented                                TruthValue,
      dot11OFDMCCAEDRequired                                   TruthValue,
      dot11OFDMEDThreshold                                     Unsigned32,
      dot11STATransmitPowerClass                               INTEGER,
      dot11ACRType                                             INTEGER }

dot11CurrentFrequency OBJECT-TYPE
   SYNTAX Unsigned32 (0..200)
   MAX-ACCESS read-only
```

```
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by the SME.
        Changes take effect as soon as practical in the implementation.

        The number of the current operating frequency channel of the OFDM PHY."
    ::= { dot11PhyOFDMEntry 1 }

dot11TIThreshold OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-write
    STATUS deprecated
    DESCRIPTION
        "Superseded by PHY-specific or regulatory CCA energy detect limits.

        The Threshold being used to detect a busy medium (frequency). CCA reports
        a busy medium upon detecting the RSSI above this threshold."
    ::= { dot11PhyOFDMEntry 2 }

dot11FrequencyBandsImplemented OBJECT-TYPE
    SYNTAX Unsigned32 (1..127)
    MAX-ACCESS read-only
    STATUS deprecated
            DESCRIPTION
        "Superseded by subband-specific supported operating classes.
        This is a capability variable.
        Its value is determined by device capabilities.

        The capability of the OFDM PHY implementation to operate in the 4.9 GHz
        and 5 GHz bands. Coded as an integer value with bit 0 LSB as follows:
        bit 0 .. capable of operating in the 5.15-5.25 GHz band
        bit 1 .. capable of operating in the 5.25-5.35 GHz band
        bit 2 .. capable of operating in the 5.725-5.825 GHz band
        bit 3 .. capable of operating in the 5.47-5.725 GHz band
        bit 4 .. capable of operating in the lower Japanese (5.15-5.25 GHz) band
        bit 5 .. capable of operating in the 5.03-5.091 GHz band
        bit 6 .. capable of operating in the 4.94-4.99 GHz band
        For example, for an implementation capable of operating in the 5.15-5.35
        GHz bands this attribute would take the value 3."
    ::= { dot11PhyOFDMEntry 3 }

dot11ChannelStartingFactor OBJECT-TYPE
    SYNTAX Unsigned32 (8000..10000)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        The base factor from which channel center frequencies are calculated. This
        number is multiplied by 500 kHz to form the base frequency to be added to
        the channel number x 5 MHz."
    DEFVAL { 10000 }
    ::= { dot11PhyOFDMEntry 4 }

dot11FiveMHzOperationImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.
```

```
      This attribute, when true, indicates that the 5 MHz Operation is imple-
      mented."
   DEFVAL { false }
   ::= { dot11PhyOFDMEntry 5 }

dot11TenMHzOperationImplemented OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      This attribute, when true, indicates that the 10 MHz Operation is imple-
      mented."
   DEFVAL { false }
   ::= { dot11PhyOFDMEntry 6 }

dot11TwentyMHzOperationImplemented OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      This attribute, when true, indicates that the 20 MHz Operation is imple-
      mented."
   DEFVAL { true }
   ::= { dot11PhyOFDMEntry 7 }

dot11PhyOFDMChannelWidth OBJECT-TYPE
   SYNTAX INTEGER { width5MHz(1), width10MHz(2), width20MHz(3)}
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by the SME.
      Changes take effect as soon as practical in the implementation.

      This is an 8-bit integer value that identifies the OFDM PHY channel width.
      Currently defined values and their corresponding Channel widths are:
      5MHz = 01, 10MHz = 02, 20MHz = 03"
   ::= { dot11PhyOFDMEntry 8 }

dot11OFDMCCAEDImplemented OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      This attribute indicates that the OFDM PHY is capable of CCA-Energy
      Detect."
   ::= { dot11PhyOFDMEntry 9 }

dot11OFDMCCAEDRequired OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by the SME when the device is initialized for operation in a
      band defined by an Operating Class.

      This attribute indicates that the PHY CCA-Energy Detect functionality is
      enabled."
   ::= { dot11PhyOFDMEntry 10 }

dot11OFDMEDThreshold OBJECT-TYPE
   SYNTAX Unsigned32 (0..255)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written the SME when the device is initialized for operation in a
      band defined by an Operating Class, or written by an external management
      entity.
      Changes take effect as soon as practical in the implementation.

      The current Energy Detect Threshold being used by the OFDM PHY."
   ::= { dot11PhyOFDMEntry 11 }

dot11STATransmitPowerClass OBJECT-TYPE
   SYNTAX INTEGER { classA(1), classB(2), classC(3), classD(4) }
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "The station transmit power class: Class A=1, Class B=2, Class C=3, Class
      D=4 (as defined in D.2.2)."
   DEFVAL { 1 }
   ::=  { dot11PhyOFDMEntry 12 }

dot11ACRType OBJECT-TYPE
   SYNTAX INTEGER { standard(1), enhanced(2) }
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the SME.

      The Adjacent and Nonadjacent Channel Rejection performance:
      when this attribute = 1 the levels in Table 18-14 apply; when this attri-
      bute = 2 the levels in Table 18-15 apply."
   DEFVAL { 1 }
   ::=  { dot11PhyOFDMEntry 13 }

-- **********************************************************************
-- * End of dot11PhyOFDM TABLE
-- **********************************************************************

-- **********************************************************************
-- * dot11PhyHRDSSSEntry TABLE
-- **********************************************************************

dot11PhyHRDSSSTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11PhyHRDSSSEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "Entry of attributes for dot11PhyHRDSSSEntry. Implemented as a table
      indexed on ifIndex to allow for multiple instances on an Agent."
   ::= { dot11phy 12 }

dot11PhyHRDSSSEntry OBJECT-TYPE
   SYNTAX Dot11PhyHRDSSSEntry
```

```
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry in the dot11PhyHRDSSSEntry Table.

        ifIndex - Each IEEE 802.11 interface is represented by an ifEntry. Inter-
        face tables in this MIB module are indexed by ifIndex."
    INDEX { ifIndex }
    ::= { dot11PhyHRDSSSTable 1 }

Dot11PhyHRDSSSEntry ::=
    SEQUENCE {
        dot11ShortPreambleOptionImplemented                 TruthValue,
        dot11PBCCOptionImplemented                          TruthValue,
        dot11ChannelAgilityPresent                          TruthValue,
        dot11ChannelAgilityActivated                        TruthValue,
        dot11HRCCAModeImplemented                           Unsigned32 }

dot11ShortPreambleOptionImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute, when true, indicates that the short preamble option as
        defined in 17.2.2.3 is implemented."
    DEFVAL { false }
    ::= { dot11PhyHRDSSSEntry 1 }

dot11PBCCOptionImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute, when true, indicates that the PBCC modulation option as
        defined in  17.4.6.7 is implemented."
    DEFVAL { false }
    ::= { dot11PhyHRDSSSEntry 2 }

dot11ChannelAgilityPresent OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute indicates that the PHY is capable of channel agility."
    ::= { dot11PhyHRDSSSEntry 3 }

dot11ChannelAgilityActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect for the next MLME-START.request primitive or MLME-
        JOIN.request primitive.
```

```
      This attribute indicates that the PHY channel agility functionality is
      enabled."
   ::= { dot11PhyHRDSSSEntry 4 }

dot11HRCCAModeImplemented OBJECT-TYPE
   SYNTAX Unsigned32 (1..31)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      dot11HRCCAModeImplemented is a bit-significant value, representing all of
      the CCA modes supported by the PHY. Valid values are:
      energy detect only (ED_ONLY) = 01,
      carrier sense only (CS_ONLY) = 02,
      carrier sense and energy detect (ED_and_CS)= 04,
      carrier sense with timer (CS_and_Timer)= 08,
      high rate carrier sense and energy detect (HRCS_and_ED)= 16
      or the logical sum of any of these values. In the high rate extension PHY,
      this attribute is used in preference to the dot11CCAModeSupported attri-
      bute."
   ::= { dot11PhyHRDSSSEntry 5 }

-- **********************************************************************
-- * End of dot11PhyHRDSSSEntry TABLE
-- **********************************************************************

-- **********************************************************************
-- * dot11HoppingPattern TABLE
-- **********************************************************************

dot11HoppingPatternTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11HoppingPatternEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "The (conceptual) table of attributes necessary for a frequency hopping
      implementation to be able to create the hopping sequences necessary to
      operate in the subband for the associated domain country string."
   ::= { dot11phy 13 }

dot11HoppingPatternEntry OBJECT-TYPE
   SYNTAX Dot11HoppingPatternEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "An entry (conceptual row) in the Hopping Pattern Table that indicates the
      random hopping sequence to be followed.

      IfIndex - Each IEEE 802.11 interface is represented by an ifEntry. Inter-
      face tables in this MIB are indexed by ifIndex."
   INDEX { ifIndex, dot11HoppingPatternIndex }
   ::= { dot11HoppingPatternTable 1 }

Dot11HoppingPatternEntry ::=
   SEQUENCE {
      dot11HoppingPatternIndex                              Unsigned32,
      dot11RandomTableFieldNumber                           Unsigned32 }

dot11HoppingPatternIndex OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS not-accessible
```

```
    STATUS current
    DESCRIPTION
        "The auxiliary variable used to identify instances of the columnar objects
        in the Hopping Pattern Table."
    ::= { dot11HoppingPatternEntry 1 }

dot11RandomTableFieldNumber OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by the SME when the device is initialized.

        This attribute indicates the value of the starting channel number in the
        hopping sequence of the subband for the associated domain country string."
    DEFVAL { 0 }
    ::= { dot11HoppingPatternEntry 2 }

-- **********************************************************************
-- * End of dot11HoppingPattern TABLE
-- **********************************************************************

-- **********************************************************************
-- * dot11PhyERP TABLE
-- **********************************************************************

dot11PhyERPTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11PhyERPEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Entry of attributes for dot11PhyERPEntry. Implemented as a table indexed
        on ifIndex to allow for multiple instances on an Agent."
    ::= { dot11phy 14 }

dot11PhyERPEntry OBJECT-TYPE
    SYNTAX Dot11PhyERPEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry in the dot11PhyERPEntry Table.
        ifIndex - Each 802.11 interface is represented by an ifEntry. Interface
        tables in this MIB module are indexed by ifIndex."
    INDEX {ifIndex}
    ::= { dot11PhyERPTable 1 }

Dot11PhyERPEntry ::=
    SEQUENCE {
        dot11ERPPBCCOptionImplemented                       TruthValue,
        dot11ERPBCCOptionActivated                          TruthValue,
        dot11DSSSOFDMOptionImplemented                      TruthValue,
        dot11DSSSOFDMOptionActivated                        TruthValue,
        dot11ShortSlotTimeOptionImplemented                 TruthValue,
        dot11ShortSlotTimeOptionActivated                   TruthValue }

dot11ERPPBCCOptionImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.
        This attribute, when true, indicates that the ERP-PBCC modulation option
        as defined in 18.6 (ERP-PBCC operation specifications) is implemented."
    DEFVAL { false }
    ::= { dot11PhyERPEntry 1 }

dot11ERPBCCOptionActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect for the next MLME-START.request or MLME-JOIN.request.

        This attribute, when true, indicates that the ERP-PBCC option as defined
        in 18.6 (ERP-PBCC operation specifications) is enabled."
    DEFVAL { false }
    ::= { dot11PhyERPEntry 2 }

dot11DSSSOFDMOptionImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute, when true, indicates that the DSSS-OFDM option as defined
        in 18.7 (DSSS-OFDM operation specifications) is implemented."
    DEFVAL { false }
    ::= { dot11PhyERPEntry 3 }

dot11DSSSOFDMOptionActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect for the next MLME-START.request primitive or MLME-
        JOIN.request primitive.

        This attribute, when true, indicates that the DSSS-OFDM option as defined
        in 18.7 (DSSS-OFDM operation specifications) is enabled."
    DEFVAL { false }
    ::= { dot11PhyERPEntry 4 }

dot11ShortSlotTimeOptionImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute, when true, indicates that the Short Slot Time option as
        defined in 8.4.1.4 is implemented."
    DEFVAL { false }
    ::= { dot11PhyERPEntry 5}

dot11ShortSlotTimeOptionActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute, when true, indicates that the Short Slot Time option as
        defined in 8.4.1.4 is enabled."
    DEFVAL { false }
    ::= { dot11PhyERPEntry 6 }

-- **********************************************************************
-- * End of dot11PhyERP TABLE
-- **********************************************************************

-- **********************************************************************
-- * dot11 Phy HT TABLE
-- **********************************************************************

dot11PhyHTTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11PhyHTEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Entry of attributes for dot11PhyHTTable. Implemented as a table indexed
        on ifIndex to allow for multiple instances on an Agent."
    ::= { dot11phy 15 }

dot11PhyHTEntry OBJECT-TYPE
    SYNTAX Dot11PhyHTEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry in the dot11PhyHTEntry Table. ifIndex - Each IEEE 802.11 inter-
        face is represented by an ifEntry. Interface tables in this MIB module are
        indexed by ifIndex."
    INDEX {ifIndex}
    ::= { dot11PhyHTTable 1 }

Dot11PhyHTEntry ::=
    SEQUENCE {
        dot11FortyMHzOperationImplemented                   TruthValue,
        dot11FortyMHzOperationActivated                     TruthValue,
        dot11CurrentPrimaryChannel                          Unsigned32,
        dot11CurrentSecondaryChannel                        Unsigned32,
        dot11NumberOfSpatialStreamsImplemented              Unsigned32,
        dot11NumberOfSpatialStreamsActivated                Unsigned32,
        dot11HTGreenfieldOptionImplemented                  TruthValue,
        dot11HTGreenfieldOptionActivated                    TruthValue,
        dot11ShortGIOptionInTwentyImplemented               TruthValue,
        dot11ShortGIOptionInTwentyActivated                 TruthValue,
        dot11ShortGIOptionInFortyImplemented                TruthValue,
        dot11ShortGIOptionInFortyActivated                  TruthValue,
        dot11LDPCCodingOptionImplemented                    TruthValue,
        dot11LDPCCodingOptionActivated                      TruthValue,
        dot11TxSTBCOptionImplemented                        TruthValue,
        dot11TxSTBCOptionActivated                          TruthValue,
        dot11RxSTBCOptionImplemented                        TruthValue,
        dot11RxSTBCOptionActivated                          TruthValue,
        dot11BeamFormingOptionImplemented                   TruthValue,
        dot11BeamFormingOptionActivated                     TruthValue,
        dot11HighestSupportedDataRate                       Unsigned32,
        dot11TxMCSSetDefined                                TruthValue,
        dot11TxRxMCSSetNotEqual                             TruthValue,
        dot11TxMaximumNumberSpatialStreamsSupported         Unsigned32,
      dot11TxUnequalModulationSupported                     TruthValue }

dot11FortyMHzOperationImplemented OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      This attribute, when true, indicates that the 40 MHz Operation is imple-
      mented."
   DEFVAL { false }
   ::= { dot11PhyHTEntry 1 }

dot11FortyMHzOperationActivated OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity.
      Changes take effect as soon as practical in the implementation.

      This attribute, when true, indicates that the 40 MHz Operation is
      enabled."
   DEFVAL { false }
   ::= { dot11PhyHTEntry 2 }

dot11CurrentPrimaryChannel OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the PHY.

      This attribute indicates the operating channel. If 20/40 MHz BSS is cur-
      rently in use then this attribute indicates the primary channel."
   ::= { dot11PhyHTEntry 3 }

dot11CurrentSecondaryChannel OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
      It is written by the PHY.

      This attribute indicates the channel number of the secondary channel. If
      20/40 MHz BSS is not currently in use, this attribute value shall be 0."
   ::= { dot11PhyHTEntry 4 }

dot11NumberOfSpatialStreamsImplemented OBJECT-TYPE
   SYNTAX Unsigned32 (1..4)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      This attribute indicates the maximum number of spatial streams imple-
      mented."
   DEFVAL { 2 }
```

```
    ::= { dot11PhyHTEntry 5 }

dot11NumberOfSpatialStreamsActivated OBJECT-TYPE
    SYNTAX Unsigned32 (1..4)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute indicates the maximum number of spatial streams enabled."
    DEFVAL { 2 }
    ::= { dot11PhyHTEntry 6 }

dot11HTGreenfieldOptionImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute, when true, indicates that the HT-greenfield option is
        implemented."
    DEFVAL { false }
    ::= { dot11PhyHTEntry 7 }

dot11HTGreenfieldOptionActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute, when true, indicates that the HT-greenfield option is
        enabled."
    DEFVAL { false }
    ::= { dot11PhyHTEntry 8 }

dot11ShortGIOptionInTwentyImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute, when true, indicates that the Short Guard option is imple-
        mented for 20 MHz operation."
    DEFVAL { false }
    ::= { dot11PhyHTEntry 9 }

dot11ShortGIOptionInTwentyActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.
        This attribute, when true, indicates that the Short Guard option is
        enabled for 20 MHz operation."
    DEFVAL { false }
    ::= { dot11PhyHTEntry 10 }

dot11ShortGIOptionInFortyImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute, when true, indicates that the Short Guard option is imple-
        mented for 40 MHz operation."
    DEFVAL { false }
    ::= { dot11PhyHTEntry 11 }

dot11ShortGIOptionInFortyActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute, when true, indicates that the Short Guard option is
        enabled for 40 MHz operation."
    DEFVAL { false }
    ::= { dot11PhyHTEntry 12 }

dot11LDPCCodingOptionImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute, when true, indicates that the LDPC coding option is imple-
        mented."
    DEFVAL { false }
    ::= { dot11PhyHTEntry 13 }

dot11LDPCCodingOptionActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute, when true, indicates that the LDPC coding option is
        enabled."
    DEFVAL { false }
    ::= { dot11PhyHTEntry 14 }

dot11TxSTBCOptionImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
```

```
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute, when true, indicates that the entity is capable of trans-
        mitting frames using STBC option."
    DEFVAL { false }
    ::= { dot11PhyHTEntry 15 }

dot11TxSTBCOptionActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute, when true, indicates that the entity's capability of
        transmitting frames using STBC option is enabled."
    DEFVAL { false }
    ::= { dot11PhyHTEntry 16 }

dot11RxSTBCOptionImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute, when true, indicates that the entity is capable of receiv-
        ing frames that are sent using the STBC."
    DEFVAL { false }
    ::= { dot11PhyHTEntry 17 }

dot11RxSTBCOptionActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute, when true, indicates that the entity's capability of
        receiving frames that are sent using the STBC is enabled."
    DEFVAL { false }
    ::= { dot11PhyHTEntry 18 }

dot11BeamFormingOptionImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute, when true, indicates that the beamforming option is imple-
        mented."
    DEFVAL { false }
    ::= { dot11PhyHTEntry 19 }

dot11BeamFormingOptionActivated OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute, when true, indicates that the beamforming option is
        enabled."
    DEFVAL { false }
    ::= { dot11PhyHTEntry 20 }

dot11HighestSupportedDataRate OBJECT-TYPE
    SYNTAX Unsigned32 (0..600)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute shall specify the Highest Data Rate in Mb/s at which the
        station may receive data."
    DEFVAL { 0 }
    ::= { dot11PhyHTEntry 21 }

dot11TxMCSSetDefined OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
        It is written by an external management entity.
        Changes take effect as soon as practical in the implementation.

        This attribute, when true, indicates that the Tx MCS set is defined."
    DEFVAL { false }
    ::= { dot11PhyHTEntry 22 }

dot11TxRxMCSSetNotEqual OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.
        It is written by the PHY.

        This attribute, when true, indicates that the supported Tx and Rx MCS sets
        are not equal."
    DEFVAL { false }
    ::= { dot11PhyHTEntry 23 }

dot11TxMaximumNumberSpatialStreamsSupported OBJECT-TYPE
    SYNTAX Unsigned32 (0..3)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute indicates the Tx maximum number of spatial streams sup-
        ported."
    DEFVAL { 0 }
    ::= { dot11PhyHTEntry 24 }
```

```
dot11TxUnequalModulationSupported OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      This attribute, when true, indicates that Tx UEQM is supported."
   DEFVAL { false }
   ::= { dot11PhyHTEntry 25 }

-- **********************************************************************
-- * End of dot11 PHY HT TABLE
-- **********************************************************************

-- **********************************************************************
-- * dot11 Supported MCS Tx TABLE
-- **********************************************************************
dot11SupportedMCSTxTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11SupportedMCSTxEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "he Transmit MCS supported by the PLCP and PMD, represented by a count
      from 1 to 127, subject to limitations of each individual PHY."
   ::= { dot11phy 16 }

dot11SupportedMCSTxEntry OBJECT-TYPE
   SYNTAX Dot11SupportedMCSTxEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "An Entry (conceptual row) in the dot11SupportedMCSTx Table.
      ifIndex - Each IEEE 802.11 interface is represented by an
      ifEntry. Interface tables in this MIB module are indexed by ifIndex."
   INDEX { ifIndex, dot11SupportedMCSTxIndex }
   ::= { dot11SupportedMCSTxTable 1 }

Dot11SupportedMCSTxEntry ::=
   SEQUENCE {
      dot11SupportedMCSTxIndex                              Unsigned32,
      dot11SupportedMCSTxValue                              Unsigned32 }

dot11SupportedMCSTxIndex OBJECT-TYPE
   SYNTAX Unsigned32 (1..255)
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "Index object that identifies which MCS to access. Range is 1..255."
   ::= { dot11SupportedMCSTxEntry 1 }

dot11SupportedMCSTxValue OBJECT-TYPE
   SYNTAX Unsigned32 (1..127)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      The Transmit MCS supported by the PLCP and PMD, represented by a count
      from 1 to 127, subject to limitations of each individual PHY."
   ::= { dot11SupportedMCSTxEntry 2 }
```

```
-- **********************************************************************
-- * End of dot11 Supported MCS Tx TABLE
-- **********************************************************************

-- **********************************************************************
-- * dot11 Supported MCS Rx TABLE
-- **********************************************************************

dot11SupportedMCSRxTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11SupportedMCSRxEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "The receive MCS supported by the PLCP and PMD, represented by a count
      from 1 to 127, subject to limitations of each individual PHY."
   ::= { dot11phy 17 }

dot11SupportedMCSRxEntry OBJECT-TYPE
   SYNTAX Dot11SupportedMCSRxEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "An Entry (conceptual row) in the dot11SupportedMCSRx Table. ifIndex -
      Each IEEE 802.11 interface is represented by an ifEntry. Interface tables
      in this MIB module are indexed by ifIndex."
   INDEX { ifIndex, dot11SupportedMCSRxIndex }
   ::= { dot11SupportedMCSRxTable 1 }

Dot11SupportedMCSRxEntry ::=
   SEQUENCE {
      dot11SupportedMCSRxIndex                              Unsigned32,
      dot11SupportedMCSRxValue                              Unsigned32 }

dot11SupportedMCSRxIndex OBJECT-TYPE
   SYNTAX Unsigned32 (1..255)
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "Index object that identifies which MCS to access. Range is 1..255."
   ::= { dot11SupportedMCSRxEntry 1 }

dot11SupportedMCSRxValue OBJECT-TYPE
   SYNTAX Unsigned32 (1..127)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      The receive MCS supported by the PLCP and PMD, represented by a count from
      1 to 127, subject to limitations of each individual PHY."
   ::= { dot11SupportedMCSRxEntry 2 }

-- **********************************************************************
-- * End of dot11 Supported MCS Rx TABLE
-- **********************************************************************

-- **********************************************************************
-- * dot11 Transmit Beamforming Config TABLE
-- **********************************************************************

dot11TransmitBeamformingConfigTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11TransmitBeamformingConfigEntry
   MAX-ACCESS not-accessible
```

```
    STATUS current
    DESCRIPTION
        "Entry of attributes for dot11TransmitBeamformingConfigTable. Implemented
        as a table indexed on ifIndex to allow for multiple instances on an
        Agent."
    ::= { dot11phy 18 }

dot11TransmitBeamformingConfigEntry OBJECT-TYPE
    SYNTAX Dot11TransmitBeamformingConfigEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry in the dot11TransmitBeamformingConfig Table.
        ifIndex - Each IEEE 802.11 interface is represented by an ifEntry. Inter-
        face tables in this MIB module are indexed by ifIndex."
    INDEX {ifIndex}
    ::= { dot11TransmitBeamformingConfigTable 1 }

Dot11TransmitBeamformingConfigEntry ::=
    SEQUENCE {
        dot11ReceiveStaggerSoundingOptionImplemented           TruthValue,
        dot11TransmitStaggerSoundingOptionImplemented          TruthValue,
        dot11ReceiveNDPOptionImplemented                       TruthValue,
        dot11TransmitNDPOptionImplemented                      TruthValue,
        dot11ImplicitTransmitBeamformingOptionImplemented      TruthValue,
        dot11CalibrationOptionImplemented                      INTEGER,
        dot11ExplicitCSITransmitBeamformingOptionImplemented   TruthValue,
        dot11ExplicitNonCompressedBeamformingMatrixOptionImplemented
                                                               TruthValue,
        dot11ExplicitTransmitBeamformingCSIFeedbackOptionImplemented
                                                               INTEGER,
        dot11ExplicitNonCompressedBeamformingFeedbackOptionImplemented
                                                               INTEGER,
        dot11ExplicitCompressedBeamformingFeedbackOptionImplemented
                                                               INTEGER,
        dot11NumberBeamFormingCSISupportAntenna                Unsigned32,
        dot11NumberNonCompressedBeamformingMatrixSupportAntenna
                                                               Unsigned32,
        dot11NumberCompressedBeamformingMatrixSupportAntenna   Unsigned32 }

dot11ReceiveStaggerSoundingOptionImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute, when true, indicates that the STA implementation supports
        the receiving of staggered sounding frames."
    DEFVAL { false }
    ::= { dot11TransmitBeamformingConfigEntry 1 }

dot11TransmitStaggerSoundingOptionImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute, when true, indicates that the STA implementation supports
        the transmission of staggered sounding frames."
    DEFVAL { false }
    ::= { dot11TransmitBeamformingConfigEntry 2 }

dot11ReceiveNDPOptionImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute, when true, indicates that the STA implementation is capa-
        ble of receiving NDP as sounding frames."
    DEFVAL { false }
    ::= { dot11TransmitBeamformingConfigEntry 3 }

dot11TransmitNDPOptionImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute, when true, indicates that the STA implementation is capa-
        ble of transmitting NDP as sounding frames."
    DEFVAL { false }
    ::= { dot11TransmitBeamformingConfigEntry 4 }

dot11ImplicitTransmitBeamformingOptionImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute, when true, indicates that STA implementation is capable of
        applying implicit transmit beamforming."
    DEFVAL { false }
    ::= { dot11TransmitBeamformingConfigEntry 5 }

dot11CalibrationOptionImplemented OBJECT-TYPE
    SYNTAX INTEGER {
        inCapable (0),
        unableToInitiate (1),
        ableToInitiate (2),
        fullyCapable (3) }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
        Its value is determined by device capabilities.

        This attribute indicates the level of calibration supported by the STA
        implementation."
    DEFVAL { inCapable }
    ::= { dot11TransmitBeamformingConfigEntry 6 }

dot11ExplicitCSITransmitBeamformingOptionImplemented OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a capability variable.
```

```
      Its value is determined by device capabilities.

      This attribute, when true, indicates that STA implementation is capable of
      applying transmit beamforming using CSI explicit feedback in its transmis-
      sion."
   DEFVAL { false }
   ::= { dot11TransmitBeamformingConfigEntry 7 }

dot11ExplicitNonCompressedBeamformingMatrixOptionImplemented OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      This attribute, when true, indicates that STA implementation is capable of
      applying transmit beamforming using noncompressed beamforming feedback
      matrix explicit feedback in its transmission."
   DEFVAL { false }
   ::= { dot11TransmitBeamformingConfigEntry 8 }

dot11ExplicitTransmitBeamformingCSIFeedbackOptionImplemented OBJECT-TYPE
   SYNTAX INTEGER {
      inCapable (0),
      delayed (1),
      immediate (2),
      unsolicitedImmediate (3),
      aggregated (4),
      delayedAggregated (5),
      immediateAggregated(6),
      unsolicitedImmediateAggregated (7) }
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      This attribute indicates the level of CSI explicit feedback returned by
      the STA implementation."
   DEFVAL { inCapable }
   ::= { dot11TransmitBeamformingConfigEntry 9 }

dot11ExplicitNonCompressedBeamformingFeedbackOptionImplemented OBJECT-TYPE
   SYNTAX INTEGER {
      inCapable (0),
      delayed (1),
      immediate (2),
      unsolicitedImmediate (3),
      aggregated (4),
      delayedAggregated (5),
      immediateAggregated(6),
      unsolicitedImmediateAggregated (7) }
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      This attribute indicates the level of noncompressed beamforming feedback
      matrix explicit feedback returned by the STA implementation."
   DEFVAL { inCapable }
   ::= { dot11TransmitBeamformingConfigEntry 10 }
```

```
dot11ExplicitCompressedBeamformingFeedbackOptionImplemented OBJECT-TYPE
   SYNTAX INTEGER {
      inCapable (0),
      delayed (1),
      immediate (2),
      unsolicitedImmediate (3),
      aggregated (4),
      delayedAggregated (5),
      immediateAggregated(6),
      unsolicitedImmediateAggregated (7) }
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      This attribute indicates the level of noncompressed beamforming feedback
      matrix explicit feedback returned by the STA implementation."
   DEFVAL { inCapable }
   ::= { dot11TransmitBeamformingConfigEntry 11 }

dot11NumberBeamFormingCSISupportAntenna OBJECT-TYPE
   SYNTAX Unsigned32 (1..4)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      This attribute indicates the maximum number of beamforming antennas the
      beamformee can support when CSI feedback is required."
   ::= { dot11TransmitBeamformingConfigEntry 12 }

dot11NumberNonCompressedBeamformingMatrixSupportAntenna OBJECT-TYPE
   SYNTAX Unsigned32 (1..4)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      This attribute indicates the maximum number of beamforming antennas the
      beamformee can support when noncompressed beamforming feedback matrix
      feedback is required."
   ::= { dot11TransmitBeamformingConfigEntry 13 }

dot11NumberCompressedBeamformingMatrixSupportAntenna OBJECT-TYPE
   SYNTAX Unsigned32 (1..4)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a capability variable.
      Its value is determined by device capabilities.

      This attribute indicates the maximum number of beamforming antennas the
      beamformee can support when compressed beamforming feedback matrix feed-
      back is required."
   ::= { dot11TransmitBeamformingConfigEntry 14 }

-- **********************************************************************
-- * End of dot11 Transmit Beamforming Config TABLE
-- **********************************************************************

-- **********************************************************************
-- * End of dot11PhyERP TABLE
-- **********************************************************************


-- Interworking Management (IMT) Attributes
-- DEFINED AS "The Interworking management object class provides
-- the necessary support for an SSPN Interface function to manage
-- interworking with external systems. IMT objects are conceptual
-- objects for Interworking Service and are defined only for the
-- AP."

dot11imt OBJECT IDENTIFIER ::= {ieee802dot11 6}

-- IMT GROUPS
-- dot11BSSIdTable                  ::= { dot11imt 1 }
-- dot11InterworkingTable           ::= { dot11imt 2 }
-- dot11APLCI                       ::= { dot11imt 3 }
-- dot11APCivicLocation             ::= { dot11imt 4 }
-- dot11RoamingConsortiumTable      ::= { dot11imt 5 }
-- dot11DomainNameTable             ::= { dot11imt 6 }

-- Generic Advertisement Service (GAS) Attributes
-- DEFINED AS "The Generic Advertisement Service management
-- object class provides the necessary support for an Advertisement
-- service to interwork with external systems."

-- GAS GROUPS
-- dot11GASAdvertisementTable       ::= { dot11imt 7 }

--**********************************************************************
-- * dot11BSSId TABLE
--**********************************************************************

dot11BSSIdTable OBJECT-TYPE
   SYNTAX         SEQUENCE OF Dot11BSSIdEntry
   MAX-ACCESS     not-accessible
   STATUS         current
   DESCRIPTION
      "This object is a table of BSSIDs contained within an Access Point (AP)."
   ::= { dot11imt 1 }

dot11BSSIdEntry OBJECT-TYPE
   SYNTAX     Dot11BSSIdEntry
   MAX-ACCESS not-accessible
   STATUS     current
   DESCRIPTION
      "This object provides the attributes identifying a particular BSSID within
      an AP."
   INDEX { dot11APMacAddress }
   ::= { dot11BSSIdTable 1 }

Dot11BSSIdEntry ::=
   SEQUENCE{
      dot11APMacAddress                                    MacAddress
      }
dot11APMacAddress OBJECT-TYPE
   SYNTAX     MacAddress
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
      "This is a status variable.

      Changes take effect for the next MLME-START.request primitive.
```

```
        This object specifies the MAC address of the BSSID represented on a par-
        ticular BSSID interface and uniquely identifies this entry."
    ::= { dot11BSSIdEntry 1 }

--**********************************************************************
-- * End of dot11BSSId TABLE
--**********************************************************************

--**********************************************************************
-- * dot11Interworking TABLE
--**********************************************************************
dot11InterworkingTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11InterworkingEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "This table represents the non-AP STAs associated to the AP. An entry is
        created automatically by the AP when the STA becomes associated to the AP.
        The corresponding entry is deleted when the STA disassociates. Each STA
        added to this table is uniquely identified by its MAC address. This table
        is moved to a new AP following a successful STA BSS transition event."
    ::= { dot11imt 2 }

dot11InterworkingEntry OBJECT-TYPE
    SYNTAX Dot11InterworkingEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Each entry represents a conceptual row in the dot11InterworkingTable and
        provides information about permissions received from an SSPN Interface. If
        a non-AP STA does not receive permissions for one or more of these
        objects, then the object's default values or AP's locally defined config-
        uration may be used instead. If the AP's local policy(s) is more restric-
        tive than an object's value received from the SSPN Interface, then the
        AP's local policy shall be enforced. An entry is identified by the AP's
        MAC address to which the STA is associated and the STA's MAC address."
    INDEX { dot11APMacAddress, dot11NonAPStationMacAddress }
    ::= { dot11InterworkingTable 1 }

Dot11InterworkingEntry ::=
    SEQUENCE {
        dot11NonAPStationMacAddress                         MacAddress,
        dot11NonAPStationUserIdentity                       DisplayString,
        dot11NonAPStationInterworkingCapability             BITS,
        dot11NonAPStationAssociatedSSID                     OCTET STRING,
        dot11NonAPStationUnicastCipherSuite                 OCTET STRING,
        dot11NonAPStationBroadcastCipherSuite               OCTET STRING,
        dot11NonAPStationAuthAccessCategories               BITS,
        dot11NonAPStationAuthMaxVoiceRate                   Unsigned32,
        dot11NonAPStationAuthMaxVideoRate                   Unsigned32,
        dot11NonAPStationAuthMaxBestEffortRate              Unsigned32,
        dot11NonAPStationAuthMaxBackgroundRate              Unsigned32,
        dot11NonAPStationAuthMaxVoiceOctets                 Unsigned32,
        dot11NonAPStationAuthMaxVideoOctets                 Unsigned32,
        dot11NonAPStationAuthMaxBestEffortOctets            Unsigned32,
        dot11NonAPStationAuthMaxBackgroundOctets            Unsigned32,
        dot11NonAPStationAuthMaxHCCAHEMMOctets              Unsigned32,
        dot11NonAPStationAuthMaxTotalOctets                 Unsigned32,
        dot11NonAPStationAuthHCCAHEMM                       TruthValue,
        dot11NonAPStationAuthMaxHCCAHEMMRate                Unsigned32,
        dot11NonAPStationAuthHCCAHEMMDelay                  Unsigned32,
        dot11NonAPStationAuthSourceMulticast                TruthValue,
        dot11NonAPStationAuthMaxSourceMulticastRate         Unsigned32,
        dot11NonAPStationVoiceMSDUCount                     Counter32,
        dot11NonAPStationDroppedVoiceMSDUCount                Counter32,
        dot11NonAPStationVoiceOctetCount                      Counter32,
        dot11NonAPStationDroppedVoiceOctetCount               Counter32,
        dot11NonAPStationVideoMSDUCount                       Counter32,
        dot11NonAPStationDroppedVideoMSDUCount                Counter32,
        dot11NonAPStationVideoOctetCount                      Counter32,
        dot11NonAPStationDroppedVideoOctetCount               Counter32,
        dot11NonAPStationBestEffortMSDUCount                  Counter32,
        dot11NonAPStationDroppedBestEffortMSDUCount           Counter32,
        dot11NonAPStationBestEffortOctetCount                 Counter32,
        dot11NonAPStationDroppedBestEffortOctetCount          Counter32,
        dot11NonAPStationBackgroundMSDUCount                  Counter32,
        dot11NonAPStationDroppedBackgroundMSDUCount           Counter32,
        dot11NonAPStationBackgroundOctetCount                 Counter32,
        dot11NonAPStationDroppedBackgroundOctetCount          Counter32,
        dot11NonAPStationHCCAHEMMMSDUCount                    Counter32,
        dot11NonAPStationDroppedHCCAHEMMMSDUCount             Counter32,
        dot11NonAPStationHCCAHEMMOctetCount                   Counter32,
        dot11NonAPStationDroppedHCCAHEMMOctetCount            Counter32,
        dot11NonAPStationMulticastMSDUCount                   Counter32,
        dot11NonAPStationDroppedMulticastMSDUCount            Counter32,
        dot11NonAPStationMulticastOctetCount                  Counter32,
        dot11NonAPStationDroppedMulticastOctetCount           Counter32,
        dot11NonAPStationPowerManagementMode                  INTEGER,
        dot11NonAPStationAuthDls                              TruthValue,
        dot11NonAPStationVLANId                               Unsigned32,
        dot11NonAPStationVLANName                             DisplayString,
        dot11NonAPStationAddtsResultCode                      INTEGER}

  dot11NonAPStationMacAddress OBJECT-TYPE
     SYNTAX     MacAddress
     MAX-ACCESS read-only
     STATUS     current
     DESCRIPTION
        "This is a status variable.

        It is written by the SME after a non-AP STA associates to the BSS.

        This object specifies the MAC address of the non-AP STA for this entry and
        uniquely identifies this entry."
     ::= { dot11InterworkingEntry 1 }

  dot11NonAPStationUserIdentity OBJECT-TYPE
     SYNTAX DisplayString (SIZE(0..255))
     MAX-ACCESS read-only
     STATUS current
     DESCRIPTION
        "This is a status variable.

        It is written by the SME after a non-AP STA associates to the BSS.

        This attribute reflects the user identity for the subscriber operating
        this non-AP STA"
     ::= { dot11InterworkingEntry 2 }

  dot11NonAPStationInterworkingCapability OBJECT-TYPE
     SYNTAX BITS {
        interworkingCapability(0),
        qosMapCapability(1),
        expeditedBwReqCapability(2),
        msgcfCapability(3)
     }
     MAX-ACCESS read-only
     STATUS current
```

```
    DESCRIPTION
       "This is a status variable.

       It is written by the SME after a non-AP STA associates to the BSS.

       This attribute defines the Interworking capabilities possessed by a non-AP
       STA. Interworking Capability is set to 1 when the STA includes the Inter-
       working element in its (Re)Association request. The QosMapCapability,
       ExpeditedBwReqCapability and MSGCFCapability bits reflect the same values
       and meanings as those defined in 8.4.2.29"
    ::= { dot11InterworkingEntry 3 }

dot11NonAPStationAssociatedSSID OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(0..32))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
       "This is a status variable.

       It is written by the SME after a non-AP STA associates to the BSS.

       This attribute reflects the SSID to which the non-AP STA is associated"
    ::= { dot11InterworkingEntry 4 }

dot11NonAPStationUnicastCipherSuite OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(4))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
       "This is a status variable.

       It is written by the SME after a non-AP STA authenticates with the BSS.

       The selector of the AKM cipher suite that is currently in use by the non-
       AP STA. It consists of an OUI (the first 3 octets) and a cipher suite
       identifier (the last octet)."
    ::= { dot11InterworkingEntry 5 }

dot11NonAPStationBroadcastCipherSuite OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(4))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
       "This is a status variable.

       It is written by the SME after a non-AP STA authenticates with the BSS.

       The selector of an AKM suite for broadcast and group addressed frame
       transmissions. It consists of an OUI (the first 3 octets) and a cipher
       suite identifier the last octet)."
    ::= { dot11InterworkingEntry 6 }

dot11NonAPStationAuthAccessCategories OBJECT-TYPE
    SYNTAX BITS {
       bestEffort(0),
       background(1),
       video(2),
       voice(3)
    }
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
       "This is a control variable.
```

```
        It is written by the SME after the AP receives the permissions for the
        non-AP STA from the SSPN Interface.

        The object that represents the access categories which the non-AP STA is
        permitted to use when admission control is configured on that AC. An AC is
        permitted to be used if its corresponding bit is set to 1; otherwise it is
        not permitted to be used."
    DEFVAL { { bestEffort, background, video, voice } }
    ::= { dot11InterworkingEntry 7 }

dot11NonAPStationAuthMaxVoiceRate OBJECT-TYPE
    SYNTAX Unsigned32 (1..4294967295)
    UNITS  "kb/s"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by the SME after the AP receives the permissions for the
        non-AP STA from the SSPN Interface.

        This attribute indicates the maximum authorized data rate in kb/s the non-
        AP STA may use, either transmitting to an AP or receiving from an AP on
        the voice access category. If this rate is exceeded, the AP should police
        the flows traversing this AC. The value '4294967295', which is the default
        value, means that the SSP is not requesting the AP to limit the data rate
        used by the non-AP STA. Local configuration of the AP, however, may cause
        the rate to be limited, especially when the AC is configured for mandatory
        admission control."
    DEFVAL {4294967295}
    ::= { dot11InterworkingEntry 8 }

dot11NonAPStationAuthMaxVideoRate OBJECT-TYPE
    SYNTAX Unsigned32 (1..4294967295)
    UNITS "kb/s"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by the SME after the AP receives the permissions for the
        non-AP STA from the SSPN Interface.

        This attribute indicates the maximum authorized data rate in kb/s the non-
        AP STA may use, either transmitting to an AP or receiving from an AP on
        the video access category. If this rate is exceeded, the AP should police
        the flows traversing this AC. The value '4294967295', which is the default
        value, means that the SSP is not requesting the AP to limit the data rate
        used by the non-AP STA. Local configuration of the AP, however, may cause
        the rate to be limited, especially when the AC is configured for mandatory
        admission control."
    DEFVAL {4294967295}
    ::= { dot11InterworkingEntry 9 }

dot11NonAPStationAuthMaxBestEffortRate OBJECT-TYPE
    SYNTAX Unsigned32 (1..4294967295)
    UNITS "kb/s"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by the SME after the AP receives the permissions for the
        non-AP STA from the SSPN Interface.
        This attribute indicates the maximum authorized data rate in kb/s the non-
        AP STA may use, either transmitting to an AP or receiving from an AP on
        the best effort access category. If this rate is exceeded, the AP should
        police the flows traversing this AC. The value '4294967295', which is the
        default value, means that the SSP is not requesting the AP to limit the
        data rate used by the non-AP STA. Local configuration of the AP, however,
        may cause the rate to be limited, especially when the AC is configured for
        mandatory admission control."
    DEFVAL {4294967295}
    ::= { dot11InterworkingEntry 10 }

dot11NonAPStationAuthMaxBackgroundRate OBJECT-TYPE
    SYNTAX Unsigned32 (1..4294967295)
    UNITS "kb/s"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by the SME after the AP receives the permissions for the
        non-AP STA from the SSPN Interface.

        This attribute indicates the maximum authorized data rate in kb/s the non-
        AP STA may use, either transmitting to an AP or receiving from an AP on
        the background access category. If this rate is exceeded, the AP should
        police the flows traversing this AC. The value '4294967295', which is the
        default value, means that the SSP is not requesting the AP to limit the
        data rate used by the non-AP STA. Local configuration of the AP, however,
        may cause the rate to be limited, especially when the AC is configured for
        mandatory admission control."
    DEFVAL {4294967295}
    ::= { dot11InterworkingEntry 11 }

dot11NonAPStationAuthMaxVoiceOctets OBJECT-TYPE
    SYNTAX Unsigned32 (0..4294967295)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by the SME after the AP receives the permissions for the
        non-AP STA from the SSPN Interface.

        This attribute indicates the maximum authorized total octet count that a
        STA may use on the voice access category. If this octet count is exceeded,
        the AP should disassociate the non-AP STA. A value of 0 indicates that
        there is no octet limit."
    DEFVAL {0}
    ::= { dot11InterworkingEntry 12 }

dot11NonAPStationAuthMaxVideoOctets OBJECT-TYPE
    SYNTAX Unsigned32 (0..4294967295)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by the SME after the AP receives the permissions for the
        non-AP STA from the SSPN Interface.

        This attribute indicates the maximum authorized total octet count that a
        STA may use on the video access category. If this octet count is exceeded,
        the AP should disassociate the non-AP STA. A value of 0 indicates that
```

```
      there is no octet limit."
   DEFVAL {0}
   ::= { dot11InterworkingEntry 13 }

dot11NonAPStationAuthMaxBestEffortOctets OBJECT-TYPE
   SYNTAX Unsigned32 (0..4294967295)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a control variable.

      It is written by the SME after the AP receives the permissions for the
      non-AP STA from the SSPN Interface.

      This attribute indicates the maximum authorized total octet count that a
      STA may use on the best effort access category. If this octet count is
      exceeded, the AP should disassociate the non-AP STA. A value of 0 indi-
      cates that there is no octet limit."
   DEFVAL {0}
   ::= { dot11InterworkingEntry 14 }

dot11NonAPStationAuthMaxBackgroundOctets OBJECT-TYPE
   SYNTAX Unsigned32 (0..4294967295)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a control variable.

      It is written by the SME after the AP receives the permissions for the
      non-AP STA from the SSPN Interface.

      This attribute indicates the maximum authorized total octet count that a
      STA may use on the background access category. If this octet count is
      exceeded, the AP should disassociate the non-AP STA. A value of 0 indi-
      cates that there is no octet limit."
   DEFVAL {0}
   ::= { dot11InterworkingEntry 15 }

dot11NonAPStationAuthMaxHCCAHEMMOctets OBJECT-TYPE
   SYNTAX Unsigned32 (0..4294967295)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a control variable.

      It is written by the SME after the AP receives the permissions for the
      non-AP STA from the SSPN Interface.

      This attribute indicates the maximum authorized total octet count that a
      STA may use with HCCA or HEMM access. If this octet count is exceeded, the
      AP should disassociate the non-AP STA. A value of 0 indicates that there
      is no octet limit."
   DEFVAL {0}
   ::= { dot11InterworkingEntry 16 }

dot11NonAPStationAuthMaxTotalOctets OBJECT-TYPE
   SYNTAX Unsigned32 (0..4294967295)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a control variable.

      It is written by the SME after the AP receives the permissions for the
      non-AP STA from the SSPN Interface.
```

```
        This attribute indicates the maximum authorized total octet count that a
        STA may use on all access categories combined. If this octet count is
        exceeded, the AP should disassociate the non-AP STA. A value of 0 indi-
        cates that there is no octet limit."
    DEFVAL {0}
    ::= { dot11InterworkingEntry 17 }

dot11NonAPStationAuthHCCAHEMM OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by the SME after the AP receives the permissions for the
        non-AP STA from the SSPN Interface.

        This attribute, when true, indicates that the non-AP STA is permitted by
        the SSP to request HCCA or HEMM service via ADDTS management frames. If
        this attribute is false, then HCCA or HEMM service is not permitted by the
        SSP."
    DEFVAL {true}
    ::= { dot11InterworkingEntry 18 }

dot11NonAPStationAuthMaxHCCAHEMMRate OBJECT-TYPE
    SYNTAX Unsigned32 (1..4294967295)
    UNITS "kb/s"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by the SME after the AP receives the permissions for the
        non-AP STA from the SSPN Interface.

        This attribute indicates the maximum authorized data rate in kb/s the non-
        AP STA may use, either transmitting to an AP or receiving from an AP via
        HCCA or HEMM. The value '4294967295', which is the default value, means
        that the SSP is not requesting the AP to limit the data rate used by the
        non-AP STA. Local configuration of the AP, however, may cause the rate to
        be otherwise limited."
    DEFVAL {4294967295}
    ::= { dot11InterworkingEntry 19 }

dot11NonAPStationAuthHCCAHEMMDelay OBJECT-TYPE
    SYNTAX Unsigned32 (1..4294967295)
    UNITS "microseconds"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by the SME after the AP receives the permissions for the
        non-AP STA from the SSPN Interface.

        This attribute indicates the delay bound for frames queued at an AP to a
        non-AP STA in the HCCA or HEMM queue. An AP should deliver frames to the
        non-AP STA within the time period specified in this attribute. When a non-
        AP STA requests admission control to the HCCA or HEMM queue, the requested
        delay will be equal to or higher than this value. The value '4294967295',
        which is the default value, means that the SSP is not requesting the AP
        limit the delay bound in this queue for transmissions to the non-AP STA."
    DEFVAL {4294967295}
    ::= { dot11InterworkingEntry 20 }

dot11NonAPStationAuthSourceMulticast OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by the SME after the AP receives the permissions for the
        non-AP STA from the SSPN Interface.

        This attribute, when true, indicates that the AP's MAC sublayer shall per-
        form rate limiting to enforce the resource utilization limit in
        dot11NonAPStationAuthMaxSourceMulticastRate in the dot11InterworkingEntry
        identified by the source MAC address of the received frame. If this attri-
        bute is false, at an AP for which dot11SSPNInterfaceActivated is true,
        upon receipt of a frame of type data with group DA, then the AP's MAC sub-
        layer shall discard the frame."
    DEFVAL {true}
    ::= { dot11InterworkingEntry 21 }

dot11NonAPStationAuthMaxSourceMulticastRate OBJECT-TYPE
    SYNTAX Unsigned32 (1..4294967295)
    UNITS "kb/s"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by the SME after the AP receives the permissions for the
        non-AP STA from the SSPN Interface.

        This attribute indicates the maximum authorized data rate in kb/s which
        the non-AP STA may transmit group addressed frames to an AP. If this rate
        is exceeded, the AP should police the flows. The value '4294967295', which
        is the default value, means that the SSP is not requesting the AP to limit
        the multicast data rate used by the non-AP STA."
    DEFVAL {4294967295}
    ::= { dot11InterworkingEntry 22 }

dot11NonAPStationVoiceMSDUCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.

        It is written by the AP's MAC after the completion of an MA-UNITDATA.con-
        firm or MA-UNITDATA.indication primitive.

        For EDCA operation, this counter shall be incremented for each MSDU suc-
        cessfully transmitted by the AP on the voice access category and for each
        MSDU successfully received on either user priority 6 or 7."
    ::= { dot11InterworkingEntry 23 }

dot11NonAPStationDroppedVoiceMSDUCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.

        It is written by the AP's MAC after the completion of an MA-UNITDATA.con-
```

```
      firm or MA-UNITDATA.indication primitive.

      For EDCA operation, this counter shall be incremented for each MSDU
      dropped by the AP on the voice access category."
   ::= { dot11InterworkingEntry 24 }

dot11NonAPStationVoiceOctetCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.

      It is written by the AP’s MAC after the completion of an MA-UNITDATA.con-
      firm or MA-UNITDATA.indication primitive.

      For EDCA operation, this counter shall be incremented by the octet length
      of each MSDU successfully transmitted by the AP on the voice access cate-
      gory and by the octet length of each MSDU successfully received on either
      user priority 6 or 7."
   ::= { dot11InterworkingEntry 25 }

dot11NonAPStationDroppedVoiceOctetCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.

      It is written by the AP’s MAC after the completion of an MA-UNITDATA.con-
      firm or MA-UNITDATA.indication primitive.

      For EDCA operation, this counter shall be incremented for each octet
      dropped by the AP on the voice access category."
   ::= { dot11InterworkingEntry 26 }

dot11NonAPStationVideoMSDUCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.

      It is written by the AP’s MAC after the completion of an MA-UNITDATA.con-
      firm or MA-UNITDATA.indication primitive.

      For EDCA operation, this counter shall be incremented for each MSDU suc-
      cessfully transmitted by the AP on the video access category and for each
      MSDU successfully received on either user priority 4 or 5."
   ::= { dot11InterworkingEntry 27 }

dot11NonAPStationDroppedVideoMSDUCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.

      It is written by the AP’s MAC after the completion of an MA-UNITDATA.con-
      firm or MA-UNITDATA.indication primitive.

      For EDCA operation, this counter shall be incremented for each MSDU
      dropped by the AP on the video access category."
   ::= { dot11InterworkingEntry 28 }
```

```
dot11NonAPStationVideoOctetCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.

        It is written by the AP’s MAC after the completion of an MA-UNITDATA.confirm or MA-UNITDATA.indication primitive.

        For EDCA operation, this counter shall be incremented by the octet length
        of each MSDU successfully transmitted by the AP on the voice access category and by the octet length of each MSDU successfully received on either
        user priority 4 or 5."
    ::= { dot11InterworkingEntry 29 }

dot11NonAPStationDroppedVideoOctetCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.

        It is written by the AP’s MAC after the completion of an MA-UNITDATA.confirm or MA-UNITDATA.indication primitive.

        For EDCA operation, this counter shall be incremented for each octet
        dropped by the AP on the video access category."
    ::= { dot11InterworkingEntry 30 }

dot11NonAPStationBestEffortMSDUCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.

        It is written by the AP’s MAC after the completion of an MA-UNITDATA.confirm or MA-UNITDATA.indication primitive.

        For EDCA operation, this counter shall be incremented for each MSDU successfully transmitted by the AP on the best effort access category and for
        each MSDU successfully received on either user priority 0 or 3. For DCF or
        PCF operation, this counter shall be incremented for each MSDU successfully transmitted or received by the AP."
    ::= { dot11InterworkingEntry 31 }

dot11NonAPStationDroppedBestEffortMSDUCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.

        It is written by the AP’s MAC after the completion of an MA-UNITDATA.confirm or MA-UNITDATA.indication primitive.

        For EDCA operation, this counter shall be incremented for each MSDU
        dropped by the AP on the best effort access category and for each MSDU
        dropped by the AP on either user priority 0 or 3. For DCF or PCF operation, this counter shall be incremented for each MSDU dropped by the AP."
    ::= { dot11InterworkingEntry 32 }

dot11NonAPStationBestEffortOctetCount OBJECT-TYPE
```

```
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.

        It is written by the AP’s MAC after the completion of an MA-UNITDATA.con-
        firm or MA-UNITDATA.indication primitive.

        For EDCA operation, this counter shall be incremented by the octet length
        of each MSDU successfully transmitted by the AP on the best effort access
        category and by the octet length of each MSDU successfully received on
        either user priority 0 or 3. For DCF or PCF operation, this counter shall
        be incremented the octet length of each MSDU successfully transmitted or
        received by the AP."
    ::= { dot11InterworkingEntry 33 }

dot11NonAPStationDroppedBestEffortOctetCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.

        It is written by the AP’s MAC after the completion of an MA-UNITDATA.con-
        firm or MA-UNITDATA.indication primitive.

        For EDCA operation, this counter shall be incremented for the octet length
        of each MSDU dropped by the AP on the best effort access category and by
        the octet length of each MSDU dropped by the AP for either user priority 0
        or 3. For DCF or PCF operation, this counter shall be incremented for the
        octet length of each MSDU dropped by the AP."
    ::= { dot11InterworkingEntry 34 }

dot11NonAPStationBackgroundMSDUCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.

        It is written by the AP’s MAC after the completion of an MA-UNITDATA.con-
        firm or MA-UNITDATA.indication primitive.

        For EDCA operation, this counter shall be incremented for each MSDU suc-
        cessfully transmitted by the AP on the background access category and for
        each MSDU successfully received on either user priority 1 or 2."
    ::= { dot11InterworkingEntry 35 }

dot11NonAPStationDroppedBackgroundMSDUCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.

        It is written by the AP’s MAC after the completion of an MA-UNITDATA.con-
        firm or MA-UNITDATA.indication primitive.

        For EDCA operation, this counter shall be incremented for each MSDU
        dropped by the AP on the background access category"
    ::= { dot11InterworkingEntry 36 }

dot11NonAPStationBackgroundOctetCount OBJECT-TYPE
```

```
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.

      It is written by the AP’s MAC after the completion of an MA-UNITDATA.con-
      firm or MA-UNITDATA.indication primitive.

      For EDCA operation, this counter shall be incremented by the octet length
      of each MSDU successfully transmitted by the AP on the background access
      category and by the octet length of each MSDU successfully received on
      either user priority 1 or 2."
   ::= { dot11InterworkingEntry 37 }

dot11NonAPStationDroppedBackgroundOctetCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.

      It is written by the AP’s MAC after the completion of an MA-UNITDATA.con-
      firm or MA-UNITDATA.indication primitive.

      For EDCA operation, this counter shall be incremented by the octet length
      of each MSDU dropped by the AP on the background access category"
   ::= { dot11InterworkingEntry 38 }

dot11NonAPStationHCCAHEMMMSDUCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.

      It is written by the AP’s MAC after the completion of an MA-UNITDATA.con-
      firm or MA-UNITDATA.indication primitive.

      For HCCA or HEMM operation, this counter shall be incremented for each
      MSDU successfully transmitted by the AP and for each MSDU successfully
      received on either."
   ::= { dot11InterworkingEntry 39 }

dot11NonAPStationDroppedHCCAHEMMMSDUCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.

      It is written by the AP’s MAC after the completion of an MA-UNITDATA.con-
      firm or MA-UNITDATA.indication primitive.

      For HCCA or HEMM operation, this counter shall be incremented for each
      MSDU dropped by the AP."
   ::= { dot11InterworkingEntry 40 }

dot11NonAPStationHCCAHEMMOctetCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
```

```
       It is written by the AP's MAC after the completion of an MA-UNITDATA.con-
       firm or MA-UNITDATA.indication primitive.

       For HCCA or HEMM operation, this counter shall be incremented by the octet
       length of each MSDU successfully transmitted by the AP and by the octet
       length of each MSDU successfully received."
   ::= { dot11InterworkingEntry 41 }

dot11NonAPStationDroppedHCCAHEMMOctetCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
       "This is a status variable.

       It is written by the AP's MAC after the completion of an MA-UNITDATA.con-
       firm or MA-UNITDATA.indication primitive.

       For HCCA or HEMM operation, this counter shall be incremented by the octet
       length of each MSDU dropped by the AP."
   ::= { dot11InterworkingEntry 42 }

dot11NonAPStationMulticastMSDUCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
       "This is a status variable.

       It is written by the AP's MAC after the completion of an MA-UNITDATA.con-
       firm or MA-UNITDATA.indication primitive.

       For Multicast operation, this counter shall be incremented for each Multi-
       cast MSDU successfully transmitted by the AP and for each Multicast MSDU
       successfully received at the AP."
   ::= { dot11InterworkingEntry 43 }

dot11NonAPStationDroppedMulticastMSDUCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
       "This is a status variable.

       It is written by the AP's MAC after the completion of an MA-UNITDATA.con-
       firm or MA-UNITDATA.indication primitive.

       For Multicast operation, this counter shall be incremented for each Multi-
       cast MSDU dropped by the AP."
   ::= { dot11InterworkingEntry 44 }

dot11NonAPStationMulticastOctetCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
       "This is a status variable.

       It is written by the AP's MAC after the completion of an MA-UNITDATA.con-
       firm or MA-UNITDATA.indication primitive.

       For Multicast operation, this counter shall be incremented by the octet
       length of each MSDU successfully transmitted by the AP and by the octet
```

```
      length of each Multicast MSDU successfully received."
   ::= { dot11InterworkingEntry 45 }

dot11NonAPStationDroppedMulticastOctetCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.

      It is written by the AP's MAC after the completion of an MA-UNITDATA.con-
      firm or MA-UNITDATA.indication primitive.

      For Multicast operation, this counter shall be incremented by the octet
      length of each Multicast MSDU dropped by the AP."
   ::= { dot11InterworkingEntry 46 }

dot11NonAPStationPowerManagementMode OBJECT-TYPE
   SYNTAX INTEGER { active(1), powersave(2) }
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.

      It is written by the AP's MAC after the non-AP STA changes it's power man-
      agement mode.

      This attribute indicates the power management mode of the non-AP STA."
   ::= { dot11InterworkingEntry 47 }

dot11NonAPStationAuthDls OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a control variable.

      It is written by the SME after the AP receives the permissions for the
      non-AP STA from the SSPN Interface.

      This attribute, when true, indicates that the non-AP STA is permitted by
      the SSPN Interface to use direct link service (DLS). Note this attribute
      is an SSP permission and is independent of whether DLS is allowed in the
      BSS as governed by dot11DLSAllowedInQBSS. This service is disabled other-
      wise."
   DEFVAL {true}
   ::= { dot11InterworkingEntry 48 }

dot11NonAPStationVLANId OBJECT-TYPE
   SYNTAX Unsigned32 (0..4095)
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.

      It is written by the SME after a non-AP STA associates to the BSS.

      This attribute indicates the VLAN ID on the an external network to which
      frames from the non-AP STA are bridged."
   ::= { dot11InterworkingEntry 49 }

dot11NonAPStationVLANName OBJECT-TYPE
   SYNTAX DisplayString (SIZE(0..64))
   MAX-ACCESS read-only
```

```
    STATUS current
    DESCRIPTION
        "This is a status variable.

        It is written by the SME after a non-AP STA associates to the BSS.

        This attribute indicates the VLAN name corresponding to the VLAN ID on the
        external network to which frames from the non-AP STA are bridged."
    ::= { dot11InterworkingEntry 50 }

dot11NonAPStationAddtsResultCode OBJECT-TYPE
    SYNTAX INTEGER {
        success(1),
        invalidParameters(2),
        rejectedWithSuggestedChanges(3),
        rejectedForDelayPeriod(4),
        rejectedForSspPermissions(5),
        rejectedWithSuggestedBssTransition (6),
        requestedTclasNotSupported (7),
        tclasResourcesExhausted (8),
        rejectedHomeWithSuggestedChanges (9)
    }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is a status variable.

        It is written by the AP's HC after the AP transmits an ADDTS Response to
        the non-AP STA or after the AP includes a RIC element in a Reassociation
        Response frame.

        This attribute indicates the most recent result code returned by the AP in
        an ADDTS Response."
    ::= { dot11InterworkingEntry 51 }

-- **********************************************************************
-- * End of dot11Interworking TABLE
-- **********************************************************************


-- **********************************************************************
-- * dot11APLCI TABLE
-- **********************************************************************

dot11APLCITable OBJECT-TYPE
    SYNTAX          SEQUENCE OF Dot11APLCIEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "This table represents the Geospatial location of the AP as specified in
        8.4.2.23.10."
    ::= { dot11imt 3 }

dot11APLCIEntry OBJECT-TYPE
    SYNTAX Dot11APLCIEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "AP location in Geospatial coordinates"
    INDEX { dot11APLCIIndex }
    ::= { dot11APLCITable 1 }

Dot11APLCIEntry ::=
    SEQUENCE {
        dot11APLCIIndex                                      Unsigned32,
        dot11APLCILatitudeResolution                         Unsigned32,
        dot11APLCILatitudeInteger                            Integer32,
        dot11APLCILatitudeFraction                           Integer32,
        dot11APLCILongitudeResolution                        Unsigned32,
        dot11APLCILongitudeInteger                           Integer32,
        dot11APLCILongitudeFraction                          Integer32,
        dot11APLCIAltitudeType                               INTEGER,
        dot11APLCIAltitudeResolution                         Unsigned32,
        dot11APLCIAltitudeInteger                            Integer32,
        dot11APLCIAltitudeFraction                           Integer32,
        dot11APLCIDatum                                      INTEGER,
        dot11APLCIAzimuthType                                INTEGER,
        dot11APLCIAzimuthResolution                          Unsigned32,
        dot11APLCIAzimuth                                    Integer32
        }
dot11APLCIIndex OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Index for AP LCI elements in dot11APLCITable, greater than 0."
    ::= { dot11APLCIEntry 1 }

dot11APLCILatitudeResolution OBJECT-TYPE
    SYNTAX Unsigned32 (0..63)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        Latitude resolution is 6 bits indicating the number of valid bits in the
        fixed-point value of Latitude. This field is derived from IETF RFC 3825."
    ::= { dot11APLCIEntry 2 }

dot11APLCILatitudeInteger OBJECT-TYPE
    SYNTAX Integer32 (-90..90)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        Latitude is a 34 bit fixed point value consisting of 9 bits of integer and
        25 bits of fraction. This field contains the 9 bits of integer portion of
        Latitude. This field is derived from IETF RFC 3825."
    ::= { dot11APLCIEntry 3 }

dot11APLCILatitudeFraction OBJECT-TYPE
    SYNTAX Integer32 (-16777215..16777215)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        Latitude is a 34 bit fixed point value consisting of 9 bits of integer and
        25 bits of fraction. This field contains the 25 bits of fraction portion
        of Latitude. This field is derived from IETF RFC 3825."
    ::= { dot11APLCIEntry 4 }

dot11APLCILongitudeResolution OBJECT-TYPE
    SYNTAX Unsigned32 (0..63)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        Longitude resolution is 6 bits indicating the number of valid bits in the
        fixed-point value of Longitude. This field is derived from IETF RFC 3825."
    ::= { dot11APLCIEntry 5 }

dot11APLCILongitudeInteger OBJECT-TYPE
    SYNTAX Integer32 (-180..180)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        Longitude is a 34 bit fixed point value consisting of 9 bits of integer
        and 25 bits of fraction. This field contains the 9 bits of integer portion
        of Longitude. This field is derived from IETF RFC 3825."
    ::= { dot11APLCIEntry 6 }

dot11APLCILongitudeFraction OBJECT-TYPE
    SYNTAX Integer32 (-16777215..16777215)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        Longitude is a twos complement 34 bit fixed point value consisting of 9
        bits of integer and 25 bits of fraction. This field contains the 25 bits
        of fraction portion of Longitude. This field is derived from IETF RFC
        3825."
    ::= { dot11APLCIEntry 7 }

dot11APLCIAltitudeType OBJECT-TYPE
    SYNTAX INTEGER {
        meters(1),
        floors(2),
        hagm (3) }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        Altitude Type is four bits encoding the type of altitude. Codes defined
        are: meters in 2s-complement fixed-point 22-bit integer part with 8-bit
```

```
        fraction floors in 2s-complement fixed-point 22-bit integer part with 8-
        bit fraction hagm: Height Above Ground in meters, in 2s-complement fixed-
        point 22-bit integer part with 8-bit fraction. This field is derived from
        IETF RFC 3825."
    ::= { dot11APLCIEntry 8 }

dot11APLCIAltitudeResolution OBJECT-TYPE
    SYNTAX Unsigned32 (0..63)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        Altitude resolution is 6 bits indicating the number of valid bits in the
        altitude. This field is derived from IETF RFC 3825."
    ::= { dot11APLCIEntry 9 }

dot11APLCIAltitudeInteger OBJECT-TYPE
    SYNTAX Integer32 (-2097151..2097151)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        Altitude is a 30 bit value defined by the Altitude type field. The field
        is encoded as a 2s-complement fixed-point 22-bit integer Part with 8-bit
        fraction. This field contains the fixed-point Part of Altitude. This field
        is derived from IETF RFC 3825."
    ::= { dot11APLCIEntry 10 }

dot11APLCIAltitudeFraction OBJECT-TYPE
    SYNTAX Integer32 (-127..127)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        Altitude is a 30 bit value defined by the Altitude type field. The field
        is encoded as a 2s-complement fixed-point 22-bit integer Part with 8-bit
        fraction. This field is derived from IETF RFC 3825."
    ::= { dot11APLCIEntry 11 }

dot11APLCIDatum OBJECT-TYPE
    SYNTAX INTEGER {
        wgs84 (1),
        nad83navd88 (2),
        nad93mllwvd (3)
    }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.
```

```
      Datum is an 8-bit value encoding the horizontal and vertical references
      used for the coordinates given in this LCI. IETF RFC 3825 defines the val-
      ues of Datum. Type 1 is WGS-84, the coordinate system used by GPS. Type 2
      is NAD83 with NAVD88 vertical reference. Type 3 is NAD83 with Mean Lower
      Low Water vertical datum. All other types are reserved. This field is
      derived from IETF RFC 3825."
   ::= { dot11APLCIEntry 12 }

dot11APLCIAzimuthType OBJECT-TYPE
   SYNTAX INTEGER {
      frontSurfaceOfSTA(0),
      radioBeam(1) }
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.

      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      Azimuth Type is a one bit attribute encoding the type of Azimuth. Codes
      defined are: front surface of STA: in 2s-complement fixed-point 9-bit
      integer radio beam: in 2s-complement fixed-point 9-bit integer."
   ::= { dot11APLCIEntry 13 }

dot11APLCIAzimuthResolution OBJECT-TYPE
   SYNTAX Unsigned32 (0..15)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.

      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      Azimuth Resolution is 4 bits indicating the number of valid bits in the
      azimuth."
   ::= { dot11APLCIEntry 14 }

dot11APLCIAzimuth OBJECT-TYPE
   SYNTAX Integer32 (-511..511)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.

      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      Azimuth is a 9 bit value defined by the Azimuth Type field.The field is
      encoded as a 2s-complement fixed-point 9-bit integer horizontal angle in
      degrees from true north."
   ::= { dot11APLCIEntry 15 }

-- **********************************************************************
-- * End of dot11APLCI TABLE
-- **********************************************************************

-- **********************************************************************
-- * dot11APCivicLocation TABLE
-- **********************************************************************
dot11APCivicLocationTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11ApCivicLocationEntry
```

```
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "This table represents the location of the AP in Civic format using the
        Civic Address Type elements defined in IETF RFC-5139 [B42]."
    ::= { dot11imt 4 }

dot11APCivicLocationEntry OBJECT-TYPE
    SYNTAX Dot11ApCivicLocationEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Civic Address location of the AP described with Civic Address Type ele-
        ments defined in IETF RFC-5139 [B42]."
    INDEX {dot11APCivicLocationIndex} ::= {dot11APCivicLocationTable 1}

Dot11ApCivicLocationEntry ::=
    SEQUENCE {
        dot11APCivicLocationIndex                          Unsigned32,
        dot11APCivicLocationCountry                        OCTET STRING,
        dot11APCivicLocationA1                             OCTET STRING,
        dot11APCivicLocationA2                             OCTET STRING,
        dot11APCivicLocationA3                             OCTET STRING,
        dot11APCivicLocationA4                             OCTET STRING,
        dot11APCivicLocationA5                             OCTET STRING,
        dot11APCivicLocationA6                             OCTET STRING,
        dot11APCivicLocationPrd                            OCTET STRING,
        dot11APCivicLocationPod                            OCTET STRING,
        dot11APCivicLocationsSts                           OCTET STRING,
        dot11APCivicLocationHno                            OCTET STRING,
        dot11APCivicLocationHns                            OCTET STRING,
        dot11APCivicLocationLmk                            OCTET STRING,
        dot11APCivicLocationLoc                            OCTET STRING,
        dot11APCivicLocationNam                            OCTET STRING,
        dot11APCivicLocationPc                             OCTET STRING,
        dot11APCivicLocationBld                            OCTET STRING,
        dot11APCivicLocationUnit                           OCTET STRING,
        dot11APCivicLocationFlr                            OCTET STRING,
        dot11APCivicLocationRoom                           OCTET STRING,
        dot11APCivicLocationPlc                            OCTET STRING,
        dot11APCivicLocationPcn                            OCTET STRING,
        dot11APCivicLocationPobox                          OCTET STRING,
        dot11APCivicLocationAddcode                        OCTET STRING,
        dot11APCivicLocationSeat                           OCTET STRING,
        dot11APCivicLocationRd                             OCTET STRING,
        dot11APCivicLocationRdsec                          OCTET STRING,
        dot11APCivicLocationRdbr                           OCTET STRING,
        dot11APCivicLocationRdsubbr                        OCTET STRING,
        dot11APCivicLocationPrm                            OCTET STRING,
        dot11APCivicLocationPom                            OCTET STRING
        }
dot11APCivicLocationIndex OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        Index for APCivicLocation elements in dot11APCivicLocationTable, greater
        than 0."
    ::= { dot11APCivicLocationEntry 1 }
```

```
dot11APCivicLocationCountry OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(0..255))
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.

      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      This attribute contains the two uppercase characters which correspond to
      the alpha-2 codes in ISO 3166-1. Example: US."
   ::= { dot11APCivicLocationEntry 2 }

dot11APCivicLocationA1 OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(0..255))
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.

      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      This attribute contains the national subdivisions (state, Region, prov-
      ince, prefecture). Example: California. The A1 element is used for the top
      level subdivision within a country. In the absence of a country-specific
      guide on how to use the A-series of elements, the second part of the ISO
      3166-2 code [ISO.3166-2] for a country subdivision SHOULD be used. The ISO
      3166-2 code is a formed of a country code and hyphen plus a code of one,
      two or three characters or numerals. For the A1 element, the leading coun-
      try code and hyphen are omitted and only the subdivision code is included.

      For example, the codes for Canada include CA-BC, CA-ON, CA-QC;Luxembourg
      has just three single character codes: LU-D, LU-G And LU-L; Australia uses
      both two and three character codes: AU-ACT, AU-NSW, AU-NT; France uses
      numerical codes for mainland France and letters for territories: FR-75,
      FR-NC."
   ::= { dot11APCivicLocationEntry 3 }

dot11APCivicLocationA2 OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(0..255))
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.

      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      This attribute contains the county, parish, gun (JP), district (IN). Exam-
      ple: King’s County."
   ::= { dot11APCivicLocationEntry 4 }

dot11APCivicLocationA3 OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(0..255))
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.

      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.
      This attribute contains the city, township, shi (JP). Example: San Fran-
      cisco."
   ::= { dot11APCivicLocationEntry 5 }

dot11APCivicLocationA4 OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(0..255))
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.

      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      This attribute contains the city division, borough, city district, ward,
      chou (JP). Example: Manhattan."
   ::= { dot11APCivicLocationEntry 6 }

dot11APCivicLocationA5 OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(0..255))
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.

      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      This attribute contains the neighborhood, block. Example: Morningside
      Heights."
   ::= { dot11APCivicLocationEntry 7 }

dot11APCivicLocationA6 OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(0..255))
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.

      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      This attribute contains the street. Example: Broadway. The A6 element is
      retained for use in those countries that require this level of detail.
      Where A6 was previously used for street names in IETF RFC 5139 [B42], it
      will not be used, the RD element will be used for thorough fare data. How-
      ever, without additional information these fields will not be interchanged
      when converting between different Civic formats. Where Civic address
      information is obtained from another format, such as the DHCP form IETF
      RFC 4776 [B40], the A6 element will be copied directly from the source
      format."
   ::= { dot11APCivicLocationEntry 8 }

dot11APCivicLocationPrd OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(0..255))
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.

      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.
      This attribute contains the leading street direction. Example: NW."
   ::= { dot11APCivicLocationEntry 9 }

dot11APCivicLocationPod OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(0..255))
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.

      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      This attribute contains the trailing street suffix. Example: SW."
   ::= { dot11APCivicLocationEntry 10 }

dot11APCivicLocationSts OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(0..255))
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.

      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      This attribute contains the street suffix. Example: Avenue, 'Platz,
      Street'."
   ::= { dot11APCivicLocationEntry 11 }

dot11APCivicLocationHno OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(0..255))
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.

      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      This attribute contains the numeric part only of the
      House number. Example: 123."
   ::= { dot11APCivicLocationEntry 12 }

dot11APCivicLocationHns OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(0..255))
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.

      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      This attribute contains the house number suffix. Example: A, 1/2"
   ::= { dot11APCivicLocationEntry 13 }

dot11APCivicLocationLmk OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(0..255))
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      This attribute contains the landmark or vanity address. Example: Low
      Library."
   ::= { dot11APCivicLocationEntry 14 }

dot11APCivicLocationLoc OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(0..255))
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.

      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      This attribute contains additional location information. Example: Room
      543."
   ::= { dot11APCivicLocationEntry 15 }

dot11APCivicLocationNam OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(0..255))
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.

      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      This attribute contains the name (residence, business, or office occupant.
      Example: Joe’s Barbershop."
   ::= { dot11APCivicLocationEntry 16 }

dot11APCivicLocationPc OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(0..255))
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.

      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      This attribute contains the postal code. Example: 10027-0401."
   ::= { dot11APCivicLocationEntry 17 }

dot11APCivicLocationBld OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(0..255))
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.

      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      This attribute contains the building (structure). Example: Hope Theater."
   ::= { dot11APCivicLocationEntry 18 }

dot11APCivicLocationUnit OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(0..255))
   MAX-ACCESS read-write
```

```
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        This attribute contains the unit (apartment, suite). Example: 12a."
    ::= { dot11APCivicLocationEntry 19 }

dot11APCivicLocationFlr OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(0..255))
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        This attribute contains the floor number. Example: 5."
    ::= { dot11APCivicLocationEntry 20 }

dot11APCivicLocationRoom OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(0..255))
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        This attribute contains the room. Example: 450F."
    ::= { dot11APCivicLocationEntry 21 }

dot11APCivicLocationPlc OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(0..255))
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        This attribute contains the place type. Example: office."
    ::= { dot11APCivicLocationEntry 22 }

dot11APCivicLocationPcn OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(0..255))
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        This attribute contains the postal community name. Example: Leonia."
    ::= { dot11APCivicLocationEntry 23 }

dot11APCivicLocationPobox OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(0..255))
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        This attribute contains the post office box (P.O. Box). Example: U40."
    ::= { dot11APCivicLocationEntry 24 }

dot11APCivicLocationAddcode OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(0..255))
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        This attribute contains the additional code. Example: 13203000003."
    ::= { dot11APCivicLocationEntry 25 }

dot11APCivicLocationSeat OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(0..255))
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        This attribute contains the seat (desk, cubicle, workstation). Example: WS
        181."
    ::= { dot11APCivicLocationEntry 26 }

dot11APCivicLocationRd OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(0..255))
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        This attribute contains the primary road or street. Example: Broadway."
    ::= { dot11APCivicLocationEntry 27 }

dot11APCivicLocationRdsec OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(0..255))
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        This attribute contains the road section. Example: 14.In some countries a
        thoroughfare can be broken up into sections, and it is not uncommon for
        street numbers to be repeated between sections. A road section identifier
        is required to ensure that an address is unique. For example, West Alice
        Parade has 5 sections, each numbered from 1; unless the section is speci-
        fied 7 West Alice Parade could exist in 5 different places."
    ::= { dot11APCivicLocationEntry 28 }

dot11APCivicLocationRdbr OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(0..255))
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        This attribute contains the road branch. Example: 'Lane 7'. Minor streets
        can share the same name, so that they can only Be distinguished by the
        major thoroughfare with which they intersect. For example, both West Alice
        Parade, Section 3 and Bob Street could both be interested by a Carol Lane.
        This element is used to specify a road branch where the name of the branch
        does not uniquely identify the road. Road branches MAY also be used where
        a major thoroughfare is split into sections."
    ::= { dot11APCivicLocationEntry 29 }

dot11APCivicLocationRdsubbr OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(0..255))
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        This attribute contains the road sub-branch. Example: Alley 8."
    ::= { dot11APCivicLocationEntry 30 }

dot11APCivicLocationPrm OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(0..255))
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        This attribute contains the road premodifier. Example: Old."
    ::= { dot11APCivicLocationEntry 31 }

dot11APCivicLocationPom OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(0..255))
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        This attribute contains the road post-modifier. Example: Extended."
    ::= { dot11APCivicLocationEntry 32 }

-- **********************************************************************
-- * End of dot11APCivicLocation TABLE
-- **********************************************************************

-- **********************************************************************
-- * dot11RoamingConsortium TABLE
-- **********************************************************************
dot11RoamingConsortiumTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11RoamingConsortiumEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "This is a Table of OIs which are to be transmitted in an ANQP Roaming
      Consortium ANQP-element. Each table entry corresponds to a roaming consor-
      tium or single SSP. The first 3 entries in this table are transmitted in
      Beacon and Probe Response frames."
   ::= { dot11imt 5 }

dot11RoamingConsortiumEntry OBJECT-TYPE
   SYNTAX Dot11RoamingConsortiumEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "Each OI identifies a roaming consortium (group of SSPs with inter-SSP
      roaming agreement) or a single SSP. A non-AP STA in possession of security
      credentials for the SSPN(s) identified by the OI, should be able to suc-
      cessfully authenticate to this AP."
   INDEX { dot11RoamingConsortiumOI }
   ::= { dot11RoamingConsortiumTable 1 }

Dot11RoamingConsortiumEntry ::=
   SEQUENCE {
      dot11RoamingConsortiumOI OCTET STRING,
      dot11RoamingConsortiumRowStatus RowStatus
      }

dot11RoamingConsortiumOI OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(16))
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.

      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      This attribute contains the IEEE defined OI as defined in 8.4.1.31."
   ::= { dot11RoamingConsortiumEntry 1 }

dot11RoamingConsortiumRowStatus OBJECT-TYPE
   SYNTAX RowStatus
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This object represents the status column for a conceptual row in this
      table."
   ::= { dot11RoamingConsortiumEntry 2 }

-- **********************************************************************
-- * End of dot11RoamingConsortium TABLE
-- **********************************************************************

-- **********************************************************************
-- * dot11DomainName TABLE
-- **********************************************************************
```

```
dot11DomainNameTable   OBJECT-TYPE
   SYNTAX                 SEQUENCE OF Dot11DomainNameEntry
   MAX-ACCESS             not-accessible
   STATUS                 current
   DESCRIPTION
      "This is a table of Domain Names which form the Domain Name list in Access
      Network Query Protocol. The Domain Name list may be transmitted to a non-
      AP STA in a GAS Response. Each table entry corresponds to a single Domain
      Name."
   ::= { dot11imt 6 }

dot11DomainNameEntry OBJECT-TYPE
   SYNTAX     Dot11DomainNameEntry
   MAX-ACCESS not-accessible
   STATUS     current
   DESCRIPTION
      "This is a control variable.

      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      Each Domain Name identifies a domain names of the entity operating the
      IEEE 802.11 access network."
   INDEX { dot11DomainNameOui }
   ::= { dot11DomainNameTable 1 }

Dot11DomainNameEntry ::=
   SEQUENCE {
      dot11DomainName                                       OCTET STRING,
      dot11DomainNameRowStatus                              RowStatus,
      dot11DomainNameOui                                    OCTET STRING
      }

dot11DomainName OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(255))
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.

      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      This attribute contains a Domain Name of up to 255 octets formatted in
      accordance with the 'Preferred Name Syntax' as defined in IETF RFC 1035."
   ::= { dot11DomainNameEntry 1 }

dot11DomainNameRowStatus OBJECT-TYPE
   SYNTAX RowStatus
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This object represents the status column for a conceptual row in this
      table."
   ::= { dot11DomainNameEntry 2 }
dot11DomainNameOui OBJECT-TYPE
   SYNTAX OCTET STRING (SIZE(3..5))
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This object represents an organizationally unique identifier, used as an
      index into the Domain Name table."
   ::= { dot11DomainNameEntry 3 }
```

```
-- **********************************************************************
-- * End of dot11NameTable TABLE
-- **********************************************************************

-- **********************************************************************
-- * dot11GASAdvertisement TABLE
-- **********************************************************************
dot11GASAdvertisementTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11GASAdvertisementEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "This object is a table of GAS counters that allows for multiple instanti-
      ations of those counters on an STA."
   ::= { dot11imt 7 }

dot11GASAdvertisementEntry OBJECT-TYPE
   SYNTAX Dot11GASAdvertisementEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "This object provides the attributes identifying a GAS counter within an
      STA."
   INDEX { dot11GASAdvertisementId }
   ::= { dot11GASAdvertisementTable 1 }

Dot11GASAdvertisementEntry ::=
   SEQUENCE{
      dot11GASAdvertisementId                              Unsigned32,
      dot11GASPauseForServerResponse                       TruthValue,
      dot11GASResponseTimeout                              Unsigned32,
      dot11GASComebackDelay                                Unsigned32,
      dot11GASResponseBufferingTime                        Unsigned32,
      dot11GASQueryResponseLengthLimit                     Unsigned32,
      dot11GASQueries                                      Counter32,
      dot11GASQueryRate                                    Gauge32,
      dot11GASResponses                                    Counter32,
      dot11GASResponseRate                                 Gauge32,
      dot11GASTransmittedFragmentCount                     Counter32,
      dot11GASReceivedFragmentCount                        Counter32,
      dot11GASNoRequestOutstanding                         Counter32,
      dot11GASResponsesDiscarded                           Counter32,
      dot11GASFailedResponses                              Counter32
      }

dot11GASAdvertisementId OBJECT-TYPE
   SYNTAX Unsigned32 (0..255)
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "This is a control variable.

      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      The 1 octet identification number for the GAS Advertisement Protocol, as
      defined in Table 8-175, for which statistics are stored the logical row of
      the GASAdvertisement table."
   ::= { dot11GASAdvertisementEntry 1 }

dot11GASPauseForServerResponse OBJECT-TYPE
   SYNTAX TruthValue
```

```
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        This attribute is only used by the responding STA in a GAS exchange. When
        true, it indicates that the responding STA will not transmit a GAS Initial
        Response frame until it receives the query response from the Advertisement
        Server or a timeout occurs. When false, the STA will not wait for a
        response from the Advertisement Server before transmiting the GAS Initial
        Response frame. The setting of this MIB object is outside the scope of
        this standard."
    ::= { dot11GASAdvertisementEntry 2 }

dot11GASResponseTimeout OBJECT-TYPE
    SYNTAX Unsigned32 (1000..65535)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        This parameter shall indicate the GAS response timeout value in TUs."
    DEFVAL {5000}
    ::= { dot11GASAdvertisementEntry 3 }

dot11GASComebackDelay OBJECT-TYPE
    SYNTAX Unsigned32 (0..65535)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        This object identifies the GAS Comeback Delay (in TUs) to be used for this
        Advertisement Protocol"
    DEFVAL {1000}
    ::= { dot11GASAdvertisementEntry 4 }

dot11GASResponseBufferingTime OBJECT-TYPE
    SYNTAX Unsigned32 (0..65535)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        This object defines the time duration after the expiry of the GAS Comeback
        Delay that an STA will buffer a Query Response. The units of this MIB
        object are TUs. Upon expiry of this time, the STA may discard the Query
        Response."
    DEFVAL {1000}

    ::= { dot11GASAdvertisementEntry 5 }
```

```
dot11GASQueryResponseLengthLimit OBJECT-TYPE
   SYNTAX Unsigned32 (1..127)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.

      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      This object indicates the maximum number of octets an AP will transmit in
      one or more Query Response fields contained within GAS Comeback Response
      frame(s). A value of 127 means the maximum limit enforced is  contained by
      the maximum allowable number of fragments in the GAS Query  Fragment
      Response ID"
   ::= { dot11GASAdvertisementEntry 6 }

dot11GASQueries OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.

      It is written by the SME after transmission of a MLME-GAS.request or
      receipt of an MLME-GAS.indication primitive.

      The number of GAS queries sent or received for the protocol identified by
      dot11GASAdvertisementId."
   ::= { dot11GASAdvertisementEntry 7 }

dot11GASQueryRate OBJECT-TYPE
   SYNTAX Gauge32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.

      It is updated by the SME after receipt of an MLME-GAS.indication primi-
      tive.

      The number of GAS queries per minute received for the protocol identified
      by dot11GASAdvertisementId, averaged over the previous ten minutes."
   ::= { dot11GASAdvertisementEntry 8 }

dot11GASResponses OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.

      It is written by the SME after transmission of a MLME-GAS.response or
      receipt of an MLME-GAS.confirm primitive.

      The number of GAS responses sent or received for the protocol identified
      by dot11GASAdvertisementId."
   ::= { dot11GASAdvertisementEntry 9 }

dot11GASResponseRate OBJECT-TYPE
   SYNTAX Gauge32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
```

```
      "This is a status variable.

      It is updated by the SME after transmission of an MLME-GAS.response prim-
      itive.

      The number of responses to GAS queries per minute transmitted by an AP for
      the protocol identified by dot11GASAdvertisementId, averaged over the pre-
      vious ten minutes. This MIB variable is not used in non-AP STAs."
   ::= { dot11GASAdvertisementEntry 10 }

dot11GASTransmittedFragmentCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.

      It is updated by the SME after transmission of an MLME-GAS.response prim-
      itive.

      This counter shall be incremented for an acknowledged GAS MMPDU for the
      protocol identified by dot11GASAdvertisementId."
   ::= { dot11GASAdvertisementEntry 11 }

dot11GASReceivedFragmentCount OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.

      It is updated by the MAC after transmission of an MLME-GAS.confirm primi-
      tive.

      This counter shall be incremented for each successfully received MMPDU of
      type Data"
   ::= { dot11GASAdvertisementEntry 12 }

dot11GASNoRequestOutstanding OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.

      It is updated by the SME after transmission of an MLME-GAS.response prim-
      itive.

      This counter shall be incremented each time a STA returns a status code of
      no request outstanding in a GAS Initial Response or GAS Comeback Response
      frame for the protocol identified by dot11GASAdvertisementId."
   ::= { dot11GASAdvertisementEntry 13 }

dot11GASResponsesDiscarded OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.

      It is updated by the SME after transmission of an MLME-GAS.response prim-
      itive.

      This counter shall be incremented each a STA discards a GAS response due
      to the expiry of the dot11GASResponseBufferingTime timer for the protocol
      identified by dot11GASAdvertisementId."
   ::= { dot11GASAdvertisementEntry 14 }

dot11GASFailedResponses OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.

      It is updated by the SME after transmission of an MLME-GAS.response prim-
      itive.

      This counter shall be incremented each time a STA commences transmitting a
      GAS response but fails to successfully complete the transmission of all
      GAS fragments in that response due to the expiry of the
      dot11GASResponseTimeout timer for the protocol identified by
      dot11GASAdvertisementId."
   ::= { dot11GASAdvertisementEntry 15 }

-- **********************************************************************
-- * End of dot11GASAdvertisement TABLE
-- **********************************************************************


-- **********************************************************************
-- * MAC State Generic Convergence
-- **********************************************************************

-- MAC State Generic Convergence Function attributes
   -- DEFINED AS "The MAC state generic convergence function object
   -- class provides the necessary support for support of event-driven
   -- triggers to higher layer protocols and the capabilities to
   -- support those triggers."

dot11MSGCF OBJECT IDENTIFIER ::= { ieee802dot11 7}

      -- MAC State GROUPS
      -- dot11MACStateConfigTable ::= { dot11MSGCF 1 }
      -- dot11MACStateParameterTable ::= { dot11MSGCF 2 }
      -- dot11MACStateESSLinkTable ::= { dot11MSGCF 3 }

-- *********************************************************************
-- * dot11ESSLinkIdentifier type definition
-- *********************************************************************
Dot11ESSLinkIdentifier ::= OCTET STRING (SIZE(0..38))
   -- This object type holds the identifier for an 802.11
   -- network. It is composed of the SSID string concatenated
   -- with the HESSID, if present.

-- *********************************************************************
-- * dot11MACStateConfig TABLE
-- *********************************************************************
dot11MACStateConfigTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11MACStateConfigEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "This table holds configuration parameters for the 802.11 MAC
      State Convergence Function."
   ::= { dot11MSGCF 1 }

dot11MACStateConfigEntry OBJECT-TYPE
```

```
    SYNTAX Dot11MACStateConfigEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Each entry represents a conceptual row in the dot11MACStateConfigTable
        and provides information about network configuration parameters used in
        the MAC State Generic Convergence Function."
    INDEX { dot11MSCEESSLinkIdentifier, dot11MSCENonAPStationMacAddress }
    ::= { dot11MACStateConfigTable 1 }

Dot11MACStateConfigEntry ::=
    SEQUENCE {
        dot11ESSDisconnectFilterInterval Unsigned32,
        dot11ESSLinkDetectionHoldInterval Unsigned32,
        dot11MSCEESSLinkIdentifier Dot11ESSLinkIdentifier,
        dot11MSCENonAPStationMacAddress MacAddress
        }

dot11ESSDisconnectFilterInterval OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        This attribute is set to the number of time units (TUs) that will elapse
        after an MLME-DISASSOCIATE.confirm or MLME-DEAUTHENTICATE.confirm primi-
        tive without a subsequent association before the link is declared down.
        This interval is intended to allow a non-AP STA time to transition to
        another AP within the same ESS before declaring that the link to the ESS
        is lost."
    ::= { dot11MACStateConfigEntry 1 }

dot11ESSLinkDetectionHoldInterval OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        This attribute is set to the number of time units (TUs) that an ESS is
        held in the dot11MACStateESSLink table after its last observation before
        purging the entry from the table."
    ::= { dot11MACStateConfigEntry 2 }

dot11MSCEESSLinkIdentifier OBJECT-TYPE
    SYNTAX Dot11ESSLinkIdentifier
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "This is an auxiliary variable used to identify instances of the columnar
        objects in the dot11MACStateConfigTable table."

    ::= { dot11MACStateConfigEntry 3 }

dot11MSCENonAPStationMacAddress OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS not-accessible
```

```
   STATUS current
   DESCRIPTION
      "This is an auxiliary variable used to identify instances of the columnar
      objects in the dot11MACStateConfigTable table."

   ::= { dot11MACStateConfigEntry 4 }

-- ********************************************************************
-- * End of dot11MACStateConfig TABLE
-- ********************************************************************

-- ********************************************************************
-- * dot11MACStateParameter TABLE
-- ********************************************************************

dot11MACStateParameterTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11MACStateParameterEntry
   MAX-ACCESS not-accessible
   STATUS     current
   DESCRIPTION
      "This table holds the current parameters used for each 802.11 network for
      802.11 MAC convergence functions."
   ::= { dot11MSGCF 2 }

dot11MACStateParameterEntry OBJECT-TYPE
   SYNTAX         Dot11MACStateParameterEntry
   MAX-ACCESS     not-accessible
   STATUS         current
   DESCRIPTION
      "Each entry represents a conceptual row in the dot11MACStateParameterTable
      and provides information about link configuration parameters used in the
      MAC State Generic Convergence Function."
   INDEX { dot11MSPEESSLinkIdentifier, dot11MSPENonAPStationMacAddress }
   ::= { dot11MACStateParameterTable 1 }

Dot11MACStateParameterEntry ::=
   SEQUENCE {
      dot11ESSLinkDownTimeInterval                          Unsigned32,
      dot11ESSLinkRssiDataThreshold                         Unsigned32,
      dot11ESSLinkRssiBeaconThreshold                       Unsigned32,
      dot11ESSLinkDataSnrThreshold                          Unsigned32,
      dot11ESSLinkBeaconSnrThreshold                        Unsigned32,
      dot11ESSLinkBeaconFrameErrorRateThresholdInteger      Unsigned32,
      dot11ESSLinkBeaconFrameErrorRateThresholdFraction     Unsigned32,
      dot11ESSLinkBeaconFrameErrorRateThresholdExponent     Unsigned32,
      dot11ESSLinkFrameErrorRateThresholdInteger            Unsigned32,
      dot11ESSLinkFrameErrorRateThresholdFraction           Unsigned32,
      dot11ESSLinkFrameErrorRateThresholdExponent           Unsigned32,
      dot11PeakOperationalRate                              Unsigned32,
      dot11MinimumOperationalRate                           Unsigned32,
      dot11ESSLinkDataThroughputInteger                     Unsigned32,
      dot11ESSLinkDataThroughputFraction                    Unsigned32,
      dot11ESSLinkDataThroughputExponent                    Unsigned32,
      dot11MSPEESSLinkIdentifier                Dot11ESSLinkIdentifier,
      dot11MSPENonAPStationMacAddress                       MacAddress
      }


dot11ESSLinkDownTimeInterval OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.
      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      This attribute defines the desired time interval that the MAC State
      Generic convergence function will attempt to predict the failure of an
      802.11 network in time units (TUs). The convergence function should issue
      predicted network failure events at least this time interval before the
      network failure is detected."
   ::= { dot11MACStateParameterEntry 2 }

dot11ESSLinkRssiDataThreshold OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.

      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      This attribute defines the threshold value for RSSI on Data frames. When
      the RSSI drops below this threshold, a report is issued."
   ::= { dot11MACStateParameterEntry 3 }

dot11ESSLinkRssiBeaconThreshold OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.

      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      This attribute defines the threshold value for RSSI on Beacon frames. When
      the RSSI drops below this threshold, a report is issued."
   ::= { dot11MACStateParameterEntry 4 }

dot11ESSLinkBeaconSnrThreshold OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.

      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      This attribute defines the threshold value for SNR on received Beacon
      frames. When the SNR drops below this threshold, a report is issued"
   ::= { dot11MACStateParameterEntry 5 }

dot11ESSLinkDataSnrThreshold OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.

      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      This attribute defines the threshold value for SNR on received Data
      frames. When the SNR drops below this threshold, a report is issued."
   ::= { dot11MACStateParameterEntry 6 }

dot11ESSLinkBeaconFrameErrorRateThresholdInteger OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.

      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      The Beacon frame error rate is stored in scientific notation as a signif-
      icant and exponent. This attribute contains the integer value of the sig-
      nificand."
   ::= { dot11MACStateParameterEntry 7 }

dot11ESSLinkBeaconFrameErrorRateThresholdFraction OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.

      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      The Beacon frame error rate is stored in scientific notation as a signif-
      icant and exponent. This attribute contains the fractional value of the
      significand."
   ::= { dot11MACStateParameterEntry 8 }

dot11ESSLinkBeaconFrameErrorRateThresholdExponent OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.

      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      The Beacon frame error rate is stored in scientific notation as a signif-
      icant and exponent. This attribute contains the integer value of the expo-
      nent."
   ::= { dot11MACStateParameterEntry 9 }

dot11ESSLinkFrameErrorRateThresholdInteger OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.

      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      The frame error rate of the network is stored in scientific notation as a
      significant and exponent. This attribute contains the integer value of the
      significand."
   ::= { dot11MACStateParameterEntry 10 }

dot11ESSLinkFrameErrorRateThresholdFraction OBJECT-TYPE
```

```
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        The frame error rate of the network is stored in scientific notation as a
        significant and exponent. This attribute contains the fractional value of
        the significand."
    ::= { dot11MACStateParameterEntry 11 }

dot11ESSLinkFrameErrorRateThresholdExponent OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        The frame error rate of the network is stored in scientific notation as a
        significant and exponent. This attribute contains the integer value of the
        exponent."
    ::= { dot11MACStateParameterEntry 12 }

dot11PeakOperationalRate OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        The highest operational rate used for transmission of data frames, encoded
        as defined in 8.4.2.3."
    ::= { dot11MACStateParameterEntry 13 }

dot11MinimumOperationalRate OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.

        It is written by an external management entity or the SME.
        Changes take effect as soon as practical in the implementation.

        The lowest operational rate used for transmission of data frames, encoded
        as defined in 8.4.2.3."
    ::= { dot11MACStateParameterEntry 14 }

dot11ESSLinkDataThroughputInteger OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is a control variable.
```

```
      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      The data throughput rate is of the network is stored in scientific nota-
      tion as a significant and exponent. This attribute contains the integer
      value of the significand."
   ::= { dot11MACStateParameterEntry 15 }

dot11ESSLinkDataThroughputFraction OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.

      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      The data throughput rate is of the network is stored in scientific nota-
      tion as a significant and exponent. This attribute contains the fractional
      value of the significand."
   ::= { dot11MACStateParameterEntry 16 }

dot11ESSLinkDataThroughputExponent OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      "This is a control variable.

      It is written by an external management entity or the SME.
      Changes take effect as soon as practical in the implementation.

      The data throughput rate is of the network is stored in scientific nota-
      tion as a significant and exponent. This attribute contains the integer
      value of the exponent."
   ::= { dot11MACStateParameterEntry 17 }

dot11MSPEESSLinkIdentifier OBJECT-TYPE
   SYNTAX Dot11ESSLinkIdentifier
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "This is an auxiliary variable used to identify instances of the columnar
      objects in the dot11MACStateParameterTable table."

   ::= { dot11MACStateParameterEntry 18 }

dot11MSPENonAPStationMacAddress OBJECT-TYPE
   SYNTAX MacAddress
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "This is an auxiliary variable used to identify instances of the columnar
      objects in the dot11MACStateParameterTable table."

   ::= { dot11MACStateParameterEntry 19 }


-- **********************************************************************
-- * End of dot11MACStateParameter TABLE
-- **********************************************************************

-- **********************************************************************
-- * dot11MACStateESSLink TABLE
-- **********************************************************************
dot11MACStateESSLinkDetectedTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Dot11MACStateESSLinkDetectedEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "This table holds the detected 802.11 network list used for MAC conver-
      gence functions."
   ::= { dot11MSGCF 3 }

dot11MACStateESSLinkDetectedEntry OBJECT-TYPE
   SYNTAX Dot11MACStateESSLinkDetectedEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "Each entry represents a conceptual row in the dot11MACStateESSLinkTable
      and provides information about available networks for use in the MAC State
      Generic Convergence Function."
   INDEX { dot11MSELDEESSLinkIdentifier, dot11MSELDENonAPStationMacAddress }
   ::= { dot11MACStateESSLinkDetectedTable 1 }

Dot11MACStateESSLinkDetectedEntry ::=
   SEQUENCE {
      dot11ESSLinkDetectedIndex                      Unsigned32,
      dot11ESSLinkDetectedNetworkId                  OCTET STRING,
      dot11ESSLinkDetectedNetworkDetectTime          Unsigned32,
      dot11ESSLinkDetectedNetworkModifiedTime        Unsigned32,
      dot11ESSLinkDetectedNetworkMIHCapabilities     BITS,
      dot11MSELDEESSLinkIdentifier                   Dot11ESSLinkIdentifier,
      dot11MSELDENonAPStationMacAddress              MacAddress
      }

dot11ESSLinkDetectedIndex OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "Index for ESSLinkDetected elements in dot11ESSLinkDetectedTable, greater
      than 0."
   ::= { dot11MACStateESSLinkDetectedEntry 1 }

dot11ESSLinkDetectedNetworkId OBJECT-TYPE
   SYNTAX OCTET STRING
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.

      It is written by the MSGCF after reception of a MSGCF-ESS-Link-
      Detected.indication primitive.

      The string used to identify the network represented by this row in the
      table. It is composed of the SSID of the network concatenated with the
      HESSID, if present."
   ::= { dot11MACStateESSLinkDetectedEntry 2 }


dot11ESSLinkDetectedNetworkDetectTime OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This is a status variable.
```

```
       It is written by the MSGCF after reception of a MSGCF-ESS-Link-
       Detected.indication primitive.

       The STA's TSF timer when any BSSID supporting the network was first
       detected."
    ::= { dot11MACStateESSLinkDetectedEntry 4 }

dot11ESSLinkDetectedNetworkModifiedTime OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
       "This is a status variable.

       It is written by the MSGCF after reception of a MSGCF-ESS-Link-
       Detected.indication primitive.

       The STA's TSF timer value when changes were made to any part of this row
       in the table, such as by addition of a BSSID to the BSSID list."
    ::= { dot11MACStateESSLinkDetectedEntry 5 }

dot11ESSLinkDetectedNetworkMIHCapabilities OBJECT-TYPE
    SYNTAX BITS {
       mihIsSupport(0),
       mihCsEsSupport(1)
    }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
       "This is a status variable.

       It is written by the MSGCF after reception of a MSGCF-ESS-Link-
       Detected.indication primitive.

       The object reports whether the network supports IEEE 802.21 MIH informa-
       tion services and/or IEEE 802.21 MIH command/event services. These values
       are determined by examining the Interworking information in frames that
       caused the network to be detected."
    ::= { dot11MACStateESSLinkDetectedEntry 6 }

dot11MSELDEESSLinkIdentifier OBJECT-TYPE
    SYNTAX Dot11ESSLinkIdentifier
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
       "This is an auxiliary variable used to identify instances of the columnar
       objects in the dot11MACStateESSLinkDetectedTable table."

    ::= { dot11MACStateESSLinkDetectedEntry 7 }

dot11MSELDENonAPStationMacAddress OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
       "This is an auxiliary variable used to identify instances of the columnar
       objects in the dot11MACStateESSLinkDetectedTable table."

    ::= { dot11MACStateESSLinkDetectedEntry 8 }


-- **********************************************************************
-- * End of dot11MACStateESSLink TABLE
-- **********************************************************************
-- **********************************************************************
-- * Conformance Information
-- **********************************************************************

dot11Conformance OBJECT IDENTIFIER ::= { ieee802dot11 5 }
dot11Groups      OBJECT IDENTIFIER ::= { dot11Conformance 1 }
dot11Compliances OBJECT IDENTIFIER ::= { dot11Conformance 2 }

-- **********************************************************************
-- *   Groups - units of conformance
-- **********************************************************************

dot11SMTbase OBJECT-GROUP
   OBJECTS {
      dot11StationID,
      dot11MediumOccupancyLimit,
      dot11CFPollable,
      dot11CFPPeriod,
      dot11CFPMaxDuration,
      dot11AuthenticationResponseTimeOut,
      dot11PrivacyOptionImplemented,
      dot11PowerManagementMode,
      dot11DesiredSSID, dot11DesiredBSSType,
      dot11OperationalRateSet,
      dot11BeaconPeriod, dot11DTIMPeriod,
      dot11AssociationResponseTimeOut }
   STATUS deprecated
   DESCRIPTION
      "Superseded by dot11SMTbase2.
      The SMT object class provides the necessary support at the STA to manage
      the processes in the STA such that the STA may work cooperatively as a
      part of an IEEE 802.11 network."
   ::= { dot11Groups 1 }

dot11SMTprivacy OBJECT-GROUP
   OBJECTS {
      dot11PrivacyInvoked,
      dot11WEPKeyMappingLengthImplemented,
      dot11ExcludeUnencrypted,
      dot11WEPICVErrorCount,
      dot11WEPExcludedCount,
      dot11WEPDefaultKeyID,
      dot11WEPDefaultKeyValue,
      dot11WEPKeyMappingWEPOn,
      dot11WEPKeyMappingValue,
      dot11WEPKeyMappingAddress,
      dot11WEPKeyMappingStatus }
   STATUS current
   DESCRIPTION
      "The SMTPrivacy package is a set of attributes that are present if WEP is
      implemented in the STA."
   ::= { dot11Groups 2 }

dot11MACbase OBJECT-GROUP
   OBJECTS {
      dot11MACAddress,
      dot11Address,
      dot11GroupAddressesStatus,
      dot11RTSThreshold,
      dot11ShortRetryLimit,
      dot11LongRetryLimit,
      dot11FragmentationThreshold,
      dot11MaxTransmitMSDULifetime,
      dot11MaxReceiveLifetime,
      dot11ManufacturerID,
      dot11ProductID }
   STATUS deprecated
   DESCRIPTION
      "Superseded by dot11MACbase2.
      The MAC object class provides the necessary support for the access con-
      trol, generation, and verification of frame check sequences (FCSs), and
      proper delivery of valid data to upper layers."
   ::= { dot11Groups 3 }

dot11MACStatistics OBJECT-GROUP
   OBJECTS {
      dot11RetryCount,
      dot11MultipleRetryCount,
      dot11RTSSuccessCount,
      dot11RTSFailureCount,
      dot11ACKFailureCount,
      dot11FrameDuplicateCount }
   STATUS current
   DESCRIPTION
      "The MACStatistics package provides extended statistical information on
      the operation of the MAC. This package is completely optional."
   ::= { dot11Groups 4 }

dot11ResourceTypeID OBJECT-GROUP
   OBJECTS {
      dot11ResourceTypeIDName,
      dot11manufacturerOUI,
      dot11manufacturerName,
      dot11manufacturerProductName,
      dot11manufacturerProductVersion }
   STATUS current
   DESCRIPTION
      "Attributes used to identify a STA, its manufacturer, and various product
      names and versions."
   ::= { dot11Groups 5 }

dot11SmtAuthenticationAlgorithms OBJECT-GROUP
   OBJECTS {
      dot11AuthenticationAlgorithm,
      dot11AuthenticationAlgorithmsActivated }
   STATUS current
   DESCRIPTION
      "Authentication Algorithm Table."
   ::= { dot11Groups 6 }

dot11PhyOperationComplianceGroup OBJECT-GROUP
   OBJECTS { dot11PHYType, dot11CurrentRegDomain, dot11TempType }
   STATUS deprecated
          DESCRIPTION
      "Superseded by dot11PhyOperationComplianceGroup2.
      PHY layer operations attributes."
   ::= { dot11Groups 7 }

dot11PhyAntennaComplianceGroup OBJECT-GROUP
   OBJECTS {
      dot11CurrentTxAntenna,
      dot11DiversitySupportImplemented,
      dot11CurrentRxAntenna }
   STATUS deprecated
   DESCRIPTION
      "Attributes for Data Rates for IEEE 802.11."
   ::= { dot11Groups 8 }
```

```
dot11PhyTxPowerComplianceGroup OBJECT-GROUP
   OBJECTS {
      dot11NumberSupportedPowerLevelsImplemented,
      dot11TxPowerLevel1,
      dot11TxPowerLevel2,
      dot11TxPowerLevel3,
      dot11TxPowerLevel4,
      dot11TxPowerLevel5,
      dot11TxPowerLevel6,
      dot11TxPowerLevel7,
      dot11TxPowerLevel8,
      dot11CurrentTxPowerLevel }
   STATUS current
   DESCRIPTION
      "Attributes for Control and Management of transmit power."
   ::= { dot11Groups 9 }

dot11PhyFHSSComplianceGroup OBJECT-GROUP
   OBJECTS {
      dot11HopTime,
      dot11CurrentChannelNumber,
      dot11MaxDwellTime,
      dot11CurrentDwellTime,
      dot11CurrentSet,
      dot11CurrentPattern,
      dot11CurrentIndex }
   STATUS deprecated
   DESCRIPTION
      "Superseded by dot11PhyFHSSComplianceGroup2.
      Attributes that configure the Frequency Hopping for IEEE 802.11."
   ::= { dot11Groups 10 }

dot11PhyDSSSComplianceGroup OBJECT-GROUP
   OBJECTS {
      dot11CurrentChannel,
      dot11CCAModeSupported,
      dot11CurrentCCAMode,
      dot11EDThreshold }
   STATUS current
   DESCRIPTION
      "Attributes that configure the DSSS for IEEE 802.11."
   ::= { dot11Groups 11 }

dot11PhyIRComplianceGroup OBJECT-GROUP
   OBJECTS {
      dot11CCAWatchdogTimerMax,
      dot11CCAWatchdogCountMax,
      dot11CCAWatchdogTimerMin,
      dot11CCAWatchdogCountMin }
   STATUS current
   DESCRIPTION
      "Attributes that configure the baseband IR for IEEE 802.11."
   ::= { dot11Groups 12 }

dot11PhyRegDomainsSupportGroup OBJECT-GROUP
   OBJECTS { dot11RegDomainsImplementedValue  }
   STATUS deprecated
   DESCRIPTION
      "Attributes that specify the supported Regulation Domains."
   ::= { dot11Groups 13 }

dot11PhyAntennasListGroup OBJECT-GROUP
   OBJECTS {
        dot11TxAntennaImplemented,
        dot11RxAntennaImplemented,
        dot11DiversitySelectionRxImplemented }
    STATUS current
    DESCRIPTION
        "Attributes that specify the supported Regulation Domains."
    ::= { dot11Groups 14 }

dot11PhyRateGroup OBJECT-GROUP
    OBJECTS {
        dot11ImplementedDataRatesTxValue,
        dot11ImplementedDataRatesRxValue }
    STATUS current
    DESCRIPTION
        "Attributes for Data Rates for IEEE 802.11."
    ::= { dot11Groups 15 }

dot11CountersGroup OBJECT-GROUP
    OBJECTS {
        dot11TransmittedFragmentCount,
        dot11GroupTransmittedFrameCount,
        dot11FailedCount,
        dot11ReceivedFragmentCount,
        dot11GroupReceivedFrameCount,
        dot11FCSErrorCount,
        dot11WEPUndecryptableCount,
        dot11TransmittedFrameCount }
    STATUS deprecated
    DESCRIPTION
        "Superseded by dot11CountersGroup2.
        Attributes from the dot11CountersGroup that are not described in the
        dot11MACStatistics group. These objects are mandatory."
    ::= { dot11Groups 16 }

dot11NotificationGroup NOTIFICATION-GROUP
    NOTIFICATIONS {
        dot11Disassociate,
        dot11Deauthenticate,
        dot11AuthenticateFail,
        dot11Associate,
        dot11AssociateFailed,
        dot11Reassociate,
        dot11ReassociateFailed   }
    STATUS current
    DESCRIPTION
        "IEEE 802.11 notifications"
    ::= { dot11Groups 17 }

dot11SMTbase2 OBJECT-GROUP
    OBJECTS {
        dot11MediumOccupancyLimit,
        dot11CFPollable,
        dot11CFPPeriod,
        dot11CFPMaxDuration,
        dot11AuthenticationResponseTimeOut,
        dot11PrivacyOptionImplemented,
        dot11PowerManagementMode,
        dot11DesiredSSID,
        dot11DesiredBSSType,
        dot11OperationalRateSet,
        dot11BeaconPeriod,
        dot11DTIMPeriod,
        dot11AssociationResponseTimeOut,
        dot11DisassociateReason,
        dot11DisassociateStation,
        dot11DeauthenticateReason,
        dot11DeauthenticateStation,
        dot11AuthenticateFailStatus,
        dot11AuthenticateFailStation }
    STATUS deprecated
    DESCRIPTION
        "Superseded by dot11SMTbase3.
        The SMTbase2 object class provides the necessary support at the STA to
        manage the processes in the STA such that the STA may work cooperatively
        as a part of an IEEE 802.11 network."
    ::= { dot11Groups 18 }

dot11PhyOFDMComplianceGroup OBJECT-GROUP
    OBJECTS {
        dot11CurrentFrequency,
        dot11TIThreshold,
        dot11FrequencyBandsImplemented,
        dot11ChannelStartingFactor }
    STATUS deprecated
    DESCRIPTION
        "Superseded by dot11PhyOFDMComplianceGroup2.
        Attributes that configure the OFDM for IEEE 802.11."
    ::= { dot11Groups 19 }

dot11SMTbase3 OBJECT-GROUP
    OBJECTS {
        dot11MediumOccupancyLimit,
        dot11CFPollable,
        dot11CFPPeriod,
        dot11CFPMaxDuration,
        dot11AuthenticationResponseTimeOut,
        dot11PrivacyOptionImplemented,
        dot11PowerManagementMode,
        dot11DesiredSSID,
        dot11DesiredBSSType,
        dot11OperationalRateSet,
        dot11BeaconPeriod,
        dot11DTIMPeriod,
        dot11AssociationResponseTimeOut,
        dot11DisassociateReason,
        dot11DisassociateStation,
        dot11DeauthenticateReason,
        dot11DeauthenticateStation,
        dot11AuthenticateFailStatus,
        dot11AuthenticateFailStation,
        dot11MultiDomainCapabilityImplemented,
        dot11MultiDomainCapabilityActivated,
        dot11CountryString }
    STATUS deprecated
    DESCRIPTION
        "Superseded by dot11SMTbase4.
        The SMTbase3 object class provides the necessary support at the STA to
        manage the processes in the STA such that the STA may work cooperatively
        as a part of an IEEE 802.11 network, when the STA is capable of multi-
        domain operation. This object group should be implemented when the multi-
        domain capability option is implemented."
    ::= { dot11Groups 20 }

dot11MultiDomainCapabilityGroup OBJECT-GROUP
    OBJECTS {
        dot11FirstChannelNumber,
        dot11NumberofChannels,
        dot11MaximumTransmitPowerLevel }
    STATUS current
    DESCRIPTION
        "The dot11MultiDomainCapabilityGroup object class provides the objects
        necessary to manage the channels usable by a STA, when the multi-domain
        capability option is implemented."
    ::= { dot11Groups 21 }

dot11PhyFHSSComplianceGroup2 OBJECT-GROUP
    OBJECTS {
        dot11HopTime,
        dot11CurrentChannelNumber,
        dot11MaxDwellTime,
        dot11CurrentDwellTime,
        dot11CurrentSet,
        dot11CurrentPattern,
        dot11CurrentIndex,
        dot11EHCCPrimeRadix,
        dot11EHCCNumberofChannelsFamilyIndex,
        dot11EHCCCapabilityImplemented,
        dot11EHCCCapabilityActivated,
        dot11HopAlgorithmAdopted,
        dot11RandomTableFlag,
        dot11NumberofHoppingSets,
        dot11HopModulus,
        dot11HopOffset,
        dot11RandomTableFieldNumber }
    STATUS current
    DESCRIPTION
        "Attributes that configure the Frequency Hopping for IEEE 802.11 when
        multi-domain capability option is implemented."
    ::= { dot11Groups 22 }

dot11PhyHRDSSSComplianceGroup OBJECT-GROUP
    OBJECTS {
        dot11CurrentChannel,
        dot11CCAModeSupported,
        dot11CurrentCCAMode,
        dot11EDThreshold,
        dot11ShortPreambleOptionImplemented,
        dot11PBCCOptionImplemented,
        dot11ChannelAgilityPresent,
        dot11ChannelAgilityActivated,
        dot11HRCCAModeImplemented }
    STATUS current
    DESCRIPTION
        "Attributes that configure the HRDSSS for IEEE 802.11."
    ::= { dot11Groups 23 }

dot11PhyERPComplianceGroup OBJECT-GROUP
    OBJECTS { dot11CurrentChannel,
        dot11ShortPreambleOptionImplemented,
        dot11ChannelAgilityPresent,
        dot11ChannelAgilityActivated,
        dot11DSSSOFDMOptionImplemented,
        dot11DSSSOFDMOptionActivated,
        dot11PBCCOptionImplemented,
        dot11ERPPBCCOptionImplemented,
        dot11ERPBCCOptionActivated,
        dot11ShortSlotTimeOptionImplemented,
        dot11ShortSlotTimeOptionActivated }
    STATUS current
    DESCRIPTION
        "Attributes that configure the ERP."
    ::= { dot11Groups 24 }
```

```
dot11RSNAadditions OBJECT-GROUP
    OBJECTS {
        dot11RSNAActivated,
        dot11RSNAPreauthenticationActivated }
    STATUS current
    DESCRIPTION
        "This object class provides the objects from the IEEE 802.11 MIB required
        to manage RSNA functionality. Note that additional objects for managing
        this functionality are located in the IEEE 802.11 RSN MIB."
    ::= { dot11Groups 25 }

dot11SMTbase4 OBJECT-GROUP
    OBJECTS {
        dot11MediumOccupancyLimit,
        dot11CFPollable,
        dot11CFPPeriod,
        dot11CFPMaxDuration,
        dot11AuthenticationResponseTimeOut,
        dot11PrivacyOptionImplemented,
        dot11PowerManagementMode,
        dot11DesiredSSID, dot11DesiredBSSType,
        dot11OperationalRateSet,
        dot11BeaconPeriod, dot11DTIMPeriod,
        dot11AssociationResponseTimeOut,
        dot11DisassociateReason,
        dot11DisassociateStation,
        dot11DeauthenticateReason,
        dot11DeauthenticateStation,
        dot11AuthenticateFailStatus,
        dot11AuthenticateFailStation,
        dot11MultiDomainCapabilityImplemented,
        dot11MultiDomainCapabilityActivated,
        dot11CountryString,
        dot11RSNAOptionImplemented }
    STATUS deprecated
    DESCRIPTION
        "Superseded by dot11SMTbase5.
        The SMTbase4 object class provides the necessary support at the IEEE STA
        to manage the processes in the STA so that the STA may work cooperatively
        as a part of an IEEE 802.11 network."
    ::= { dot11Groups 26 }

-- ********************************************************************
-- * Groups - units of conformance - RSN
-- ********************************************************************

dot11RSNBase OBJECT-GROUP
    OBJECTS {
        dot11RSNAConfigVersion,
        dot11RSNAConfigPairwiseKeysImplemented,
        dot11RSNAConfigGroupCipher,
        dot11RSNAConfigGroupRekeyMethod,
        dot11RSNAConfigGroupRekeyTime,
        dot11RSNAConfigGroupRekeyPackets,
        dot11RSNAConfigGroupRekeyStrict,
        dot11RSNAConfigPSKValue,
        dot11RSNAConfigPSKPassPhrase,
        dot11RSNAConfigGroupUpdateCount,
        dot11RSNAConfigPairwiseUpdateCount,
        dot11RSNAConfigGroupCipherSize,
        dot11RSNAConfigPairwiseCipherImplemented,
        dot11RSNAConfigPairwiseCipherActivated,
        dot11RSNAConfigPairwiseCipherSizeImplemented,
        dot11RSNAConfigAuthenticationSuiteImplemented,
        dot11RSNAConfigAuthenticationSuiteActivated,
        dot11RSNAConfigNumberOfPTKSAReplayCountersImplemented,
        dot11RSNAConfigSATimeout,
        dot11RSNAConfigNumberOfGTKSAReplayCountersImplemented,
        dot11RSNAAuthenticationSuiteSelected,
        dot11RSNAPairwiseCipherSelected,
        dot11RSNAGroupCipherSelected,
        dot11RSNAPMKIDUsed,
        dot11RSNAAuthenticationSuiteRequested,
        dot11RSNAPairwiseCipherRequested,
        dot11RSNAGroupCipherRequested,
        dot11RSNATKIPCounterMeasuresInvoked,
        dot11RSNA4WayHandshakeFailures,
        dot11RSNAStatsSTAAddress,
        dot11RSNAStatsVersion,
        dot11RSNAStatsSelectedPairwiseCipher,
        dot11RSNAStatsTKIPICVErrors,
        dot11RSNAStatsTKIPLocalMICFailures,
        dot11RSNAStatsTKIPRemoteMICFailures,
        dot11RSNAStatsCCMPReplays,
        dot11RSNAStatsCCMPDecryptErrors,
        dot11RSNAStatsTKIPReplays,
        dot11RSNAConfigSTKKeysImplemented,
        dot11RSNAConfigSTKCipher,
        dot11RSNAConfigSTKRekeyTime,
        dot11RSNAConfigSMKUpdateCount,
        dot11RSNAConfigSTKCipherSize,
        dot11RSNAConfigNumberOfSTKSAReplayCountersImplemented,
        dot11RSNAPairwiseSTKSelected,
        dot11RSNAPreauthenticationImplemented,
        dot11RSNASMKHandshakeFailures }
    STATUS current
    DESCRIPTION
        "The dot11RSNBase object class provides the necessary support for managing
        RSNA functionality in the STA."
    ::= { dot11Groups 27 }

dot11RSNPMKcachingGroup OBJECT-GROUP
    OBJECTS {
        dot11RSNAConfigPMKLifetime,
        dot11RSNAConfigPMKReauthThreshold }
    STATUS current
    DESCRIPTION
        "The dot11RSNPMKcachingGroup object class provides the necessary support
        for managing PMK caching functionality in the STA"
    ::= { dot11Groups 28 }

dot11RSNSMKcachingGroup OBJECT-GROUP
    OBJECTS {
        dot11RSNAConfigSMKLifetime,
        dot11RSNAConfigSMKReauthThreshold }
    STATUS current
    DESCRIPTION
        "The dot11RSNSMKcachingGroup object class provides the necessary support
        for managing SMK caching functionality in the STA."
    ::= { dot11Groups 29 }

dot11SMTbase5 OBJECT-GROUP
    OBJECTS {
        dot11MediumOccupancyLimit,
        dot11CFPollable,
        dot11CFPPeriod,
        dot11CFPMaxDuration,
        dot11AuthenticationResponseTimeOut,
        dot11PrivacyOptionImplemented,
        dot11PowerManagementMode,
        dot11DesiredSSID, dot11DesiredBSSType,
        dot11OperationalRateSet,
        dot11BeaconPeriod, dot11DTIMPeriod,
        dot11AssociationResponseTimeOut,
        dot11DisassociateReason,
        dot11DisassociateStation,
        dot11DeauthenticateReason,
        dot11DeauthenticateStation,
        dot11AuthenticateFailStatus,
        dot11AuthenticateFailStation,
        dot11MultiDomainCapabilityImplemented,
        dot11MultiDomainCapabilityActivated,
        dot11CountryString,
        dot11SpectrumManagementImplemented,
        dot11SpectrumManagementRequired,
        dot11RSNAOptionImplemented,
        dot11OperatingClassesImplemented,
        dot11OperatingClassesRequired }
    STATUS current
    DESCRIPTION
        "The SMTbase5 object class provides the necessary support at the STA to
        manage the processes in the STA so that the STA may work cooperatively as
        a part of an IEEE 802.11 network, when the STA is capable of multidomain
        operation. This object group should be implemented when the multidomain
        capability option is implemented."
    ::= { dot11Groups 30 }

dot11MACbase2 OBJECT-GROUP
    OBJECTS {
        dot11MACAddress,
        dot11Address,
        dot11GroupAddressesStatus,
        dot11RTSThreshold,
        dot11ShortRetryLimit,
        dot11LongRetryLimit,
        dot11FragmentationThreshold,
        dot11MaxTransmitMSDULifetime,
        dot11MaxReceiveLifetime,
        dot11ManufacturerID,
        dot11ProductID,
        dot11CAPLimit,
        dot11HCCWmin,
        dot11HCCWmax,
        dot11HCCAIFSN,
        dot11ADDBAResponseTimeout,
        dot11ADDTSResponseTimeout,
        dot11ChannelUtilizationBeaconInterval,
        dot11ScheduleTimeout,
        dot11DLSResponseTimeout,
        dot11QAPMissingAckRetryLimit,
        dot11EDCAAveragingPeriod }
    STATUS deprecated
        DESCRIPTION
        "Superseded by dot11MACbase3.
        The MAC object class provides the necessary support for the access con-
        trol, generation, and verification of frame check sequences (FCSs), and
        proper delivery of valid data to upper layers."
    ::= { dot11Groups 31 }

dot11CountersGroup2 OBJECT-GROUP
    OBJECTS {
        dot11TransmittedFragmentCount,
        dot11GroupTransmittedFrameCount,
        dot11FailedCount,
        dot11ReceivedFragmentCount,
        dot11GroupReceivedFrameCount,
        dot11FCSErrorCount,
        dot11WEPUndecryptableCount,
        dot11TransmittedFrameCount,
        dot11QosDiscardedFragmentCount,
        dot11AssociatedStationCount,
        dot11QosCFPollsReceivedCount,
        dot11QosCFPollsUnusedCount,
        dot11QosCFPollsUnusableCount }
    STATUS deprecated
        DESCRIPTION
        "Superseded by dot11CountersGroup3.
        Attributes from the dot11CountersGroup that are not described in the
        dot11MACStatistics group. These objects are mandatory."
    ::= { dot11Groups 32 }

dot11Qosadditions OBJECT-GROUP
    OBJECTS {
--      dot11EDCATable,
        dot11EDCATableCWmin,
        dot11EDCATableCWmax,
        dot11EDCATableAIFSN,
        dot11EDCATableTXOPLimit,
        dot11EDCATableMSDULifetime,
        dot11EDCATableMandatory,

--      dot11QAPEDCATable,
        dot11QAPEDCATableCWmin,
        dot11QAPEDCATableCWmax,
        dot11QAPEDCATableAIFSN,
        dot11QAPEDCATableTXOPLimit,
        dot11QAPEDCATableMSDULifetime,
        dot11QAPEDCATableMandatory,

--      dot11QosCountersTable
        dot11QosTransmittedFragmentCount,
        dot11QosFailedCount,
        dot11QosRetryCount,
        dot11QosMultipleRetryCount,
        dot11QosFrameDuplicateCount,
        dot11QosRTSSuccessCount,
        dot11QosRTSFailureCount,
        dot11QosACKFailureCount,
        dot11QosReceivedFragmentCount,
        dot11QosTransmittedFrameCount,
        dot11QosDiscardedFrameCount,
        dot11QosMPDUsReceivedCount,
        dot11QosRetriesReceivedCount }
    STATUS current
    DESCRIPTION
        "This object class provides the objects from the IEEE 802.11 MIB required
        to manage QoS functionality."
    ::= { dot11Groups 33 }

dot11SMTbase6 OBJECT-GROUP
    OBJECTS {
        dot11MediumOccupancyLimit,
        dot11CFPollable,
        dot11CFPPeriod,
        dot11CFPMaxDuration,
        dot11AuthenticationResponseTimeOut,
        dot11PrivacyOptionImplemented,
        dot11PowerManagementMode,
        dot11DesiredSSID,
        dot11DesiredBSSType,
        dot11OperationalRateSet,
        dot11BeaconPeriod,
        dot11DTIMPeriod,
        dot11AssociationResponseTimeOut,
        dot11DisassociateReason,
        dot11DisassociateStation,
        dot11DeauthenticateReason,
        dot11DeauthenticateStation,
        dot11AuthenticateFailStatus,
        dot11AuthenticateFailStation,
        dot11MultiDomainCapabilityImplemented,
        dot11MultiDomainCapabilityActivated,
        dot11CountryString,
        dot11RSNAOptionImplemented,
        dot11OperatingClassesImplemented,
        dot11OperatingClassesRequired,
        dot11QosOptionImplemented,
        dot11ImmediateBlockAckOptionImplemented,
        dot11DelayedBlockAckOptionImplemented,
        dot11DirectOptionImplemented,
        dot11APSDOptionImplemented,
        dot11QAckOptionImplemented,
        dot11QBSSLoadImplemented,
        dot11QueueRequestOptionImplemented,
        dot11TXOPRequestOptionImplemented,
        dot11MoreDataAckOptionImplemented,
        dot11AssociateInNQBSS,
        dot11DLSAllowedInQBSS,
        dot11DLSAllowed }
    STATUS deprecated
    DESCRIPTION
        "Superseded by dot11SMTbase7.
        The SMTbase6 object class provides the necessary support at the STA to
        manage the processes in the STA such that the STA may work cooperatively
        as a part of an IEEE 802.11 network."
    ::= { dot11Groups 34 }

dot11PhyOFDMComplianceGroup2 OBJECT-GROUP
    OBJECTS {
        dot11CurrentFrequency,
        dot11TIThreshold,
        dot11FrequencyBandsImplemented,
        dot11ChannelStartingFactor,
        dot11FiveMHzOperationImplemented,
        dot11TenMHzOperationImplemented,
        dot11TwentyMHzOperationImplemented,
        dot11PhyOFDMChannelWidth }
    STATUS deprecated
    DESCRIPTION
        "Superseded by dot11PhyOFDMComplianceGroup3.
        Attributes that configure the OFDM for IEEE 802.11."
    ::= { dot11Groups 35}

dot11SMTbase7 OBJECT-GROUP
    OBJECTS{
        dot11MediumOccupancyLimit,
        dot11CFPollable,
        dot11CFPPeriod,
        dot11CFPMaxDuration,
        dot11AuthenticationResponseTimeOut,
        dot11PrivacyOptionImplemented,
        dot11PowerManagementMode,
        dot11DesiredSSID,
        dot11DesiredBSSType,
        dot11OperationalRateSet,
        dot11BeaconPeriod,
        dot11DTIMPeriod,
        dot11AssociationResponseTimeOut,
        dot11DisassociateReason,
        dot11DisassociateStation,
        dot11DeauthenticateReason,
        dot11DeauthenticateStation,
        dot11AuthenticateFailStatus,
        dot11AuthenticateFailStation,
        dot11MultiDomainCapabilityImplemented,
        dot11MultiDomainCapabilityActivated,
        dot11CountryString,
        dot11SpectrumManagementImplemented,
        dot11SpectrumManagementRequired,
        dot11RSNAOptionImplemented,
        dot11OperatingClassesImplemented,
        dot11OperatingClassesRequired,
        dot11QosOptionImplemented,
        dot11ImmediateBlockAckOptionImplemented,
        dot11DelayedBlockAckOptionImplemented,
        dot11DirectOptionImplemented,
        dot11APSDOptionImplemented,
        dot11QAckOptionImplemented,
        dot11QBSSLoadImplemented,
        dot11QueueRequestOptionImplemented,
        dot11TXOPRequestOptionImplemented,
        dot11MoreDataAckOptionImplemented,
        dot11AssociateInNQBSS,
        dot11DLSAllowedInQBSS,
        dot11DLSAllowed,
        dot11AssociateStation,
        dot11AssociateID,
        dot11AssociateFailStation,
        dot11AssociateFailStatus,
        dot11ReassociateStation,
        dot11ReassociateID,
        dot11ReassociateFailStation,
        dot11ReassociateFailStatus,
        dot11RadioMeasurementImplemented,
        dot11RadioMeasurementActivated,
        dot11RMMeasurementProbeDelay,
        dot11RMMeasurementPilotPeriod,
        dot11RMLinkMeasurementActivated,
        dot11RMNeighborReportActivated,
        dot11RMParallelMeasurementsActivated,
        dot11RMRepeatedMeasurementsActivated,
        dot11RMBeaconPassiveMeasurementActivated,
        dot11RMBeaconActiveMeasurementActivated,
        dot11RMBeaconTableMeasurementActivated,
        dot11RMBeaconMeasurementReportingConditionsActivated,
        dot11RMFrameMeasurementActivated,
        dot11RMChannelLoadMeasurementActivated,
        dot11RMNoiseHistogramMeasurementActivated,
        dot11RMStatisticsMeasurementActivated,
        dot11RMLCIMeasurementActivated,
        dot11RMLCIAzimuthActivated,
        dot11RMTransmitStreamCategoryMeasurementActivated,
        dot11RMTriggeredTransmitStreamCategoryMeasurementActivated,
        dot11RMAPChannelReportActivated,
        dot11RMMIBActivated,
        dot11RMMaxMeasurementDuration,
        dot11RMNonOperatingChannelMaxMeasurementDuration,
        dot11RMMeasurementPilotTransmissionInformationActivated,
        dot11RMMeasurementPilotActivated,
        dot11RMNeighborReportTSFOffsetActivated,
        dot11RMRCPIMeasurementActivated,
        dot11RMRSNIMeasurementActivated,
        dot11RMBSSAverageAccessDelayActivated,
        dot11RMBSSAvailableAdmissionCapacityActivated,
        dot11RMAntennaInformationActivated}
    STATUS deprecated
    DESCRIPTION
        "Superseded by dot11SMTbase8.
        The SMTbase7 object class provides the necessary support at the STA to
        manage the processes in the STA such that the STA may work cooperatively
        as a part of an IEEE 802.11 net-work, when the STA is capable of multi-
        domain operation. This object group should be implemented when the multi-
        domain capability option is implemented."
    ::= { dot11Groups 36 }

dot11SMTRMRequest OBJECT-GROUP
    OBJECTS {
        dot11RMRqstRowStatus,
        dot11RMRqstToken,
        dot11RMRqstRepetitions,
        dot11RMRqstIfIndex,
        dot11RMRqstType,
        dot11RMRqstTargetAdd,
        dot11RMRqstTimeStamp,
        dot11RMRqstChanNumber,
        dot11RMRqstOperatingClass,
        dot11RMRqstRndInterval,
        dot11RMRqstDuration,
        dot11RMRqstParallel,
        dot11RMRqstEnable,
        dot11RMRqstRequest,
        dot11RMRqstReport,
        dot11RMRqstDurationMandatory,
        dot11RMRqstBeaconRqstMode,
        dot11RMRqstBeaconRqstDetail,
        dot11RMRqstFrameRqstType,
        dot11RMRqstBssid,
        dot11RMRqstSSID,
        dot11RMRqstBeaconReportingCondition,
        dot11RMRqstBeaconThresholdOffset,
        dot11RMRqstSTAStatRqstGroupID,
        dot11RMRqstLCIRqstSubject,
        dot11RMRqstLCILatitudeResolution,
        dot11RMRqstLCILongitudeResolution,
        dot11RMRqstLCIAltitudeResolution,
        dot11RMRqstLCIAzimuthType,
        dot11RMRqstLCIAzimuthResolution,
        dot11RMRqstPauseTime,
        dot11RMRqstTransmitStreamPeerQSTAAddress,
        dot11RMRqstTransmitStreamTrafficIdentifier,
        dot11RMRqstTransmitStreamBin0Range,
        dot11RMRqstTrigdQoSAverageCondition,
        dot11RMRqstTrigdQoSConsecutiveCondition,
        dot11RMRqstTrigdQoSDelayCondition,
        dot11RMRqstTrigdQoSAverageThreshold,
        dot11RMRqstTrigdQoSConsecutiveThreshold,
        dot11RMRqstTrigdQoSDelayThresholdRange,
        dot11RMRqstTrigdQoSDelayThreshold,
        dot11RMRqstTrigdQoSMeasurementCount,
        dot11RMRqstTrigdQoSTimeout,
        dot11RMRqstChannelLoadReportingCondition,
        dot11RMRqstChannelLoadReference,
        dot11RMRqstNoiseHistogramReportingCondition,
        dot11RMRqstAnpiReference,
        dot11RMRqstAPChannelReport,
        dot11RMRqstSTAStatPeerSTAAddress,
        dot11RMRqstFrameTransmitterAddress,
        dot11RMRqstVendorSpecific }
    STATUS current
    DESCRIPTION
        "The SMTRMRequest package is a set of attributes that are present if the
        STA supports the Radio Measurement service."
    ::= { dot11Groups 37 }

dot11SMTRMReport OBJECT-GROUP
    OBJECTS {
        dot11ChannelLoadRprtRqstToken,
        dot11ChannelLoadRprtIfIndex,
        dot11ChannelLoadMeasuringSTAAddr,
        dot11ChannelLoadRprtChanNumber,
        dot11ChannelLoadRprtOperatingClass,
        dot11ChannelLoadRprtActualStartTime,
        dot11ChannelLoadRprtMeasurementDuration,
        dot11ChannelLoadRprtChannelLoad,
        dot11ChannelLoadRprtVendorSpecific,
        dot11ChannelLoadRprtMeasurementMode,
        dot11NoiseHistogramRprtRqstToken,
        dot11NoiseHistogramRprtIfIndex,
        dot11NoiseHistogramMeasuringSTAAddr,
        dot11NoiseHistogramRprtChanNumber,
        dot11NoiseHistogramRprtOperatingClass,
        dot11NoiseHistogramRprtActualStartTime,
        dot11NoiseHistogramRprtMeasurementDuration,
        dot11NoiseHistogramRprtAntennaID,
        dot11NoiseHistogramRprtANPI,
        dot11NoiseHistogramRprtIPIDensity0,
        dot11NoiseHistogramRprtIPIDensity1,
        dot11NoiseHistogramRprtIPIDensity2,
        dot11NoiseHistogramRprtIPIDensity3,
        dot11NoiseHistogramRprtIPIDensity4,
        dot11NoiseHistogramRprtIPIDensity5,
        dot11NoiseHistogramRprtIPIDensity6,
        dot11NoiseHistogramRprtIPIDensity7,
        dot11NoiseHistogramRprtIPIDensity8,
        dot11NoiseHistogramRprtIPIDensity9,
        dot11NoiseHistogramRprtIPIDensity10,
        dot11NoiseHistogramRprtVendorSpecific,
        dot11NoiseHistogramRprtMeasurementMode,
        dot11BeaconRprtRqstToken,
        dot11BeaconRprtIfIndex,
        dot11BeaconMeasuringSTAAddr,
        dot11BeaconRprtChanNumber,
        dot11BeaconRprtOperatingClass,
        dot11BeaconRprtActualStartTime,
        dot11BeaconRprtMeasurementDuration,
        dot11BeaconRprtPhyType,
        dot11BeaconRprtReportedFrameType,
        dot11BeaconRprtRCPI,
        dot11BeaconRprtRSNI,
        dot11BeaconRprtBSSID,
        dot11BeaconRprtAntennaID,
        dot11BeaconRprtParentTSF,
        dot11BeaconRprtReportedFrameBody,
        dot11BeaconRprtVendorSpecific,
        dot11BeaconRprtMeasurementMode,
        dot11FrameRprtIfIndex,
        dot11FrameRprtRqstToken,
        dot11FrameRprtChanNumber,
        dot11FrameRprtOperatingClass,
        dot11FrameRprtActualStartTime,
        dot11FrameRprtMeasurementDuration,
        dot11FrameRprtTransmitSTAAddress,
        dot11FrameRprtBSSID,
        dot11FrameRprtPhyType,
        dot11FrameRprtAvgRCPI,
        dot11FrameRprtLastRSNI,
        dot11FrameRprtLastRCPI,
        dot11FrameRprtAntennaID,
        dot11FrameRprtNumberFrames,
        dot11FrameRprtVendorSpecific,
        dot11FrameRptMeasurementMode,
        dot11STAStatisticsReportToken,
        dot11STAStatisticsIfIndex,
        dot11STAStatisticsSTAAddress,
        dot11STAStatisticsMeasurementDuration,
        dot11STAStatisticsGroupID,
        dot11STAStatisticsTransmittedFragmentCount,
        dot11STAStatisticsGroupTransmittedFrameCount,
        dot11STAStatisticsFailedCount,
        dot11STAStatisticsRetryCount,
        dot11STAStatisticsMultipleRetryCount,
        dot11STAStatisticsFrameDuplicateCount,
        dot11STAStatisticsRTSSuccessCount,
        dot11STAStatisticsRTSFailureCount,
        dot11STAStatisticsACKFailureCount,
        dot11STAStatisticsQosTransmittedFragmentCount,
        dot11STAStatisticsQosFailedCount,
        dot11STAStatisticsQosRetryCount,
        dot11STAStatisticsQosMultipleRetryCount,
        dot11STAStatisticsQosFrameDuplicateCount,
        dot11STAStatisticsQosRTSSuccessCount,
        dot11STAStatisticsQosRTSFailureCount,
        dot11STAStatisticsQosACKFailureCount,
        dot11STAStatisticsQosReceivedFragmentCount,
        dot11STAStatisticsQosTransmittedFrameCount,
        dot11STAStatisticsQosDiscardedFrameCount,
        dot11STAStatisticsQosMPDUsReceivedCount,
        dot11STAStatisticsQosRetriesReceivedCount,
        dot11STAStatisticsReceivedFragmentCount,
        dot11STAStatisticsGroupReceivedFrameCount,
        dot11STAStatisticsFCSErrorCount,
        dot11STAStatisticsTransmittedFrameCount,
        dot11STAStatisticsAPAverageAccessDelay,
        dot11STAStatisticsAverageAccessDelayBestEffort,
        dot11STAStatisticsAverageAccessDelayBackground,
        dot11STAStatisticsAverageAccessDelayVideo,
        dot11STAStatisticsAverageAccessDelayVoice,
        dot11STAStatisticsStationCount,
        dot11STAStatisticsChannelUtilization,
        dot11STAStatisticsVendorSpecific,
        dot11STAStatisticsRprtMeasurementMode,
        dot11LCIReportToken,
        dot11LCIIfIndex,
        dot11LCISTAAddress,
        dot11LCILatitudeResolution,
        dot11LCILatitudeInteger,
        dot11LCILatitudeFraction,
        dot11LCILongitudeResolution,
        dot11LCILongitudeInteger,
        dot11LCILongitudeFraction,
        dot11LCIAltitudeType,
        dot11LCIAltitudeResolution,
        dot11LCIAltitudeInteger,
        dot11LCIAltitudeFraction,
        dot11LCIDatum,
        dot11LCIAzimuthType,
        dot11LCIAzimuthResolution,
        dot11LCIAzimuth,
        dot11LCIVendorSpecific,
        dot11LCIRprtMeasurementMode,
        dot11TransmitStreamRprtRqstToken,
        dot11TransmitStreamRprtIfIndex,
        dot11TransmitStreamMeasuringSTAAddr,
        dot11TransmitStreamRprtActualStartTime,
        dot11TransmitStreamRprtMeasurementDuration,
        dot11TransmitStreamRprtPeerSTAAddress,
        dot11TransmitStreamRprtTID,
        dot11TransmitStreamRprtAverageQueueDelay,
        dot11TransmitStreamRprtAverageTransmitDelay,
        dot11TransmitStreamRprtTransmittedMSDUCount,
        dot11TransmitStreamRprtMSDUDiscardedCount,
        dot11TransmitStreamRprtMSDUFailedCount,
        dot11TransmitStreamRprtMultipleRetryCount,
        dot11TransmitStreamRprtCFPollsLostCount,
        dot11TransmitStreamRprtBin0Range,
        dot11TransmitStreamRprtDelayHistogram,
        dot11TransmitStreamRprtReason,
        dot11TransmitStreamRprtVendorSpecific,
        dot11TransmitStreamRprtMeasurementMode }
    STATUS current
    DESCRIPTION
        "The SMTRMReport package is a set of attributes that are present if the
        STA supports the Radio Measurement service."
    ::= { dot11Groups 38 }

dot11SMTRMConfig OBJECT-GROUP
    OBJECTS {
        dot11APChannelReportIfIndex,
        dot11APChannelReportOperatingClass,
        dot11APChannelReportChannelList,
        dot11RMNeighborReportIfIndex,
        dot11RMNeighborReportBSSID,
        dot11RMNeighborReportAPReachability,
        dot11RMNeighborReportSecurity,
        dot11RMNeighborReportCapSpectrumMgmt,
        dot11RMNeighborReportCapQoS,
        dot11RMNeighborReportCapAPSD,
        dot11RMNeighborReportCapRM,
        dot11RMNeighborReportCapDelayBlockAck,
        dot11RMNeighborReportCapImmediateBlockAck,
        dot11RMNeighborReportKeyScope,
        dot11RMNeighborReportChannelNumber,
        dot11RMNeighborReportOperatingClass,
        dot11RMNeighborReportPhyType,
        dot11RMNeighborReportNeighborTSFInfo,
        dot11RMNeighborReportPilotInterval,
        dot11RMNeighborReportPilotMultipleBSSID,
        dot11RMNeighborReportRMEnabledCapabilities,
        dot11RMNeighborReportVendorSpecific,
        dot11RMNeighborReportRowStatus,
        dot11RMNeighborReportCapHT,
        dot11RMNeighborReportHTLDPCCodingCap,
        dot11RMNeighborReportHTSupportedChannelWidthSet,
        dot11RMNeighborReportHTSMPowerSave,
        dot11RMNeighborReportHTGreenfield,
        dot11RMNeighborReportHTShortGIfor20MHz,
        dot11RMNeighborReportHTShortGIfor40MHz,
        dot11RMNeighborReportHTTxSTBC,
        dot11RMNeighborReportHTRxSTBC,
        dot11RMNeighborReportHTDelayedBlockAck,
        dot11RMNeighborReportHTMaxAMSDULength,
        dot11RMNeighborReportHTDSSCCKModein40MHz,
        dot11RMNeighborReportHTFortyMHzIntolerant,
        dot11RMNeighborReportHTLSIGTXOPProtectionSupport,
        dot11RMNeighborReportHTMaxAMPDULengthExponent,
        dot11RMNeighborReportHTMinMPDUStartSpacing,
        dot11RMNeighborReportHTRxMCSBitMask,
        dot11RMNeighborReportHTRxHighestSupportedDataRate,
        dot11RMNeighborReportHTTxMCSSetDefined,
        dot11RMNeighborReportHTTxRxMCSSetNotEqual,
        dot11RMNeighborReportHTTxMaxNumberSpatialStreamsSupported,
        dot11RMNeighborReportHTTxUnequalModulationSupported,
        dot11RMNeighborReportHTPCO,
        dot11RMNeighborReportHTPCOTransitionTime,
        dot11RMNeighborReportHTMCSFeedback,
        dot11RMNeighborReportHTCSupport,
        dot11RMNeighborReportHTRDResponder,
        dot11RMNeighborReportHTImplictTransmitBeamformingReceivingCap,
        dot11RMNeighborReportHTReceiveStaggeredSoundingCap,
        dot11RMNeighborReportHTTransmitStaggeredSoundingCap,
        dot11RMNeighborReportHTReceiveNDPCap,
        dot11RMNeighborReportHTTransmitNDPCap,
        dot11RMNeighborReportHTImplicitTransmitBeamformingCap,
        dot11RMNeighborReportHTTransmitBeamformingCalibration,
        dot11RMNeighborReportHTExplicitCSITransmitBeamformingCap,
        dot11RMNeighborReportHTExplicitNonCompressedSteeringCap,
        dot11RMNeighborReportHTExplicitCompressedSteeringCap,
        dot11RMNeighborReportHTExplicitTransmitBeamformingFeedback,
        dot11RMNbRprtHTExplicitNonCompressedBeamformingFeedbackCap,
        dot11RMNeighborReportHTExplicitCompressedBeamformingFeedbackCap,
        dot11RMNeighborReportHTTransmitBeamformingMinimalGrouping,
        dot11RMNbRprtHTCSINumberofTxBeamformingAntennasSuppt,
        dot11RMNbRprtHTNonCompressedSteeringNumofTxBmfmingAntennasSuppt,
        dot11RMNbRprtHTCompressedSteeringNumberofTxBmfmingAntennasSuppt,
        dot11RMNbRprtHTCSIMaxNumberofRowsTxBeamformingSuppt,
        dot11RMNeighborReportHTTransmitBeamformingChannelEstimationCap,
        dot11RMNeighborReportHTAntSelectionCap,
        dot11RMNeighborReportHTExplicitCSIFeedbackBasedTxASELCap,
        dot11RMNeighborReportHTAntIndicesFeedbackBasedTxASELCap,
        dot11RMNeighborReportHTExplicitCSIFeedbackBasedCap,
        dot11RMNeighborReportHTAntIndicesFeedbackCap,
        dot11RMNeighborReportHTRxASELCap,
        dot11RMNeighborReportHTTxSoundingPPDUsCap,
        dot11RMNeighborReportHTInfoPrimaryChannel,
        dot11RMNeighborReportHTInfoSecChannelOffset,
        dot11RMNeighborReportHTInfoSTAChannelWidth,
        dot11RMNeighborReportHTInfoRIFSMode,
        dot11RMNeighborReportHTInfoProtection,
        dot11RMNeighborReportHTInfoNonGreenfieldHTSTAsPresent,
        dot11RMNeighborReportHTInfoOBSSNonHTSTAsPresent,
        dot11RMNeighborReportHTInfoDualBeacon,
        dot11RMNeighborReportHTInfoDualCTSProtection,
        dot11RMNeighborReportHTInfoSTBCBeacon,
        dot11RMNeighborReportHTInfoLSIGTXOPProtectionSup,
        dot11RMNeighborReportHTInfoPCOActive,
        dot11RMNeighborReportHTInfoPCOPhase,
        dot11RMNeighborReportHTInfoBasicMCSSet,
        dot11RMNeighborReportHTSecChannelOffset,
        dot11RMNeighborReportExtCapPSMPSupport,
        dot11RMNeighborReportExtCapSPSMPSup,
        dot11RMNeighborReportExtCapServiceIntervalGranularity }
    STATUS current
    DESCRIPTION
        "The SMTRMConfig package is a set of attributes that are present if the
        STA supports the Radio Measurement service."
    ::= { dot11Groups 39 }

dot11FTComplianceGroup OBJECT-GROUP
    OBJECTS {
        -- Dot11FastBSSTransitionConfigEntry
        dot11FastBSSTransitionActivated,
        dot11FTMobilityDomainID,
        dot11FTOverDSActivated,
        dot11FTResourceRequestSupported,
        dot11FTR0KeyHolderID,
        dot11FTR0KeyLifetime,
        dot11FTR1KeyHolderID,
        dot11FTReassociationDeadline,
        -- Dot11RMNeighborReportEntry
        dot11RMNeighborReportMobilityDomain }
    STATUS current
    DESCRIPTION
        "This object class provides the objects from the IEEE 802.11 MIB required
        to manage fast BSS transition functionality. Note that additional objects
        for managing this functionality are located in the dot11FastBSS Transi-
        tionConfigTable."
    ::= { dot11Groups 40}

dot11SMTbase8 OBJECT-GROUP
    OBJECTS {
        dot11MediumOccupancyLimit,
        dot11CFPollable,
        dot11CFPPeriod,
        dot11CFPMaxDuration,
        dot11AuthenticationResponseTimeOut,
        dot11PrivacyOptionImplemented,
        dot11PowerManagementMode,
        dot11DesiredSSID, dot11DesiredBSSType,
        dot11OperationalRateSet,
        dot11BeaconPeriod, dot11DTIMPeriod,
        dot11AssociationResponseTimeOut,
        dot11DisassociateReason,
        dot11DisassociateStation,
        dot11DeauthenticateReason,
        dot11DeauthenticateStation,
        dot11AuthenticateFailStatus,
        dot11AuthenticateFailStation,
        dot11MultiDomainCapabilityImplemented,
        dot11MultiDomainCapabilityActivated,
        dot11CountryString,
        dot11SpectrumManagementImplemented,
        dot11SpectrumManagementRequired,
        dot11RSNAOptionImplemented,
        dot11OperatingClassesImplemented,
        dot11OperatingClassesRequired,
        dot11QosOptionImplemented,
        dot11ImmediateBlockAckOptionImplemented,
        dot11DelayedBlockAckOptionImplemented,
        dot11DirectOptionImplemented,
        dot11APSDOptionImplemented,
        dot11QAckOptionImplemented,
        dot11QBSSLoadImplemented,
        dot11QueueRequestOptionImplemented,
        dot11TXOPRequestOptionImplemented,
        dot11MoreDataAckOptionImplemented,
        dot11AssociateInNQBSS,
        dot11DLSAllowedInQBSS,
        dot11DLSAllowed,
        dot11AssociateStation,
        dot11AssociateID,
        dot11AssociateFailStation,
        dot11AssociateFailStatus,
        dot11ReassociateStation,
        dot11ReassociateID,
        dot11ReassociateFailStation,
        dot11ReassociateFailStatus,
        dot11RadioMeasurementImplemented,
        dot11RadioMeasurementActivated,
        dot11RMMeasurementProbeDelay,
        dot11RMMeasurementPilotPeriod,
        dot11RMLinkMeasurementActivated,
        dot11RMNeighborReportActivated,
        dot11RMParallelMeasurementsActivated,
        dot11RMRepeatedMeasurementsActivated,
        dot11RMBeaconPassiveMeasurementActivated,
        dot11RMBeaconActiveMeasurementActivated,
        dot11RMBeaconTableMeasurementActivated,
        dot11RMBeaconMeasurementReportingConditionsActivated,
        dot11RMFrameMeasurementActivated,
        dot11RMChannelLoadMeasurementActivated,
        dot11RMNoiseHistogramMeasurementActivated,
        dot11RMStatisticsMeasurementActivated,
        dot11RMLCIMeasurementActivated,
        dot11RMLCIAzimuthActivated,
        dot11RMTransmitStreamCategoryMeasurementActivated,
        dot11RMTriggeredTransmitStreamCategoryMeasurementActivated,
        dot11RMAPChannelReportActivated,
        dot11RMMIBActivated,
        dot11RMMaxMeasurementDuration,
        dot11RMNonOperatingChannelMaxMeasurementDuration,
        dot11RMMeasurementPilotTransmissionInformationActivated,
        dot11RMMeasurementPilotActivated,
        dot11RMNeighborReportTSFOffsetActivated,
        dot11RMRCPIMeasurementActivated,
        dot11RMRSNIMeasurementActivated,
        dot11RMBSSAverageAccessDelayActivated,
        dot11RMBSSAvailableAdmissionCapacityActivated,
        dot11RMAntennaInformationActivated,
        dot11FastBSSTransitionImplemented }
    STATUS deprecated
    DESCRIPTION
        "Superseded by dot11SMTbase9.
        The SMTbase8 object class provides the necessary support at the STA to
        manage the processes in the STA so that the STA may work cooperatively as
        a part of an IEEE 802.11 network, when the STA is capable of multidomain
        operation. This object group should be implemented when the multidomain
        capability option is implemented."
    ::= { dot11Groups 41 }

 dot11PhyOFDMComplianceGroup3 OBJECT-GROUP
    OBJECTS {
        dot11CurrentFrequency,
        dot11FrequencyBandsImplemented,
        dot11ChannelStartingFactor,
        dot11FiveMHzOperationImplemented,
        dot11TenMHzOperationImplemented,
        dot11TwentyMHzOperationImplemented,
        dot11PhyOFDMChannelWidth,
        dot11OFDMCCAEDImplemented,
        dot11OFDMCCAEDRequired,
        dot11OFDMEDThreshold,
        dot11STATransmitPowerClass,
        dot11ACRType }
    STATUS current
    DESCRIPTION
        "Attributes that configure the OFDM for IEEE 802.11."
    ::= { dot11Groups 42}

dot11SMTbase9 OBJECT-GROUP
    OBJECTS {
        dot11MediumOccupancyLimit,
        dot11CFPollable,
        dot11CFPPeriod,
        dot11CFPMaxDuration,
        dot11AuthenticationResponseTimeOut,
        dot11PrivacyOptionImplemented,
        dot11PowerManagementMode,
        dot11DesiredSSID, dot11DesiredBSSType,
        dot11OperationalRateSet,
        dot11BeaconPeriod, dot11DTIMPeriod,
        dot11AssociationResponseTimeOut,
        dot11DisassociateReason,
        dot11DisassociateStation,
        dot11DeauthenticateReason,
        dot11DeauthenticateStation,
        dot11AuthenticateFailStatus,
        dot11AuthenticateFailStation,
        dot11MultiDomainCapabilityImplemented,
        dot11MultiDomainCapabilityActivated,
        dot11CountryString,
        dot11RSNAOptionImplemented,
        dot11OperatingClassesImplemented,
        dot11OperatingClassesRequired,
        dot11QosOptionImplemented,
        dot11ImmediateBlockAckOptionImplemented,
        dot11DelayedBlockAckOptionImplemented,
        dot11DirectOptionImplemented,
        dot11APSDOptionImplemented,
        dot11QAckOptionImplemented,
        dot11QBSSLoadImplemented,
        dot11QueueRequestOptionImplemented,
        dot11TXOPRequestOptionImplemented,
        dot11MoreDataAckOptionImplemented,
        dot11AssociateInNQBSS,
        dot11DLSAllowedInQBSS,
        dot11DLSAllowed,
        dot11RadioMeasurementImplemented,
        dot11RadioMeasurementActivated,
        dot11FastBSSTransitionImplemented,
        dot11LCIDSEImplemented,
        dot11LCIDSERequired,
        dot11DSERequired,
        dot11ExtendedChannelSwitchActivated }
    STATUS deprecated
    DESCRIPTION
```

```
        "Superseded by dot11SMTbase10.
        The SMTbase9 object class provides the necessary support at the STA to
        manage the processes in the STA such that the STA may work cooperatively
        as a part of an IEEE 802.11 network, when the STA is capable of multi-
        domain operation. This object group should be implemented when the multi-
        domain capability option is implemented."
    ::= { dot11Groups 43 }

dot11PhyAntennaComplianceGroup2 OBJECT-GROUP
    OBJECTS {
        dot11CurrentTxAntenna,
        dot11DiversitySupportImplemented,
        dot11CurrentRxAntenna,
        dot11AntennaSelectionOptionImplemented,
        dot11TransmitExplicitCSIFeedbackASOptionImplemented,
        dot11TransmitIndicesFeedbackASOptionImplemented,
        dot11ExplicitCSIFeedbackASOptionImplemented,
        dot11TransmitIndicesComputationASOptionImplemented,
        dot11ReceiveAntennaSelectionOptionImplemented }
    STATUS current
    DESCRIPTION
        "Attributes for Data Rates for IEEE 802.11."
    ::= { dot11Groups 44 }

dot11MACbase3 OBJECT-GROUP
    OBJECTS {
        dot11MACAddress,
        dot11Address,
        dot11GroupAddressesStatus,
        dot11RTSThreshold,
        dot11ShortRetryLimit,
        dot11LongRetryLimit,
        dot11FragmentationThreshold,
        dot11MaxTransmitMSDULifetime,
        dot11MaxReceiveLifetime,
        dot11ManufacturerID,
        dot11ProductID,
        dot11CAPLimit,
        dot11HCCWmin,
        dot11HCCWmax,
        dot11HCCAIFSN,
        dot11ADDBAResponseTimeout,
        dot11ADDTSResponseTimeout,
        dot11ChannelUtilizationBeaconInterval,
        dot11ScheduleTimeout,
        dot11DLSResponseTimeout,
        dot11QAPMissingAckRetryLimit,
        dot11EDCAAveragingPeriod,
        dot11HTProtection,
        dot11RIFSMode,
        dot11PSMPControlledAccess,
        dot11ServiceIntervalGranularity,
        dot11DualCTSProtection,
        dot11LSIGTXOPFullProtectionActivated,
        dot11NonGFEntitiesPresent, dot11PCOActivated,
        dot11PCOFortyMaxDuration,
        dot11PCOTwentyMaxDuration,
        dot11PCOFortyMinDuration,
        dot11PCOTwentyMinDuration }
    STATUS current
    DESCRIPTION
        "The MAC object class provides the necessary support for the access con-
        trol, generation, and verification of frame check sequences (FCSs), and
        proper delivery of valid data to upper layers."
    ::= { dot11Groups 45 }

dot11CountersGroup3 OBJECT-GROUP
    OBJECTS {
        dot11TransmittedFragmentCount,
        dot11TransmittedFrameCount,
        dot11FailedCount,
        dot11ReceivedFragmentCount,
        dot11GroupReceivedFrameCount,
        dot11FCSErrorCount,
        dot11WEPUndecryptableCount,
        dot11TransmittedFrameCount,
        dot11QosDiscardedFragmentCount,
        dot11AssociatedStationCount,
        dot11QosCFPollsReceivedCount,
        dot11QosCFPollsUnusedCount,
        dot11QosCFPollsUnusableCount,
        dot11QosCFPollsLostCount,
        dot11TransmittedAMSDUCount,
        dot11FailedAMSDUCount,
        dot11RetryAMSDUCount,
        dot11MultipleRetryAMSDUCount,
        dot11TransmittedOctetsInAMSDUCount,
        dot11AMSDUAckFailureCount,
        dot11ReceivedAMSDUCount,
        dot11ReceivedOctetsInAMSDUCount,
        dot11TransmittedAMPDUCount,
        dot11TransmittedMPDUsInAMPDUCount,
        dot11TransmittedOctetsInAMPDUCount,
        dot11AMPDUReceivedCount,
        dot11MPDUInReceivedAMPDUCount,
        dot11ReceivedOctetsInAMPDUCount,
        dot11AMPDUDelimiterCRCErrorCount,
        dot11ImplicitBARFailureCount,
        dot11ExplicitBARFailureCount,
        dot11ChannelWidthSwitchCount,
        dot11TwentyMHzFrameTransmittedCount,
        dot11FortyMHzFrameTransmittedCount,
        dot11TwentyMHzFrameReceivedCount,
        dot11FortyMHzFrameReceivedCount,
        dot11PSMPUTTGrantDuration,
        dot11PSMPUTTUsedDuration,
        dot11GrantedRDGUsedCount,
        dot11GrantedRDGUnusedCount,
        dot11TransmittedFramesInGrantedRDGCount,
        dot11TransmittedOctetsInGrantedRDGCount,
        dot11BeamformingFrameCount,
        dot11DualCTSSuccessCount,
        dot11DualCTSFailureCount,
        dot11STBCCTSSuccessCount,
        dot11STBCCTSFailureCount,
        dot11nonSTBCCTSSuccessCount,
        dot11nonSTBCCTSFailureCount,
        dot11RTSLSIGSuccessCount,
        dot11RTSLSIGFailureCount }
    STATUS current
    DESCRIPTION
        "Attributes from the dot11CountersGroup that are not described in the
        dot11MACStatistics group. These objects are mandatory."
    ::= { dot11Groups 46 }

dot11SMTbase10 OBJECT-GROUP
    OBJECTS {
        dot11MediumOccupancyLimit,
        dot11CFPollable,
        dot11CFPPeriod,
        dot11CFPMaxDuration,
        dot11AuthenticationResponseTimeOut,
        dot11PrivacyOptionImplemented,
        dot11PowerManagementMode,
        dot11DesiredSSID,
        dot11DesiredBSSType,
        dot11OperationalRateSet,
        dot11BeaconPeriod,
        dot11DTIMPeriod,
        dot11AssociationResponseTimeOut,
        dot11DisassociateReason,
        dot11DisassociateStation,
        dot11DeauthenticateReason,
        dot11DeauthenticateStation,
        dot11AuthenticateFailStatus,
        dot11AuthenticateFailStation,
        dot11MultiDomainCapabilityImplemented,
        dot11MultiDomainCapabilityActivated,
        dot11CountryString,
        dot11RSNAOptionImplemented,
        dot11OperatingClassesImplemented,
        dot11OperatingClassesRequired,
        dot11QosOptionImplemented,
        dot11ImmediateBlockAckOptionImplemented,
        dot11DelayedBlockAckOptionImplemented,
        dot11DirectOptionImplemented,
        dot11APSDOptionImplemented,
        dot11QAckOptionImplemented,
        dot11QBSSLoadImplemented,
        dot11QueueRequestOptionImplemented,
        dot11TXOPRequestOptionImplemented,
        dot11MoreDataAckOptionImplemented,
        dot11AssociateInNQBSS,
        dot11DLSAllowedInQBSS,
        dot11DLSAllowed,
        dot11AssociateStation,
        dot11AssociateID,
        dot11AssociateFailStation,
        dot11AssociateFailStatus,
        dot11ReassociateStation,
        dot11ReassociateID,
        dot11ReassociateFailStation,
        dot11ReassociateFailStatus,
        dot11RadioMeasurementImplemented,
        dot11RadioMeasurementActivated,
        dot11FastBSSTransitionImplemented,
        dot11LCIDSEImplemented,
        dot11LCIDSERequired,
        dot11DSERequired,
        dot11ExtendedChannelSwitchActivated,
        dot11HighThroughputOptionImplemented,
        dot11RSNAPBACRequired,
        dot11PSMPOptionImplemented }
    STATUS deprecated
    DESCRIPTION
        "Superceded by dot11SMTbase 11.
        The SMTbase10 object class provides the necessary support at the STA to
        manage the processes in the STA such that the STA may work cooperatively
        as a part of an IEEE 802.11 network."
    ::= { dot11Groups 47 }

dot11PhyMCSGroup OBJECT-GROUP
```

```
    OBJECTS {
        dot11SupportedMCSTxValue,
        dot11SupportedMCSRxValue }
    STATUS current
    DESCRIPTION
        "Attributes for MCS for IEEE 802.11 HT."
    ::= { dot11Groups 48 }

dot11PhyHTComplianceGroup OBJECT-GROUP
    OBJECTS {
        dot11HighThroughputOptionImplemented,
        dot11FortyMHzOperationImplemented,
        dot11FortyMHzOperationActivated,
        dot11FortyMHzIntolerant,
        dot11FortyMHzOptionImplemented,
        dot11CurrentPrimaryChannel,
        dot11CurrentSecondaryChannel,
        dot11HTGreenfieldOptionImplemented,
        dot11HTGreenfieldOptionActivated,
        dot11ShortGIOptionInTwentyImplemented,
        dot11ShortGIOptionInTwentyActivated,
        dot11ShortGIOptionInFortyImplemented,
        dot11ShortGIOptionInFortyActivated,
        dot11LDPCCodingOptionImplemented,
        dot11LDPCCodingOptionActivated,
        dot11TxSTBCOptionImplemented,
        dot11TxSTBCOptionActivated,
        dot11RxSTBCOptionImplemented,
        dot11RxSTBCOptionActivated,
        dot11BeamFormingOptionImplemented,
        dot11BeamFormingOptionActivated,
        dot11NumberOfSpatialStreamsImplemented,
        dot11NumberOfSpatialStreamsActivated,
        dot11HighestSupportedDataRate,
        dot11TxMCSSetDefined,
        dot11TxRxMCSSetNotEqual,
        dot11TxMaximumNumberSpatialStreamsSupported,
        dot11TxUnequalModulationSupported,
        dot11TransmitSoundingPPDUOptionImplemented,
        dot11NumberOfActiveRxAntennas  }
    STATUS current
    DESCRIPTION
        "Attributes that configure the HT for IEEE 802.11."
    ::= { dot11Groups 49 }

dot11HTMACAdditions OBJECT-GROUP
    OBJECTS {
        dot11HTOperationalMCSSet,
        dot11MIMOPowerSave,
        dot11NDelayedBlockAckOptionImplemented,
        dot11MaxAMSDULength,
        dot11STBCControlFrameOptionImplemented,
        dot11LsigTxopProtectionOptionImplemented,
        dot11MaxRxAMPDUFactor,
        dot11MinimumMPDUStartSpacing,
        dot11PCOOptionImplemented,
        dot11TransitionTime,
        dot11MCSFeedbackOptionImplemented,
        dot11HTControlFieldSupported,
        dot11RDResponderOptionImplemented,
        dot11BSSWidthTriggerScanInterval,
        dot11BSSWidthChannelTransitionDelayFactor,
        dot11OBSSScanPassiveDwell,
        dot11OBSSScanActiveDwell,
        dot11OBSSScanPassiveTotalPerChannel,
        dot11OBSSScanActiveTotalPerChannel,
        dot112040BSSCoexistenceManagementSupport,
        dot11OBSSScanActivityThreshold,
        dot11SPPAMSDUCapable,
        dot11SPPAMSDURequired }
    STATUS current
    DESCRIPTION
        "Attributes that configure the HT for IEEE 802.11."
    ::= { dot11Groups 50 }

dot11TransmitBeamformingGroup OBJECT-GROUP
    OBJECTS {
        dot11ReceiveStaggerSoundingOptionImplemented,
        dot11TransmitStaggerSoundingOptionImplemented,
        dot11ReceiveNDPOptionImplemented,
        dot11TransmitNDPOptionImplemented,
        dot11ImplicitTransmitBeamformingOptionImplemented,
        dot11CalibrationOptionImplemented,
        dot11ExplicitCSITransmitBeamformingOptionImplemented,
        dot11ExplicitNonCompressedBeamformingMatrixOptionImplemented,
        dot11ExplicitTransmitBeamformingCSIFeedbackOptionImplemented,
        dot11ExplicitNonCompressedBeamformingFeedbackOptionImplemented,
        dot11ExplicitCompressedBeamformingFeedbackOptionImplemented,
        dot11NumberBeamFormingCSISupportAntenna,
        dot11NumberNonCompressedBeamformingMatrixSupportAntenna,
        dot11NumberCompressedBeamformingMatrixSupportAntenna }
    STATUS current
    DESCRIPTION
        "Attributes that configure the Beamforming for IEEE 802.11 HT."
    ::= { dot11Groups 51 }


dot11SMTbase11 OBJECT-GROUP
    OBJECTS {
        dot11MediumOccupancyLimit,
        dot11CFPollable,
        dot11CFPPeriod,
        dot11CFPMaxDuration,
        dot11AuthenticationResponseTimeOut,
        dot11PrivacyOptionImplemented,
        dot11PowerManagementMode,
        dot11DesiredSSID,
        dot11DesiredBSSType,
        dot11OperationalRateSet,
        dot11BeaconPeriod,
        dot11DTIMPeriod,
        dot11AssociationResponseTimeOut,
        dot11DisassociateReason,
        dot11DisassociateStation,
        dot11DeauthenticateReason,
        dot11DeauthenticateStation,
        dot11AuthenticateFailStatus,
        dot11AuthenticateFailStation,
        dot11MultiDomainCapabilityImplemented,
        dot11MultiDomainCapabilityActivated,
        dot11CountryString,
        dot11SpectrumManagementImplemented,
        dot11SpectrumManagementRequired,
        dot11RSNAOptionImplemented,
        dot11OperatingClassesImplemented,
        dot11OperatingClassesRequired,
        dot11QosOptionImplemented,
        dot11ImmediateBlockAckOptionImplemented,
        dot11DelayedBlockAckOptionImplemented,
        dot11DirectOptionImplemented,
        dot11APSDOptionImplemented,
        dot11QAckOptionImplemented,
        dot11QBSSLoadImplemented,
        dot11QueueRequestOptionImplemented,
        dot11TXOPRequestOptionImplemented,
        dot11MoreDataAckOptionImplemented,
        dot11AssociateInNQBSS,
        dot11DLSAllowedInQBSS,
        dot11DLSAllowed,
        dot11AssociateStation,
        dot11AssociateID,
        dot11AssociateFailStation,
        dot11AssociateFailStatus,
        dot11ReassociateStation,
        dot11ReassociateID,
        dot11ReassociateFailStation,
        dot11ReassociateFailStatus,
        dot11RadioMeasurementImplemented,
        dot11RadioMeasurementActivated,
        dot11RMMeasurementProbeDelay,
        dot11RMMeasurementPilotPeriod,
        dot11RMLinkMeasurementActivated,
        dot11RMNeighborReportActivated,
        dot11RMParallelMeasurementsActivated,
        dot11RMRepeatedMeasurementsActivated,
        dot11RMBeaconPassiveMeasurementActivated,
        dot11RMBeaconActiveMeasurementActivated,
        dot11RMBeaconTableMeasurementActivated,
        dot11RMBeaconMeasurementReportingConditionsActivated,
        dot11RMFrameMeasurementActivated,
        dot11RMChannelLoadMeasurementActivated,
        dot11RMNoiseHistogramMeasurementActivated,
        dot11RMStatisticsMeasurementActivated,
        dot11RMLCIMeasurementActivated,
        dot11RMLCIAzimuthActivated,
        dot11RMTransmitStreamCategoryMeasurementActivated,
        dot11RMTriggeredTransmitStreamCategoryMeasurementActivated,
        dot11RMAPChannelReportActivated,
        dot11RMMIBActivated,
        dot11RMMaxMeasurementDuration,
        dot11RMNonOperatingChannelMaxMeasurementDuration,
        dot11RMMeasurementPilotTransmissionInformationActivated,
        dot11RMMeasurementPilotActivated,
        dot11RMNeighborReportTSFOffsetActivated,
        dot11RMRCPIMeasurementActivated,
        dot11RMRSNIMeasurementActivated,
        dot11RMBSSAverageAccessDelayActivated,
        dot11RMBSSAvailableAdmissionCapacityActivated,
        dot11FastBSSTransitionImplemented,
        dot11LCIDSEImplemented,
        dot11LCIDSERequired,
        dot11DSERequired,
        dot11ExtendedChannelSwitchActivated,
        dot11HighThroughputOptionImplemented,
        dot11WirelessManagementImplemented,
        dot11RSNAPBACRequired,
        dot11PSMPOptionImplemented }
    STATUS deprecated
    DESCRIPTION
        "Superseded by dot11SMTbase12.
        The SMTbase11 object class provides the necessary support at the STA to
        manage the processes in the STA so that the STA may work cooperatively as
        a part of an IEEE 802.11 network, when the STA is capable of multidomain
        operation. This object group should be implemented when the multidomain
        capability option is implemented."
    ::= { dot11Groups 53 }

dot11SMTWNMRequest OBJECT-GROUP
    OBJECTS {
        dot11WNMRqstRowStatus,
        dot11WNMRqstToken,
        dot11WNMRqstIfIndex,
        dot11WNMRqstType,
        dot11WNMRqstTargetAdd,
        dot11WNMRqstTimeStamp,
        dot11WNMRqstRndInterval,
        dot11WNMRqstDuration,
        dot11WNMRqstMcstGroup,
        dot11WNMRqstMcstTrigCon,
        dot11WNMRqstMcstTrigInactivityTimeout,
        dot11WNMRqstMcstTrigReactDelay,
        dot11WNMRqstLCRRqstSubject,
        dot11WNMRqstLCRIntervalUnits,
        dot11WNMRqstLCRServiceInterval,
        dot11WNMRqstLIRRqstSubject,
        dot11WNMRqstLIRIntervalUnits,
        dot11WNMRqstLIRServiceInterval,
        dot11WNMRqstEventToken,
        dot11WNMRqstEventType,
        dot11WNMRqstEventResponseLimit,
        dot11WNMRqstEventTargetBssid,
        dot11WNMRqstEventSourceBssid,
        dot11WNMRqstEventTransitTimeThresh,
        dot11WNMRqstEventTransitMatchValue,
        dot11WNMRqstEventFreqTransitCountThresh,
        dot11WNMRqstEventFreqTransitInterval,
        dot11WNMRqstEventRsnaAuthType,
        dot11WNMRqstEapType,
        dot11WNMRqstEapVendorId,
        dot11WNMRqstEapVendorType,
        dot11WNMRqstEventRsnaMatchValue,
        dot11WNMRqstEventPeerMacAddress,
        dot11WNMRqstOperatingClass,
        dot11WNMRqstChanNumber,
        dot11WNMRqstDiagToken,
        dot11WNMRqstDiagType,
        dot11WNMRqstDiagTimeout,
        dot11WNMRqstDiagBssid,
        dot11WNMRqstDiagProfileId,
        dot11WNMRqstDiagCredentials,
        dot11WNMRqstLocConfigLocIndParams,
        dot11WNMRqstLocConfigChanList,
        dot11WNMRqstLocConfigBcastRate,
        dot11WNMRqstLocConfigOptions,
        dot11WNMRqstBssTransitQueryReason,
        dot11WNMRqstBssTransitReqMode,
        dot11WNMRqstBssTransitDisocTimer,
        dot11WNMRqstBssTransitSessInfoURL,
        dot11WNMRqstBssTransitCandidateList,
        dot11WNMRqstColocInterfAutoEnable,
        dot11WNMRqstColocInterfRptTimeout,
        dot11WNMRqstVendorSpecific }
    STATUS current
    DESCRIPTION
        "The SMTWNMRequest package is a set of attributes that shall be present if
        the STA supports the WNM service."
    ::= { dot11Groups 54 }

dot11SMTWNMReport OBJECT-GROUP
    OBJECTS {
        dot11WNMVendorSpecificRprtRqstToken,
        dot11WNMVendorSpecificRprtIfIndex,
        dot11WNMVendorSpecificRprtContent,
        dot11WNMMulticastDiagnosticRprtRqstToken,
        dot11WNMMulticastDiagnosticRprtIfIndex,
        dot11WNMMulticastDiagnosticRprtMeasurementTime,
        dot11WNMMulticastDiagnosticRprtDuration,
        dot11WNMMulticastDiagnosticRprtMcstGroup,
        dot11WNMMulticastDiagnosticRprtReason,
        dot11WNMMulticastDiagnosticRprtRcvdMsduCount,
        dot11WNMMulticastDiagnosticRprtFirstSeqNumber,
        dot11WNMMulticastDiagnosticRprtLastSeqNumber,
        dot11WNMMulticastDiagnosticRprtMcstRate,
        dot11WNMLocationCivicRprtRqstToken,
        dot11WNMLocationCivicRprtIfIndex,
        dot11WNMLocationCivicRprtCivicLocation,
        dot11WNMLocationIdentifierRprtRqstToken,
        dot11WNMLocationIdentifierRprtIfIndex,
        dot11WNMLocationIdentifierRprtExpirationTSF,
        dot11WNMLocationIdentifierRprtPublicIdUri,
        dot11WNMEventTransitRprtRqstToken,
        dot11WNMEventTransitRprtIfIndex,
        dot11WNMEventTransitRprtEventStatus,
        dot11WNMEventTransitRprtEventTSF,
        dot11WNMEventTransitRprtUTCOffset,
        dot11WNMEventTransitRprtTimeError,
        dot11WNMEventTransitRprtSourceBssid,
        dot11WNMEventTransitRprtTargetBssid,
        dot11WNMEventTransitRprtTransitTime,
        dot11WNMEventTransitRprtTransitReason,
        dot11WNMEventTransitRprtTransitResult,
        dot11WNMEventTransitRprtSourceRCPI,
        dot11WNMEventTransitRprtSourceRSNI,
        dot11WNMEventTransitRprtTargetRCPI,
        dot11WNMEventTransitRprtTargetRSNI,
        dot11WNMEventRsnaRprtRqstToken,
        dot11WNMEventRsnaRprtIfIndex,
        dot11WNMEventRsnaRprtEventStatus,
        dot11WNMEventRsnaRprtEventTSF,
        dot11WNMEventRsnaRprtUTCOffset,
        dot11WNMEventRsnaRprtTimeError,
        dot11WNMEventRsnaRprtTargetBssid,
        dot11WNMEventRsnaRprtAuthType,
        dot11WNMEventRsnaRprtEapMethod,
        dot11WNMEventRsnaRprtResult,
        dot11WNMEventRsnaRprtRsnElement,
        dot11WNMEventPeerRprtRqstToken,
        dot11WNMEventPeerRprtIfIndex,
        dot11WNMEventPeerRprtEventStatus,
        dot11WNMEventPeerRprtEventTSF,
        dot11WNMEventPeerRprtUTCOffset,
        dot11WNMEventPeerRprtTimeError,
        dot11WNMEventPeerRprtPeerMacAddress,
        dot11WNMEventPeerRprtOperatingClass,
        dot11WNMEventPeerRprtChanNumber,
        dot11WNMEventPeerRprtStaTxPower,
        dot11WNMEventPeerRprtConnTime,
        dot11WNMEventPeerRprtPeerStatus,
        dot11WNMEventWNMLogRprtRqstToken,
        dot11WNMEventWNMLogRprtIfIndex,
        dot11WNMEventWNMLogRprtEventStatus,
        dot11WNMEventWNMLogRprtEventTSF,
        dot11WNMEventWNMLogRprtUTCOffset,
        dot11WNMEventWNMLogRprtTimeError,
        dot11WNMEventWNMLogRprtContent,
        dot11WNMDiagMfrInfoRprtRqstToken,
        dot11WNMDiagMfrInfoRprtIfIndex,
        dot11WNMDiagMfrInfoRprtEventStatus,
        dot11WNMDiagMfrInfoRprtMfrOi,
        dot11WNMDiagMfrInfoRprtMfrIdString,
        dot11WNMDiagMfrInfoRprtMfrModelString,
        dot11WNMDiagMfrInfoRprtMfrSerialNumberString,
        dot11WNMDiagMfrInfoRprtMfrFirmwareVersion,
        dot11WNMDiagMfrInfoRprtMfrAntennaType,
        dot11WNMDiagMfrInfoRprtCollocRadioType,
        dot11WNMDiagMfrInfoRprtDeviceType,
        dot11WNMDiagMfrInfoRprtCertificateID,
        dot11WNMDiagConfigProfRprtRqstToken,
        dot11WNMDiagConfigProfRprtIfIndex,
        dot11WNMDiagConfigProfRprtEventStatus,
        dot11WNMDiagConfigProfRprtProfileId,
        dot11WNMDiagConfigProfRprtSupportedOperatingClasses,
        dot11WNMDiagConfigProfRprtTxPowerMode,
        dot11WNMDiagConfigProfRprtTxPowerLevels,
        dot11WNMDiagConfigProfRprtCipherSuite,
        dot11WNMDiagConfigProfRprtAkmSuite,
        dot11WNMDiagConfigProfRprtEapType,
        dot11WNMDiagConfigProfRprtEapVendorID,
        dot11WNMDiagConfigProfRprtEapVendorType,
        dot11WNMDiagConfigProfRprtCredentialType,
        dot11WNMDiagConfigProfRprtSSID,
        dot11WNMDiagConfigProfRprtPowerSaveMode,
        dot11WNMDiagAssocRprtRqstToken,
        dot11WNMDiagAssocRprtIfIndex,
        dot11WNMDiagAssocRprtEventStatus,
        dot11WNMDiagAssocRprtBssid,
        dot11WNMDiagAssocRprtOperatingClass,
        dot11WNMDiagAssocRprtChannelNumber,
        dot11WNMDiagAssocRprtStatusCode,
        dot11WNMDiag8021xAuthRprtRqstToken,
        dot11WNMDiag8021xAuthRprtIfIndex,
        dot11WNMDiag8021xAuthRprtEventStatus,
        dot11WNMDiag8021xAuthRprtBssid,
        dot11WNMDiag8021xAuthRprtOperatingClass,
        dot11WNMDiag8021xAuthRprtChannelNumber,
        dot11WNMDiag8021xAuthRprtEapType,
        dot11WNMDiag8021xAuthRprtEapVendorID,
        dot11WNMDiag8021xAuthRprtEapVendorType,
        dot11WNMDiag8021xAuthRprtCredentialType,
        dot11WNMDiag8021xAuthRprtStatusCode,
        dot11WNMLocConfigRprtRqstToken,
        dot11WNMLocConfigRprtIfIndex,
        dot11WNMLocConfigRprtLocIndParams,
        dot11WNMLocConfigRprtLocIndChanList,
        dot11WNMLocConfigRprtLocIndBcastRate,
        dot11WNMLocConfigRprtLocIndOptions,
        dot11WNMLocConfigRprtStatusConfigSubelemId,
        dot11WNMLocConfigRprtStatusResult,
        dot11WNMLocConfigRprtVendorSpecificRprtContent,
        dot11WNMBssTransitRprtRqstToken,
        dot11WNMBssTransitRprtIfIndex,
        dot11WNMBssTransitRprtStatusCode,
        dot11WNMBssTransitRprtBSSTerminationDelay,
        dot11WNMBssTransitRprtTargetBssid,
        dot11WNMBssTransitRprtCandidateList,
        dot11WNMColocInterfRprtRqstToken,
        dot11WNMColocInterfRprtIfIndex,
        dot11WNMColocInterfRprtPeriod,
        dot11WNMColocInterfRprtInterfLevel,
        dot11WNMColocInterfRprtInterfAccuracy,
        dot11WNMColocInterfRprtInterfIndex,
        dot11WNMColocInterfRprtInterfInterval,
        dot11WNMColocInterfRprtInterfBurstLength,
        dot11WNMColocInterfRprtInterfStartTime,
        dot11WNMColocInterfRprtInterfCenterFreq,
        dot11WNMColocInterfRprtInterfBandwidth }
    STATUS current
    DESCRIPTION
        "The SMTWNMReport package is a set of attributes that shall be present if
        the STA supports the WNM service."
    ::= { dot11Groups 55 }

dot11SMTbase12 OBJECT-GROUP
    OBJECTS {
        dot11MediumOccupancyLimit,
        dot11CFPollable,
        dot11CFPPeriod,
        dot11CFPMaxDuration,
        dot11AuthenticationResponseTimeOut,
        dot11PrivacyOptionImplemented,
        dot11PowerManagementMode,
        dot11DesiredSSID,
        dot11DesiredBSSType,
        dot11OperationalRateSet,
        dot11BeaconPeriod,
        dot11DTIMPeriod,
        dot11AssociationResponseTimeOut,
        dot11DisassociateReason,
        dot11DisassociateStation,
        dot11DeauthenticateReason,
        dot11DeauthenticateStation,
        dot11AuthenticateFailStatus,
        dot11AuthenticateFailStation,
        dot11MultiDomainCapabilityImplemented,
        dot11MultiDomainCapabilityActivated,
        dot11CountryString,
        dot11SpectrumManagementImplemented,
        dot11SpectrumManagementRequired,
        dot11RSNAOptionImplemented,
        dot11OperatingClassesImplemented,
        dot11OperatingClassesRequired,
        dot11QosOptionImplemented,
        dot11ImmediateBlockAckOptionImplemented,
        dot11DelayedBlockAckOptionImplemented,
        dot11DirectOptionImplemented,
        dot11APSDOptionImplemented,
        dot11QAckOptionImplemented,
        dot11QBSSLoadImplemented,
        dot11QueueRequestOptionImplemented,
        dot11TXOPRequestOptionImplemented,
        dot11MoreDataAckOptionImplemented,
        dot11AssociateInNQBSS,
        dot11DLSAllowedInQBSS,
        dot11DLSAllowed,
        dot11AssociateStation,
        dot11AssociateID,
        dot11AssociateFailStation,
        dot11AssociateFailStatus,
        dot11ReassociateStation,
        dot11ReassociateID,
        dot11ReassociateFailStation,
        dot11ReassociateFailStatus,
        dot11RadioMeasurementImplemented,
        dot11RadioMeasurementActivated,
        dot11RMMeasurementProbeDelay,
        dot11RMMeasurementPilotPeriod,
        dot11RMLinkMeasurementActivated,
        dot11RMNeighborReportActivated,
        dot11RMParallelMeasurementsActivated,
        dot11RMRepeatedMeasurementsActivated,
        dot11RMBeaconPassiveMeasurementActivated,
        dot11RMBeaconActiveMeasurementActivated,
        dot11RMBeaconTableMeasurementActivated,
        dot11RMBeaconMeasurementReportingConditionsActivated,
        dot11RMFrameMeasurementActivated,
        dot11RMChannelLoadMeasurementActivated,
        dot11RMNoiseHistogramMeasurementActivated,
        dot11RMStatisticsMeasurementActivated,
        dot11RMLCIMeasurementActivated,
        dot11RMLCIAzimuthActivated,
        dot11RMTransmitStreamCategoryMeasurementActivated,
        dot11RMTriggeredTransmitStreamCategoryMeasurementActivated,
        dot11RMAPChannelReportActivated,
        dot11RMMIBActivated,
        dot11RMMaxMeasurementDuration,
        dot11RMNonOperatingChannelMaxMeasurementDuration,
        dot11RMMeasurementPilotTransmissionInformationActivated,
        dot11RMMeasurementPilotActivated,
        dot11RMNeighborReportTSFOffsetActivated,
        dot11RMRCPIMeasurementActivated,
        dot11RMRSNIMeasurementActivated,
        dot11RMBSSAverageAccessDelayActivated,
        dot11RMBSSAvailableAdmissionCapacityActivated,
        dot11FastBSSTransitionImplemented,
        dot11LCIDSEImplemented,
        dot11LCIDSERequired,
        dot11DSERequired,
        dot11ExtendedChannelSwitchActivated,
        dot11HighThroughputOptionImplemented,
        dot11WirelessManagementImplemented,
        dot11MeshActivated,
        dot11RSNAPBACRequired,
        dot11PSMPOptionImplemented }
    STATUS current
    DESCRIPTION
        "The SMTbase12 object class provides the necessary support at the STA to
        manage the processes in the STA such that the STA may work cooperatively
        as a part of an IEEE 802.11 network."
    ::= { dot11Groups 57 }


dot11OperatingClassesGroup OBJECT-GROUP
    OBJECTS {
        dot11OperatingClass,
        dot11CoverageClass }
    STATUS current
    DESCRIPTION
        "Attributes that configure the OFDM for IEEE 802.11 in many regulatory
        domains."
    ::= { dot11Groups 58 }

dot11PhyOperationComplianceGroup2 OBJECT-GROUP
```

```
    OBJECTS { dot11PHYType, dot11CurrentRegDomain }
    STATUS current
    DESCRIPTION
        "PHY layer operations attributes."
    ::= { dot11Groups 59 }

dot11MeshComplianceGroup OBJECT-GROUP
    OBJECTS {
        -- dot11MeshSTAConfigTable
        dot11MeshID,
        dot11MeshNumberOfPeerings,
        dot11MeshAcceptingAdditionalPeerings,
        dot11MeshConnectedToMeshGate,
        dot11MeshSecurityActivated,
        dot11MeshActiveAuthenticationProtocol,
        dot11MeshMaxRetries,
        dot11MeshRetryTimeout,
        dot11MeshConfirmTimeout,
        dot11MeshHoldingTimeout,
        dot11MeshActivePathSelectionProtocol,
        dot11MeshActivePathSelectionMetric,
        dot11MeshForwarding,
        dot11MeshTTL,
        dot11MeshGateAnnouncements,
        dot11MeshActiveCongestionControlMode,
        dot11MeshActiveSynchronizationMethod,
        dot11MeshNbrOffsetMaxNeighbor,
        dot11MBCAActivated,
        dot11MCCAImplemented,
        dot11MCCAActivated }
    STATUS current
    DESCRIPTION
        "This object class provides the objects from the IEEE 802.11 MIB required
        to manage mandatory mesh functionality. Note that additional objects for
        managing mesh functionality are located in the dot11MeshOptionGroup,
        dot11MeshHWMPComplianceGroup, and dot11PasswordAuthComplianceGroup."
    ::= { dot11Groups 56}


dot11MeshOptionGroup OBJECT-GROUP
    OBJECTS {
        -- dot11MeshSTAConfigTable
        dot11MeshConfigGroupUpdateCount,
        dot11MeshGateAnnouncementInterval,
        dot11MeshBeaconTimingReportInterval,
        dot11MeshBeaconTimingReportMaxNum,
        dot11MeshDelayedBeaconTxInterval,
        dot11MeshDelayedBeaconTxMaxDelay,
        dot11MeshDelayedBeaconTxMinDelay,
        dot11MeshAverageBeaconFrameDuration,
        dot11MeshSTAMissingAckRetryLimit,
        dot11MeshAwakeWindowDuration,
        dot11MAFlimit,
        dot11MCCAScanDuration,
        dot11MCCAAdvertPeriodMax,
        dot11MCCAMinTrackStates,
        dot11MCCAMaxTrackStates,
        dot11MCCAOPtimeout,
        dot11MCCACWmin,
        dot11MCCACWmax,
        dot11MCCAAIFSN
        }
    STATUS current
    DESCRIPTION
```

```
        "This object class provides the objects from the IEEE 802.11 MIB required
        to manage optional mesh functionality. Note that other objects for manag-
        ing mesh functionality are located in the dot11MeshComplianceGroup,
        dot11MeshHWMPComplianceGroup, and dot11PasswordAuthComplianceGroup."
    ::= { dot11Groups 60 }


dot11MeshHWMPComplianceGroup OBJECT-GROUP
    OBJECTS {
        -- dot11MeshHWMPConfigTable
        dot11MeshHWMPmaxPREQretries,
        dot11MeshHWMPnetDiameter,
        dot11MeshHWMPnetDiameterTraversalTime,
        dot11MeshHWMPpreqMinInterval,
        dot11MeshHWMPperrMinInterval,
        dot11MeshHWMPactivePathToRootTimeout,
        dot11MeshHWMPactivePathTimeout,
        dot11MeshHWMProotMode,
        dot11MeshHWMProotInterval,
        dot11MeshHWMPrannInterval,
        dot11MeshHWMPtargetOnly,
        dot11MeshHWMPmaintenanceInterval,
        dot11MeshHWMPconfirmationInterval }
    STATUS current
    DESCRIPTION
        "This object class provides the objects from the IEEE 802.11 MIB required
        to manage HWMP path selection functionality. Note that other objects for
        managing mesh functionality are located in the dot11MeshComplianceGroup,
        dot11MeshOptionGroup, and dot11PasswordAuthComplianceGroup."
    ::= { dot11Groups 61 }


dot11PasswordAuthComplianceGroup OBJECT-GROUP
    OBJECTS {
        -- dot11RSNAConfigTable
        dot11RSNASAERetransPeriod,
        dot11RSNASAEAntiCloggingThreshold,
        dot11RSNASAESync,
        -- dot11RSNAConfigPasswordValueTable
--      dot11RSNAConfigPasswordValueIndex,
        dot11RSNAConfigPasswordCredential,
        dot11RSNAConfigPasswordPeerMac,
        -- dot11RSNAConfigDLCGroupTable
--      dot11RSNAConfigDLCGroupIndex,
        dot11RSNAConfigDLCGroupIdentifier }
    STATUS current
    DESCRIPTION
        "This object class provides the objects from the IEEE 802.11 MIB required
        to manage password authentication. Note that other objects for managing
        mesh functionality are located in the dot11MeshComplianceGroup,
        dot11MeshOptionGroup, and dot11MeshHWMPComplianceGroup."
    ::= { dot11Groups 62 }


dot11SpectrumManagementGroup OBJECT-GROUP
    OBJECTS {
        -- Dot11SpectrumManagementEntry
        -- dot11SpectrumManagementIndex,
        dot11MitigationRequirement,
        dot11ChannelSwitchTime,
        dot11PowerCapabilityMaxImplemented,
        dot11PowerCapabilityMinImplemented  }
    STATUS current
    DESCRIPTION
```

```
        "This object class provides the objects from the IEEE 802.11 MIB for spec-
        trum management."
    ::= { dot11Groups 72 }

dot11ProtectedManagementFrameGroup OBJECT-GROUP
    OBJECTS {
        -- Dot11StationConfigEntry
        dot11RSNAProtectedManagementFramesActivated,
        dot11RSNAUnprotectedManagementFramesAllowed,
        dot11AssociationSAQueryMaximumTimeout,
        dot11AssociationSAQueryRetryTimeout,
        -- dot11RSNAStatsEntry
        dot11RSNAStatsCMACICVErrors,
        dot11RSNAStatsCMACReplays,
        dot11RSNAStatsRobustMgmtCCMPReplays,
        dot11RSNABIPMICErrors  }
    STATUS current
    DESCRIPTION
        "This object class provides the objects from the IEEE 802.11 MIB required
        to operate protected management frame."
    ::= { dot11Groups 73 }

dot11LCIDSEGroup OBJECT-GROUP
    OBJECTS {
        -- Dot11LCIDSEEntry
        -- dot11LCIDSEIndex,
        dot11LCIDSEIfIndex,
        dot11LCIDSECurrentOperatingClass,
        dot11LCIDSELatitudeResolution,
        dot11LCIDSELatitudeInteger,
        dot11LCIDSELatitudeFraction,
        dot11LCIDSELongitudeResolution,
        dot11LCIDSELongitudeInteger,
        dot11LCIDSELongitudeFraction,
        dot11LCIDSEAltitudeType,
        dot11LCIDSEAltitudeResolution,
        dot11LCIDSEAltitudeInteger,
        dot11LCIDSEAltitudeFraction,
        dot11LCIDSEDatum,
        dot11RegLocAgreement,
        dot11RegLocDSE,
        dot11DependentSTA,
        dot11DependentEnablementIdentifier,
        dot11DSEEnablementTimeLimit,
        dot11DSEEnablementFailHoldTime,
        dot11DSERenewalTime,
        dot11DSETransmitDivisor }
    STATUS current
    DESCRIPTION
        "This object class provides the objects from the IEEE 802.11 MIB required
        to enable 3650-3700 MHz Operation in USA."
    ::= { dot11Groups 74 }

dot11TDLSComplianceGroup OBJECT-GROUP
    OBJECTS {
        -- Dot11StationConfigEntry
        dot11TunneledDirectLinkSetupImplemented,
        dot11TDLSPeerUAPSDBufferSTAActivated,
        dot11TDLSPeerPSMActivated,
        dot11TDLSPeerUAPSDIndicationWindow,
        dot11TDLSChannelSwitchingActivated,
        dot11TDLSPeerSTAMissingAckRetryLimit,
        dot11TDLSResponseTimeout,
        dot11OCBActivated,
        dot11TDLSProbeDelay,
        dot11TDLSDiscoveryRequestWindow,
        dot11TDLSACDeterminationInterval }
    STATUS current
    DESCRIPTION
        "This object class provides the objects from the IEEE 802.11 MIB required
        to operate tunneled direct link setup."
    ::= { dot11Groups 75 }


-- **********************************************************************
-- * Compliance Statements
-- **********************************************************************

dot11Compliance MODULE-COMPLIANCE
    STATUS  current
    DESCRIPTION
        "The compliance statement for SNMPv2 entities that implement the IEEE
        802.11 MIB."
    MODULE  -- this module
    MANDATORY-GROUPS {
        dot11SMTbase12,
        dot11MACbase3,
        dot11CountersGroup3,
        dot11SmtAuthenticationAlgorithms,
        dot11ResourceTypeID,
        dot11PhyOperationComplianceGroup2 }

    GROUP dot11PhyDSSSComplianceGroup
    DESCRIPTION
        "Implementation of this group is required when object dot11PHYType is
        dsss.
        This group is mutually exclusive to the following groups:
        dot11PhyIRComplianceGroup
        dot11PhyFHSSComplianceGroup2
        dot11PhyOFDMComplianceGroup3
        dot11PhyHRDSSSComplianceGroup
        dot11PhyERPComplianceGroup
        dot11PhyHTComplianceGroup"

    GROUP dot11PhyIRComplianceGroup
    DESCRIPTION
        "Implementation of this group is required when object dot11PHYType is
        irbaseband.
        This group is mutually exclusive to the following groups:
        dot11PhyFHSSComplianceGroup2
        dot11PhyDSSSComplianceGroup
        dot11PhyOFDMComplianceGroup3
        dot11PhyHRDSSSComplianceGroup
        dot11PhyERPComplianceGroup
        dot11PhyHTComplianceGroup"

    GROUP dot11PhyFHSSComplianceGroup2
    DESCRIPTION
        "Implementation of this group is required when object dot11PHYType is
        fhss.
        This group is mutually exclusive to the following groups:
        dot11PhyIRComplianceGroup
        dot11PhyDSSSComplianceGroup
        dot11PhyOFDMComplianceGroup3
        dot11PhyHRDSSSComplianceGroup
        dot11PhyERPComplianceGroup
        dot11PhyHTComplianceGroup"
```

```
GROUP dot11PhyOFDMComplianceGroup3
DESCRIPTION
   "Implementation of this group is required when object dot11PHYType is
   ofdm.
   This group is mutually exclusive to the following groups:
   dot11PhyIRComplianceGroup
   dot11PhyFHSSComplianceGroup2
   dot11PhyDSSSComplianceGroup
   dot11PhyHRDSSSComplianceGroup
   dot11PhyERPComplianceGroup
   dot11PhyHTComplianceGroup"

GROUP dot11PhyHRDSSSComplianceGroup
DESCRIPTION
   "Implementation of this group is required when object dot11PHYType is
   hrdsss.
   This group is mutually exclusive to the following groups:
   dot11PhyIRComplianceGroup
   dot11PhyFHSSComplianceGroup2
   dot11PhyDSSSComplianceGroup
   dot11PhyOFDMComplianceGroup3
   dot11PhyERPComplianceGroup
   dot11PhyHTComplianceGroup"

GROUP dot11PhyERPComplianceGroup
DESCRIPTION
   "Implementation of this group is required when object dot11PHYType is ERP.
   This group is mutually exclusive to the following groups:
   dot11PhyIRComplianceGroup
   dot11PhyFHSSComplianceGroup2
   dot11PhyDSSSComplianceGroup
   dot11PhyOFDMComplianceGroup3
   dot11PhyHRDSSSComplianceGroup
   dot11PhyHTComplianceGroup"

GROUP dot11PhyHTComplianceGroup
DESCRIPTION
   "Implementation of this group is required when object dot11PHYType has the
   value of ht.
   This group is mutually exclusive to the following groups:
   dot11PhyIRComplianceGroup
   dot11PhyFHSSComplianceGroup2
   dot11PhyDSSSComplianceGroup
   dot11PhyOFDMComplianceGroup3
   dot11PhyHRDSSSComplianceGroup
   dot11PhyERPComplianceGroup"

GROUP dot11HTMACAdditions
DESCRIPTION
   "The dot11HTMACAdditions group is optional."

GROUP dot11SMTprivacy
DESCRIPTION
   "The dot11SMTprivacy group is optional."

GROUP dot11MACStatistics
DESCRIPTION
   "The dot11MACStatistics group is optional."

GROUP dot11PhyTxPowerComplianceGroup
DESCRIPTION
   "The dot11PhyTxPowerComplianceGroup group is optional."

GROUP dot11PhyRegDomainsSupportGroup
```

```
    DESCRIPTION
       "The dot11PhyRegDomainsSupportGroup group is optional."

    GROUP dot11PhyAntennasListGroup
    DESCRIPTION
       "The dot11PhyAntennasListGroup group is optional."

    GROUP dot11PhyRateGroup
    DESCRIPTION
       "The dot11PhyRateGroup group is optional."

    GROUP dot11MultiDomainCapabilityGroup
    DESCRIPTION
       "The dot11MultiDomainCapabilityGroup group is optional."

    GROUP dot11RSNAadditions
    DESCRIPTION
       "The dot11RSNAadditions group is optional."

    GROUP dot11OperatingClassesGroup
    DESCRIPTION
       "The dot11OperatingClassesGroup group is optional."

    GROUP dot11Qosadditions
    DESCRIPTION
       "The dot11Qosadditions group is optional."

    GROUP dot11FTComplianceGroup
    DESCRIPTION
       "The dot11FTComplianceGroup group is optional."

    GROUP dot11PhyAntennaComplianceGroup2
    DESCRIPTION
       "The dot11PhyAntennaComplianceGroup2 group is optional."

    GROUP dot11PhyMCSGroup
    DESCRIPTION
       "The dot11PhyMCSGroup group is optional."

    GROUP dot11TransmitBeamformingGroup
    DESCRIPTION
       "The dot11TransmitBeamformingGroup group is optional."

--  OPTIONAL-GROUPS {
    --  dot11SMTprivacy,
    --  dot11MACStatistics,
    --  dot11PhyTxPowerComplianceGroup,
    --  dot11PhyRegDomainsSupportGroup,
    --  dot11PhyAntennasListGroup,
    --  dot11PhyRateGroup,
    --  dot11MultiDomainCapabilityGroup,
    --  dot11PhyFHSSComplianceGroup2,
    --  dot11RSNAadditions,
    --  dot11OperatingClassesGroup,
    --  dot11Qosadditions,
    --  dot11RMCompliance,
    --  dot11FTComplianceGroup,
    --  dot11PhyAntennaComplianceGroup2,
    --  dot11HTMACadditions,
    --  dot11PhyMCSGroup,
    --  dot11TransmitBeamformingGroup,
    --  dot11WNMCompliance}

    ::= { dot11Compliances 1 }
```

```
-- **********************************************************************
-- * Compliance Statements - RSN
-- **********************************************************************
dot11RSNCompliance MODULE-COMPLIANCE
   STATUS current
   DESCRIPTION
      "The compliance statement for SNMPv2 entities that implement the IEEE
      802.11 RSN MIB."
   MODULE -- this module
   MANDATORY-GROUPS { dot11RSNBase }
-- OPTIONAL-GROUPS { dot11RSNPMKcachingGroup }
   ::= { dot11Compliances 2 }

-- **********************************************************************
-- * Compliance Statements - RM
-- **********************************************************************
dot11RMCompliance MODULE-COMPLIANCE
   STATUS current
   DESCRIPTION
      "The compliance statement for SNMPv2 entities that implement the IEEE
      802.11 MIB for Measurement Services."
   MODULE -- this module
   MANDATORY-GROUPS {
      dot11SMTRMRequest,
      dot11SMTRMReport,
      dot11SMTRMConfig }
-- OPTIONAL-GROUPS { }
   ::= { dot11Compliances 3 }

-- **********************************************************************
-- * Compliance Statements - Mesh
-- **********************************************************************

dot11MeshCompliance MODULE-COMPLIANCE
   STATUS current
   DESCRIPTION
      "The compliance statement for SNMPv2 entities that implement the IEEE
      802.11 MIB for Mesh."
   MODULE -- this module
   MANDATORY-GROUPS {
      dot11MeshComplianceGroup,
      dot11MeshHWMPComplianceGroup,
      dot11PasswordAuthComplianceGroup }
-- OPTIONAL-GROUPS { dot11MeshOptionGroup }
   ::= { dot11Compliances 4 }

-- **********************************************************************
-- * Compliance Statements - WNM
-- **********************************************************************
dot11WNMCompliance MODULE-COMPLIANCE
   STATUS current
   DESCRIPTION
      " This object class provides the objects from the IEEE 802.11
      MIB required to manage wireless network management
      functionality. Note that additional objects for managing this
      functionality are located in the IEEE 802.11 WNM MIB."

   MODULE -- this module
   MANDATORY-GROUPS { dot11SMTRMRequest, dot11SMTRMReport, dot11SMTRMConfig }

   GROUP dot11SMTbase12
   DESCRIPTION "At least the dot11WirelessManagementImplemented object is
      required from dot11SMTbase12"
   OBJECT dot11WirelessManagementImplemented
   DESCRIPTION "Required object"
   ::= { dot11Compliances 5 }

-- ********************************************************************
-- * Compliance Statements - Spectrum Management
-- ********************************************************************
dot11SpectrumManagementCompliance MODULE-COMPLIANCE
   STATUS current
   DESCRIPTION
      "This object class provides the objects from the IEEE 802.11
      MIB used to operate higher throughput."
   MODULE -- this module
   MANDATORY-GROUPS { dot11SpectrumManagementGroup }
-- OPTIONAL-GROUPS { }
   ::= { dot11Compliances 10 }

-- ********************************************************************
-- * Compliance Statements - Protected management frame
-- ********************************************************************
dot11ProtectedManagementFrameCompliance MODULE-COMPLIANCE
   STATUS current
   DESCRIPTION
      "This object class provides the objects from the IEEE 802.11
      MIB required to operate protected management frames."
   MODULE -- this module
   MANDATORY-GROUPS { dot11ProtectedManagementFrameGroup }
-- OPTIONAL-GROUPS { }
   ::= { dot11Compliances 11 }

-- ********************************************************************
-- * Compliance Statements - LCIDSE
-- ********************************************************************
dot11LCIDSECompliance MODULE-COMPLIANCE
   STATUS current
   DESCRIPTION
      "This object class provides the objects from the IEEE 802.11
      MIB required to enable 3650-3700 MHz Operation in USA."
   MODULE -- this module
   MANDATORY-GROUPS { dot11LCIDSEGroup }
-- OPTIONAL-GROUPS { }
   ::= { dot11Compliances 12 }

-- ********************************************************************
-- * Compliance Statements - TDLS
-- ********************************************************************
dot11TDLSCompliance MODULE-COMPLIANCE
   STATUS current
   DESCRIPTION
      "This object class provides the objects from the IEEE 802.11
      MIB required to manage tunnled direct link setup."
   MODULE -- this module
   MANDATORY-GROUPS { dot11TDLSComplianceGroup }
-- OPTIONAL-GROUPS {  }
   ::= { dot11Compliances 13 }

-- **********************************************************************
-- *   End of 802.11 MIB
-- **********************************************************************

   END
```

Annex D

(normative)

Regulatory references

D.1 External regulatory references

This annex and Annex E provide information and specifications for operation in many regulatory domains.

WLANs implemented in accordance with this standard and the specifications and definitions referenced in it are subject to equipment certification and operating requirements established by regional and national regulatory administrations. The specification establishes minimum technical requirements for interoperability, based upon established regulations at the time this standard was issued. These regional and national regulations are subject to revision or may be superseded. Regulatory requirements that do not affect interoperability are not addressed in this standard. Implementers are referred to the regulatory sources in Table D-1 for further information. Operation in countries within defined regulatory domains may be subject to additional or alternative national regulations.

The documents listed in Table D-1 specify current regulatory requirements for various frequency bands and geographic areas at the time this standard was developed. They are provided for information only and are subject to change or revision at any time.

Table D-1—Regulatory requirement list

Geographic area	Approval standards	Documents	Approval authority
Japan	Ministry of Internal Affairs and Communications (MIC)	MIC Equipment Ordinance (EO) for Regulating Radio Equipment Articles 7, 49.20, 49.21[a]	MIC
United States	Federal Communications Commission (FCC)	47 CFR [B9], Part 15, Sections 15.205, 15.209, and 15.247; and Subpart E, Sections 15.401–15.407, Section 90.210, Sections 90.371–383, Sections 90.1201–90.1217, Sections 90.1301–90.1337, Section 95.639, Sections 95.1501–1511	FCC
Europe	European Conference of Postal and Telecommunications (CEPT) Administrations and its Electronic Communications Committee (ECC). Also, European Radiocommunications Office, European Telecommunications Standards Institute (ETSI)	ECC DEC (04) 08, ETSI EN 300 328 [B13], ETSI EN 301 893, ETSI ES 202 663 [B15], ETSI EN 302 571 [B14], Clause 5	CEPT
China	Ministry of Industry and Information Technology (MIIT)	Xin Bu Wu [2002] #353, Xin Bu Wu [2002] #277	MIIT

[a] Frequency planning for licensed STAs in Japan is performed by the regulatory authority and the licensees, addressing the coexistence among STAs operating with a variety of air propagation times and the coexistence between STAs using 20 MHz channel spacing, STAs operating with 10 MHz channel spacing, and STAs operating with 5 MHz channel spacing. Note also the CCA mechanism is preserved in licensed operation.

Behavior limits sets are listed in Table D-2.

Table D-2—Behavior limits sets

Encoding	Behavior limits set	Description
0		Not specified
1	NomadicBehavior	The location of the station may change but is stationary while in use. The nomadic use EIRP power limits apply if the country allows more than one transmit power limit in the band. This behavior is only specified in bands where behavior 10 License Exempt bands is also allowed.
2–9	Reserved	Reserved
10	LicenseExemptBehavior	Frequency bands where some fixed stations can be operated without a license at a higher radiated transmit power than permitted for nomadic use. This behavior is only specified in bands where behavior 1 nomadic use is also allowed.
11–12	Reserved	
13	PrimaryChannelLowerBehavior[a]	20/40 MHz BSS primary channel with secondary channel above the primary channel or 20 MHz BSS primary channel operated by an FC HT AP and also 20 MHz operational channel for a non-AP STA when the non-AP STA is associated with an FC HT AP. See NOTE.
14	PrimaryChannelUpperBehavior[a]	14 20/40 MHz BSS primary channel with secondary channel below the primary channel or 20 MHz BSS primary channel operated by an FC HT AP and also 20 MHz operational channel for a non-AP STA when the non-AP STA is associated with an FC HT AP. See NOTE.
15	CCA-EDBehavior[b]	CCA shall also detect a medium busy condition when CCA-EnergyDetect detects a channel busy condition.
16	DFS_50_100_Behavior	A station operating in a band where radiolocation radar is primary, and station operation has in-service monitoring requirements for 50-100 µs radar pulses.
17	ITS_nonmobile_operations	Operations related to an ITS station operating at a fixed location registered with regulatory authorities, e.g., a Dedicated Short Range Communication Services (DSRCS) Roadside Unit (RSU).
18	ITS_mobile_operations	Operations related to an ITS mobile station, e.g., a vehicle's DSRCS On-board Unit (OBU).
19-255	Reserved	

NOTE—The fields that specify the 40 MHz channels are described in 20.3.15.4.

[a] For 20 MHz operation where the operating class signifies 40 MHz channel spacing, the 20 MHz channel corresponds to the channel number indicated.

[b] Procedures that may be used to improve sharing spectrum in addition to explicit regulatory requirements.

D.2 Radio performance specifications

D.2.1 Transmit and receive in-band and out-of-band spurious emissions

Spurious transmissions from compliant devices shall conform to national regulations.

D.2.2 Transmit power levels

The maximum allowable output power is measured in accordance with practices specified by the appropriate regulatory bodies.

The maximum allowable STA transmit power classifications for ITS nonmobile operations in the U.S. 5.85–5.925 GHz band are shown in Table D-3.

Table D-3—Maximum STA transmit power classification for the 5.85–5.925 GHz band in the United States

STA transmit power classification	Maximum STA transmit power (mW)	Maximum permitted EIRP (dBm)
A	1	23
B	10	23
C	100	33
D	760 Note that for this class higher power is permitted as long as the power level is reduced to this level at the antenna input and the emission mask specifications are met.	33 for nongovernment 44.8 for government

D.2.3 Transmit spectrum mask

Transmit spectrum masks defined in regulation are subject to change or revision at any time.

For operation in the 5.85–5.925 GHz band the transmitted spectrum shall be as follows:

a) For any STA using 5 MHz channel spacing, the transmitted spectral density shall have a 0 dBr bandwidth not exceeding 4.5 MHz and shall not exceed the spectrum mask created using the permitted power spectral density levels listed in Table D-4 for the transmit power class of the STA.

b) For any STA using 10 MHz channel spacing, the transmitted spectral density shall have a 0 dBr bandwidth not exceeding 9 MHz and shall not exceed the spectrum mask created using the permitted power spectral density levels listed in Table D-5 for the transmit power class of the STA.

c) For any STA using 20 MHz channel spacing, the transmitted spectral density shall have a 0 dBr bandwidth not exceeding 18 MHz and shall not exceed the spectrum mask created using the permitted power spectral density levels listed in Table D-6 for the transmit power class of the STA.

Table D-4—Spectrum mask data for 5 MHz channel spacing

STA transmit power class	Permitted power spectral density, dBr				
	± 2.25 MHz offset (±f1)	± 2.5 MHz offset (±f2)	± 2.75 MHz offset (±f3)	±5 MHz offset (±f4)	± 7.5 MHz offset (±f5)
Class A	0	–10	–20	–28	–40
Class B	0	–16	–20	–28	–40
Class C	0	–26	–32	–40	–50
Class D	0	–35	–45	–55	–65

Table D-5—Spectrum mask data for 10 MHz channel spacing

STA transmit power class	Permitted power spectral density, dBr				
	± 4.5 MHz offset (±f1)	± 5.0 MHz offset (±f2)	± 5.5 MHz offset (±f3)	± 10 MHz offset (±f4)	± 15 MHz offset (±f5)
Class A	0	–10	–20	–28	–40
Class B	0	–16	–20	–28	–40
Class C	0	–26	–32	–40	–50
Class D	0	–35	–45	–55	–65

Table D-6—Spectrum mask data for 20 MHz channel spacing

STA transmit power class	Permitted power spectral density, dBr				
	± 9 MHz offset (±f1)	± 10.0 MHz offset (±f2)	± 11 MHz offset (±f3)	± 20 MHz offset (±f4)	± 30 MHz offset (±f5)
Class A	0	–10	–20	–28	–40
Class B	0	–16	–20	–28	–40
Class C	0	–26	–32	–40	–50
Class D	0	–35	–45	–55	–65

The transmit spectral mask is created and applied as shown in Figure D-1 about the channel center frequency (Fc) defined by the channel starting frequency and channel number from the operating class. The 0 dBr level is the maximum power spectral density measured in the channel. The measurements of transmit spectral density are made using a 100 kHz resolution bandwidth and a 30 kHz video bandwidth.

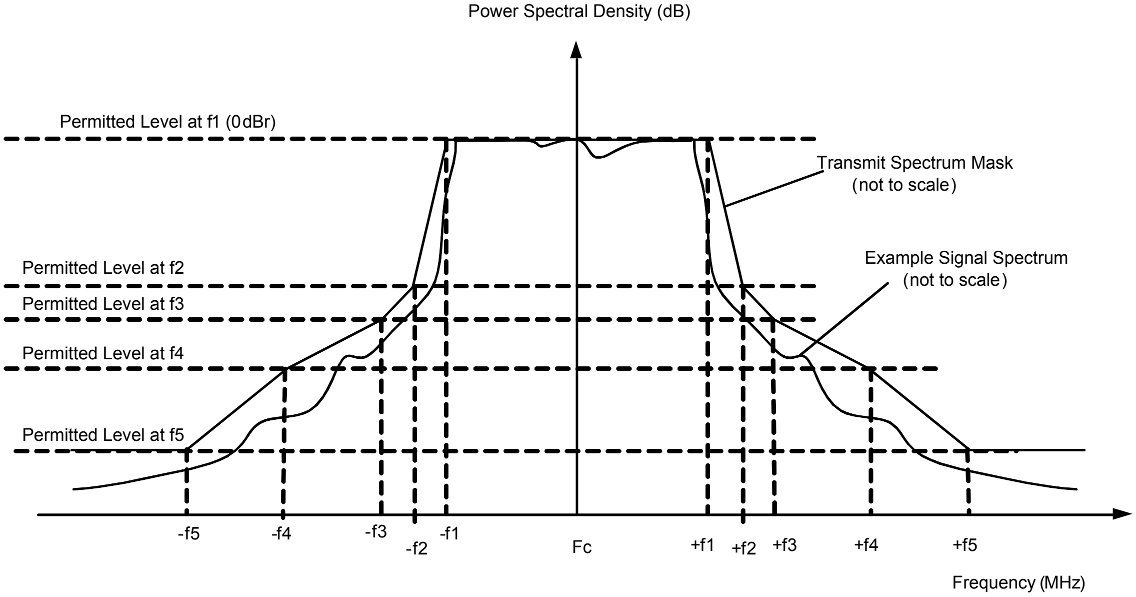


Figure D-1—Transmit spectrum mask and application

D.2.4 Transmit Mask M

This subclause defines the characteristics of transmit mask M.

The power spectral density of the emissions shall be attenuated below the output power of the transmitter as follows:

a) On any frequency removed from the center frequency between 0-45% of the channel bandwidth (BW): 0 dB.

b) On any frequency removed from the center frequency between 45-50% of the channel bandwidth: 568 log (% of (BW)/45) dB.

c) On any frequency removed from the center frequency between 50-55% of the channel bandwidth: 26 + 145 log (% of BW/50) dB.

d) On any frequency removed from the center frequency between 55-100% of the channel bandwidth: 32 + 31 log (% of (BW)/55) dB.

e) On any frequency removed from the center frequency between 100-150% of the channel bandwidth: 40 + 57 log (% of (BW)/100) dB.

f) On any frequency removed from the center frequency between above 150% of the channel bandwidth: 50 dB or 55 + 10 log (P) dB, whichever is the lesser attenuation.

g) The 0 dB reference is measured relative to the highest average power of the fundamental emission measured across the designated channel bandwidth using a resolution bandwidth of 100 kHz and a video bandwidth of 30 kHz. The power spectral density is the power measured within the resolution bandwidth of the measurement device divided by the resolution bandwidth of the measurement device. Emission levels are also based on the use of measurement instrumentation employing a resolution bandwidth of at least one percent of the occupied bandwidth.

D.2.5 CCA-ED threshold

For OFDM PHY operation with CCA-ED, the thresholds shall be less than or equal to –72 dBm for 20 MHz channel widths, –75 dBm for 10 MHz channel widths, and –78 dBm for 5 MHz channel widths (minimum sensitivity for BPSK, R=1/2 + 10 dB in Table 18-14).

Annex E

(normative)

Country elements and operating classes

E.1 Country information and operating classes

WLANs implemented in accordance with this standard and the specifications and definitions referenced in it may be subject to equipment certification and operating requirements established by regional and national regulatory administrations. The specification establishes minimum technical requirements for interoperability, taking into consideration established regulations at the time this standard was issued. These regional and national regulations may be revised or may be superseded. Regulatory requirements that do not affect interoperability are not addressed in this standard. Implementers are referred to the regulatory sources in Table D-1 and their successors for further information. Operation in countries within defined regulatory domains may be subject to additional or alternative national regulations.

The Country element (see 8.4.2.10) allows a STA to configure its PHY and MAC for operation when the operating triplet of Operating Extension Identifier, Operating Class, and Coverage Class fields is present. The operating triplet indicates both PHY and MAC configuration characteristics and operational characteristics. The First Channel Number field of subsequent subband triplet(s) is based on the dot11ChannelStartingFactor that is indicated by the Operating Class field.

The operating class is an index into a set of values for radio operation in a regulatory domain. The operating class tables also contain pointers to behaviors and signal detection limits in Annex D where further operational requirements may be found.

The channel starting frequency variable is a frequency, used together with a channel number, to calculate a channel center frequency.

Channel spacing is the frequency difference between nonoverlapping adjacent channel center frequencies when using the maximum bandwidth allowed for this operating class.

The channel set is the list of integer channel numbers that are legal for a regulatory domain and class.

A behavior limits set is an enumerated list, each element of which points to a row in Table D-2 containing behavior limits in various regulatory domains.

Operating classes for operation in the United States are enumerated in Table E-1

Table E-1—Operating classes in the United States

Operating class	Global operating class (see Table E-4)	Channel starting frequency (GHz)	Channel spacing (MHz)	Channel set	Behavior limits set
1	115	5	20	36, 40, 44, 48	
2	118	5	20	52, 56, 60, 64	DFS_50_100_Behavior
3	124	5	20	149, 153, 157, 161	NomadicBehavior
4	121	5	20	100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140	DFS_50_100_Behavior
5	125	5	20	149, 153, 157, 161, 165	LicenseExemptBehavior
6	103	4.9375	5	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	
7	103	4.9375	5	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	
8	102	4.89	10	11, 13, 15, 17, 19	
9	102	4.89	10	11, 13, 15, 17, 19	
10	101	4.85	20	21, 25	
11	101	4.85	20	21, 25	
12	81	2.407	25	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11	LicenseExemptBehavior
13	94	3.000	20	133, 137	CCA-EDBehavior
14	95	3.000	10	132, 134, 136, 138	CCA-EDBehavior
15	96	3.0025	5	131, 132, 133, 134, 135, 136, 137, 138	CCA-EDBehavior
16[a]		5.0025	5	170–184	ITS_nonmobile_operations, ITS_mobile_operations
17[a, b]		5	10	171–184	ITS_nonmobile_operations, ITS_mobile_operations
18[a, b]		5	20	172–183	ITS_nonmobile_operations, ITS_mobile_operations
19–21	Reserved	Reserved	Reserved	Reserved	Reserved
22	116	5	40	36, 44	PrimaryChannelLowerBehavior
23	119	5	40	52, 60	PrimaryChannelLowerBehavior
24	122	5	40	100, 108, 116, 124, 132	PrimaryChannelLowerBehavior, DFS_50_100_Behavior
25	126	5	40	149, 157	PrimaryChannelLowerBehavior
26	126	5	40	149, 157	LicenseExemptBehavior, PrimaryChannelLowerBehavior
27	117	5	40	40, 48	PrimaryChannelUpperBehavior
28	120	5	40	56, 64	PrimaryChannelUpperBehavior
29	123	5	40	104, 112, 120, 128, 136	NomadicBehavior, PrimaryChannelUpperBehavior, DFS_50_100_Behavior

Table E-1—Operating classes in the United States *(continued)*

Operating class	Global operating class (see Table E-4)	Channel starting frequency (GHz)	Channel spacing (MHz)	Channel set	Behavior limits set
30	127	5	40	153, 161	NomadicBehavior, PrimaryChannelUpperBehavior
31	127	5	40	153, 161	LicenseExemptBehavior, PrimaryChannelUpperBehavior
32	83	2.407	40	1–7	LicenseExemptBehavior, PrimaryChannelLowerBehavior
33	84	2.407	40	5–11	LicenseExemptBehavior, PrimaryChannelUpperBehavior
34–255	Reserved	Reserved	Reserved	Reserved	Reserved
NOTE—The channel spacing for operating classes 22 to 33 is for the supported bandwidth rather than the operating bandwidth. In these operating classes, the AP operates either a 20/40 MHz BSS or a 20 MHz BSS, and the operating bandwidth for a non-AP STA is either 20 MHz or 40 MHz.					

[a]This operating class specifies a list of channels in the 5.9 GHz band. Current regulations may only permit a subset of these channels.

[b]It is the responsibility of management layers outside the scope of this standard to ensure that channels in use at any location are nonoverlapping.

Operating classes for operation in Europe are enumerated in Table E-2.

Table E-2—Operating classes in Europe

Operating class	Global operating class (see Table E-4)	Channel starting frequency (GHz)	Channel spacing (MHz)	Channel set	Behavior limits set
1	115	5	20	36, 40, 44, 48	
2	118	5	20	52, 56, 60, 64	NomadicBehavior
3	121	5	20	100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140	
4	81	2.407	25	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13	LicenseExemptBehavior
5	116	5	40	36, 44	PrimaryChannelLowerBehavior
6	119	5	40	52, 60	PrimaryChannelLowerBehavior
7	122	5	40	100, 108, 116, 124, 132	PrimaryChannelLowerBehavior

Table E-2—Operating classes in Europe *(continued)*

Operating class	Global operating class (see Table E-4)	Channel starting frequency (GHz)	Channel spacing (MHz)	Channel set	Behavior limits set
8	117	5	40	40, 48	PrimaryChannelUpperBehavior
9	120	5	40	56, 64	PrimaryChannelUpperBehavior
10	123	5	40	104, 112, 120, 128, 136	PrimaryChannelUpperBehavior
11	83	2.407	40	1–9	LicenseExemptBehavior, PrimaryChannelLowerBehavior
12	84	2.407	40	5–13	LicenseExemptBehavior, PrimaryChannelUpperBehavior
13[a]		5.0025	5	171–184	ITS_nonmobile_operations, ITS_mobile_operations
14[a, b]		5	10	171–184	ITS_nonmobile_operations, ITS_mobile_operations
15[a, b]		5	20	172–183	ITS_nonmobile_operations, ITS_mobile_operations
16		5	20	100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140	ITS_nonmobile_operations, ITS_mobile_operations
17	125	5	20	149, 153, 157, 161, 165, 169	
18–255	Reserved	Reserved	Reserved	Reserved	Reserved
NOTE—The channel spacing for operating classes 5 to 12 is for the supported bandwidth rather than the operating bandwidth. In these operating classes, the AP operates in a 20/40 MHz BSS, and the operating bandwidth for a non-AP STA is either 20 MHz or 40 MHz.					

[a] This operating class specifies a list of channels in the 5.9 GHz band. Current regulations may only permit a subset of these channels.

[b] It is the responsibility of management layers outside the scope of this standard to ensure that channels in use at any location are nonoverlapping.

Operating classes for operation in Japan are enumerated in Table E-3.

Table E-3—Operating classes in Japan

Operating class	Global operating class (see Table E-4)	Channel starting frequency (GHz)	Channel spacing (MHz)	Channel set	Behavior limits set
1	115	5	20	34, 38, 42, 46[a] 36, 40, 44, 48	
2	112	5	20	8, 12, 16	
3	112	5	20	8, 12, 16	
4	112	5	20	8, 12, 16	
5	112	5	20	8, 12, 16	
6	112	5	20	8, 12, 16	
7	109	4	20	184, 188, 192, 196	
8	109	4	20	184, 188, 192, 196	
9	109	4	20	184, 188, 192, 196	
10	109	4	20	184, 188, 192, 196	
11	109	4	20	184, 188, 192, 196	
12	113	5	10	7, 8, 9, 11	
13	113	5	10	7, 8, 9, 11	
14	113	5	10	7, 8, 9, 11	
15	113	5	10	7, 8, 9, 11	
16	110	4	10	183, 184, 185, 187, 188, 189	
17	110	4	10	183, 184, 185, 187, 188, 189	
18	110	4	10	183, 184, 185, 187, 188, 189	
19	110	4	10	183, 184, 185, 187, 188, 189	
20	110	4	10	183, 184, 185, 187, 188, 189	
21	114	5.0025	5	6, 7, 8, 9, 10, 11	
22	114	5.0025	5	6, 7, 8, 9, 10, 11	
23	114	5.0025	5	6, 7, 8, 9, 10, 11	
24	114	5.0025	5	6, 7, 8, 9, 10, 11	
25	111	4.0025	5	182, 183, 184, 185, 186, 187, 188, 189	

Table E-3—Operating classes in Japan *(continued)*

Operating class	Global operating class (see Table E-4)	Channel starting frequency (GHz)	Channel spacing (MHz)	Channel set	Behavior limits set
26	111	4.0025	5	182, 183, 184, 185, 186, 187, 188, 189	
27	111	4.0025	5	182, 183, 184, 185, 186, 187, 188, 189	
28	111	4.0025	5	182, 183, 184, 185, 186, 187, 188, 189	
29	111	4.0025	5	182, 183, 184, 185, 186, 187, 188, 189	
30	81	2.407	25	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13	LicenseExemptBehavior
31	82	2.414	25	14	LicenseExemptBehavior
32	118	5	20	52, 56, 60, 64	
33	118	5	20	52, 56, 60, 64	
34	121	5	20	100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140	DFS_50_100_Behavior
35	121	5	20	100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140	DFS_50_100_Behavior
36	116	5	40	36, 44	PrimaryChannelLowerBehavior
37	119	5	40	52, 60	PrimaryChannelLowerBehavior
38	119	5	40	52, 60	PrimaryChannelLowerBehavior
39	122	5	40	100, 108, 116, 124, 132	PrimaryChannelLowerBehavior, DFS_50_100_Behavior
40	122	5	40	100, 108, 116, 124, 132	PrimaryChannelLowerBehavior, DFS_50_100_Behavior
41	117	5	40	40, 48	PrimaryChannelUpperBehavior
42	120	5	40	56, 64	PrimaryChannelUpperBehavior
43	120	5	40	56, 64	PrimaryChannelUpperBehavior
44	123	5	40	104, 112, 120, 128, 136	PrimaryChannelUpperBehavior, DFS_50_100_Behavior
45	123	5	40	104, 112, 120, 128, 136	PrimaryChannelUpperBehavior, DFS_50_100_Behavior
46	104	4	40	184, 192	PrimaryChannelLowerBehavior
47	104	4	40	184, 192	PrimaryChannelLowerBehavior

Table E-3—Operating classes in Japan *(continued)*

Operating class	Global operating class (see Table E-4)	Channel starting frequency (GHz)	Channel spacing (MHz)	Channel set	Behavior limits set
48	104	4	40	184, 192	PrimaryChannelLowerBehavior
49	104	4	40	184, 192	PrimaryChannelLowerBehavior
50	104	4	40	184, 192	PrimaryChannelLowerBehavior
51	105	4	40	188, 196	PrimaryChannelUpperBehavior
52	105	4	40	188, 196	PrimaryChannelUpperBehavior
53	105	4	40	188, 196	PrimaryChannelUpperBehavior
54	105	4	40	188, 196	PrimaryChannelUpperBehavior
55	105	4	40	188, 196	PrimaryChannelUpperBehavior
56	83	2.407	40	1–9	LicenseExemptBehavior, PrimaryChannelLowerBehavior
57	84	2.407	40	5–13	LicenseExemptBehavior, PrimaryChannelUpperBehavior
58	121	5	20	100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140	NomadicBehavior, LicenseExemptBehavior
59–255	Reserved	Reserved	Reserved	Reserved	Reserved

NOTE—The channel spacing for operating classes 34–55 is for the supported bandwidth rather than the operating bandwidth. In these regulatory domains, the AP operates in a 20/40 MHz BSS, and the operating bandwidth of a non-AP STA is either 20 MHz or 40 MHz.

[a] The channels 34, 38, 42, and 46 cannot be used after 2012.

Operating classes for operation anywhere in the world are enumerated in Table E-4, and are used in addition to the operating classes enumerated in Table E-1, Table E-2, and Table E-3 (see 8.4.2.56). Where a BSS includes STAs that do not support global operating classes, then all requests and Action frames to those STAs that convey elements containing operating classes shall use nonglobal operating class values.

Table E-4—Global operating classes

Operating class	Nonglobal operating class(es)	Channel starting frequency (GHz)	Channel spacing (MHz)	Channel set	Behavior limits set
1–80		Reserved	Reserved	Reserved	Reserved
81	E-1-12, E-2-4, E-3-30	2.407	25	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13	
82	E-3-31	2.414	25	14	

Table E-4—Global operating classes *(continued)*

Operating class	Nonglobal operating class(es)	Channel starting frequency (GHz)	Channel spacing (MHz)	Channel set	Behavior limits set
83	E-1-32, E-2-11, E-3-56	2.407	40	1, 2, 3, 4, 5, 6, 7, 8, 9	PrimaryChannelLowerBehavior
84	E-1-33, E-2-12, E-3-57	2.407	40	5, 6, 7, 8, 9, 10, 11, 12, 13	PrimaryChannelUpperBehavior
85-93		Reserved	Reserved	Reserved	Reserved
94	E-1-13	3	20	133, 137	CCA-EDBehavior
95	E-1-14	3	10	132, 134, 136, 138	CCA-EDBehavior
96	E-1-15	3.0025	5	131, 132, 133, 134, 135, 136, 137, 138	CCA-EDBehavior
97–100		Reserved	Reserved	Reserved	Reserved
101	E-1-10,11	4.85	20	21, 25	
102	E-1-8,9	4.89	10	11, 13, 15, 17, 19	
103	E-1-6,7	4.9375	5	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	
104	E-3-46,47,48,49,50	4	40	184, 192	PrimaryChannelLowerBehavior
105	E-3-51,52,53,54,55	4	40	188, 196	PrimaryChannelUpperBehavior
106	-	4	20	191, 195	
107	-	4	10	189, 191, 193, 195, 197	
108	-	4.0025	5	188, 189, 190, 191, 192, 193, 194, 195, 196, 197	
109	E-3-7,8,9,10,11	4	20	184, 188, 192, 196	
110	E-3-16,17,18,19,20	4	10	183, 184, 185, 186, 187, 188, 189	

Table E-4—Global operating classes *(continued)*

Operating class	Nonglobal operating class(es)	Channel starting frequency (GHz)	Channel spacing (MHz)	Channel set	Behavior limits set
111	E-3-25,26,27,28,29	4.0025	5	182, 183, 184, 185, 186, 187, 188, 189	
112	E-3-2,3,4,5,6	5	20	8, 12, 16	
113	E-3-12,13,14,15	5	10	7, 8, 9, 10, 11	
114	E-3-21,22,23,24	5.0025	5	6, 7, 8, 9, 10, 11	
115	E-1-1, E-2-1, E-3-1	5	20	36, 40, 44, 48	
116	E-1-22, E-2-5, E-3-36	5	40	36, 44	PrimaryChannelLowerBehavior
117	E-1-27, E-2-8, E-3-41	5	40	40, 48	PrimaryChannelUpperBehavior
118	E-1-2, E-2-2, E-3-32,33	5	20	52, 56, 60, 64	DFS_50_100_Behavior
119	E-1-23, E-2-6, E-3-37,38	5	40	52, 60	PrimaryChannelLowerBehavior, DFS_50_100_Behavior
120	E-1-28, E-2-9, E-3-42,43	5	40	56, 64	PrimaryChannelUpperBehavior, DFS_50_100_Behavior
121	E-1-4, E-2-3, E-3-34,35,58	5	20	100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140	DFS_50_100_Behavior
122	E-1-24, E-2-7, E-3-39,40	5	40	100, 108, 116, 124, 132	PrimaryChannelLowerBehavior, DFS_50_100_Behavior
123	E-1-29, E-2-10, E-3-44,45	5	40	104, 112, 120, 128, 136	PrimaryChannelUpperBehavior, DFS_50_100_Behavior
124	E-1-3	5	20	149, 153, 157, 161	NomadicBehavior
125	E-1-5, E-2-17	5	20	149, 153, 157, 161, 165, 169	LicenseExemptBehavior

Table E-4—Global operating classes *(continued)*

Operating class	Nonglobal operating class(es)	Channel starting frequency (GHz)	Channel spacing (MHz)	Channel set	Behavior limits set
126	E-1-25,26	5	40	149, 157	PrimaryChannelLowerBehavior
127	E-1-30,31	5	40	153, 161	PrimaryChannelUpperBehavior
128–191	-	Reserved	Reserved	Reserved	Reserved
192–254	-	Vendor specific	Vendor specific	Vendor specific	Vendor specific
255	-	Reserved	Reserved	Reserved	Reserved

Nonglobal operating classes refer to the operating classes enumerated in the leftmost column of Table E-1, Table E-2, and Table E-3 (see 8.4.2.56).

NOTE—The following example Country element (see Figure 8-90) describes USA operation (‘55’, ‘53’) using both Table E-1 class 12 (nonglobal) and Table E-4 class 81 (global) for 2.4 GHz band, 11 channels at 100 mW limit (in hexadecimal): ‘07’, ‘0F’, ‘55’, ‘53’, ‘04’, ‘C9’, ‘0C’, ‘0’, ‘01’, ‘0B’, ‘64’, ‘C9’, ‘51’, ‘0’, ‘01’, ‘0B’, ‘64’.

Vendor Specific operating classes are used to carry information not defined in this standard within a single defined format, so that reserved operating classes are not usurped for nonstandard purposes and so that interoperability is more easily achieved in the presence of nonstandard information.

E.2 Band-specific operating requirements

E.2.1 General

Subclause E.2 contains requirements specific to particular bands and regulatory domains.

E.2.2 3650–3700 MHz in the United States

The registration authority is the FCC’s Universal Licensing System (ULS).

Regulations specify the following:

- Certified mobile and portable STAs do not need to be registered, but they “must” operate under the control of an enabling STA (using DSE procedures).
- A registered STA “must” be a fixed STA.
- A registered STA “must” not operate as an enabling STA until the licensee has registered it as a “base station” in ULS.

Enabling STAs and fixed STAs are registered STAs. Dependent non-AP STAs and dependent APs are dependent STAs.

STAs shall use the following:

- CS/CCA
- TPC

- DFS
- CCA-ED (See D.2.5)

No STA shall use channel switch announcement.

No station may transmit for more than 4 ms without carrier sensing, whether transmitting fragments or frames, unless it is controlled by another STA.

STAs shall have the following elements set to true:

- dot11LCIDSERequired
- dot11SpectrumManagementRequired
- dot11MultiDomainCapabilityActivated
- dot11ExtendedChannelSwitchActivated

STAs shall be capable of receiving all channels associated with operating classes 13–15.

STAs shall be capable of transmitting on all channels associated with operating class 15.

STAs shall set the value of dot11DSETransmitDivisor to 256 and the other dot11DSE timer values as shown in Table E-5.

Table E-5—DSE timer limits

Parameter	Seconds
dot11DSEEnablementTimeLimit	32
dot11DSEEnablementFailHoldTime	512
dot11DSERenewalTime	60

E.2.3 5.9 GHz band in the United States (5.850–5.925 GHz)

STAs operating under the behavior limits set 17 in Table D-2 are required to be registered with the FCC ULS. The registration includes the following:

- Classification by coverage size, which is defined by EIRP, and
- Identification of channels the STA is permitted to use.

STAs shall be classified for operation in this band by their maximum transmit power capability, as listed in Table D-3 in D.2.2. STAs shall be compliant with the spectral emission requirements for their class listed in D.2.3.

STAs shall have dot11OCBActivated set to true.

E.2.4 5.9 GHz band in Europe (5.855–5.925 GHz)

STAs shall have dot11OCBActivated set to true.

Annex F

(normative)

HT LDPC matrix definitions

Table F-1 defines the matrix prototypes of the parity-check matrices for a codeword block length *n*=648 bits, with a subblock size *Z*=27 bits.

Table F-1—Matrix prototypes for codeword block length *n*=648 bits, subblock size is *Z* = 27 bits

(a) Coding rate R = 1/2.

0	-	-	-	0	0	-	-	0	-	-	0	1	0	-	-	-	-	-	-	-	-	-	-
22	0	-	-	17	-	0	0	12	-	-	-	-	0	0	-	-	-	-	-	-	-	-	-
6	-	0	-	10	-	-	-	24	-	0	-	-	-	0	0	-	-	-	-	-	-	-	-
2	-	-	0	20	-	-	-	25	0	-	-	-	-	-	0	0	-	-	-	-	-	-	-
23	-	-	-	3	-	-	-	0	-	9	11	-	-	-	-	0	0	-	-	-	-	-	-
24	-	23	1	17	-	3	-	10	-	-	-	-	-	-	-	-	0	0	-	-	-	-	-
25	-	-	-	8	-	-	-	7	18	-	-	0	-	-	-	-	-	0	0	-	-	-	-
13	24	-	-	0	-	8	-	6	-	-	-	-	-	-	-	-	-	-	0	0	-	-	-
7	20	-	16	22	10	-	-	23	-	-	-	-	-	-	-	-	-	-	-	0	0	-	-
11	-	-	-	19	-	-	-	13	-	3	17	-	-	-	-	-	-	-	-	-	0	0	-
25	-	8	-	23	18	-	14	9	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0
3	-	-	-	16	-	-	2	25	5	-	-	1	-	-	-	-	-	-	-	-	-	-	0

(b) Coding rate R = 2/3.

25	26	14	-	20	-	2	-	4	-	-	8	-	16	-	18	1	0	-	-	-	-	-	-
10	9	15	11	-	0	-	1	-	-	18	-	8	-	10	-	-	0	0	-	-	-	-	-
16	2	20	26	21	-	6	-	1	26	-	7	-	-	-	-	-	-	0	0	-	-	-	-
10	13	5	0	-	3	-	7	-	-	26	-	-	13	-	16	-	-	-	0	0	-	-	-
23	14	24	-	12	-	19	-	17	-	-	-	20	-	21	-	0	-	-	-	0	0	-	-
6	22	9	20	-	25	-	17	-	8	-	14	-	18	-	-	-	-	-	-	-	0	0	-
14	23	21	11	20	-	24	-	18	-	19	-	-	-	-	22	-	-	-	-	-	-	0	0
17	11	11	20	-	21	-	26	-	3	-	-	18	-	26	-	1	-	-	-	-	-	-	0

(c) Coding rate R = 3/4.

16	17	22	24	9	3	14	-	4	2	7	-	26	-	2	-	21	-	1	0	-	-	-	-
25	12	12	3	3	26	6	21	-	15	22	-	15	-	4	-	-	16	-	0	0	-	-	-
25	18	26	16	22	23	9	-	0	-	4	-	4	-	8	23	11	-	-	-	0	0	-	-
9	7	0	1	17	-	-	7	3	-	3	23	-	16	-	-	21	-	0	-	-	0	0	-
24	5	26	7	1	-	-	15	24	15	-	8	-	13	-	13	-	11	-	-	-	-	0	0
2	2	19	14	24	1	15	19	-	21	-	2	-	24	-	3	-	2	1	-	-	-	-	0

(d) Coding rate R = 5/6.

17	13	8	21	9	3	18	12	10	0	4	15	19	2	5	10	26	19	13	13	1	0	-	-
3	12	11	14	11	25	5	18	0	9	2	26	26	10	24	7	14	20	4	2	-	0	0	-
22	16	4	3	10	21	12	5	21	14	19	5	-	8	5	18	11	5	5	15	0	-	0	0
7	7	14	14	4	16	16	24	24	10	1	7	15	6	10	26	8	18	21	14	1	-	-	0

Table F-2 defines the matrix prototypes of the parity-check matrices for a codeword block length *n*=1296 bits, with a subblock size *Z*=54 bits.

Table F-2—Matrix prototypes for codeword block length *n*=1296 bits, subblock size is *Z*= 54 bits

(a) Coding rate R = 1/2.

40	-	-	-	22	-	49	23	43	-	-	-	1	0	-	-	-	-	-	-	-	-	-	-
50	1	-	-	48	35	-	-	13	-	30	-	-	0	0	-	-	-	-	-	-	-	-	-
39	50	-	-	4	-	2	-	-	-	-	49	-	-	0	0	-	-	-	-	-	-	-	-
33	-	-	38	37	-	-	4	1	-	-	-	-	-	-	0	0	-	-	-	-	-	-	-
45	-	-	-	0	22	-	-	20	42	-	-	-	-	-	-	0	0	-	-	-	-	-	-
51	-	-	48	35	-	-	-	44	-	18	-	-	-	-	-	-	0	0	-	-	-	-	-
47	11	-	-	-	17	-	-	51	-	-	-	0	-	-	-	-	-	0	0	-	-	-	-
5	-	25	-	6	-	45	-	13	40	-	-	-	-	-	-	-	-	-	0	0	-	-	-
33	-	-	34	24	-	-	-	23	-	-	46	-	-	-	-	-	-	-	-	0	0	-	-
1	-	27	-	1	-	-	-	38	-	44	-	-	-	-	-	-	-	-	-	-	0	0	-
-	18	-	-	23	-	-	8	0	35	-	-	-	-	-	-	-	-	-	-	-	-	0	0
49	-	17	-	30	-	-	-	34	-	-	19	1	-	-	-	-	-	-	-	-	-	-	0

(b) Coding rate R = 2/3.

39	31	22	43	-	40	4	-	11	-	-	50	-	-	-	6	1	0	-	-	-	-	-	-
25	52	41	2	6	-	14	-	34	-	-	-	24	-	37	-	-	0	0	-	-	-	-	-
43	31	29	0	21	-	28	-	-	2	-	-	7	-	17	-	-	-	0	0	-	-	-	-
20	33	48	-	4	13	-	26	-	-	22	-	-	46	42	-	-	-	-	0	0	-	-	-
45	7	18	51	12	25	-	-	-	50	-	-	5	-	-	-	0	-	-	-	0	0	-	-
35	40	32	16	5	-	-	18	-	-	43	51	-	32	-	-	-	-	-	-	-	0	0	-
9	24	13	22	28	-	-	37	-	-	25	-	-	52	-	13	-	-	-	-	-	-	0	0
32	22	4	21	16	-	-	-	27	28	-	38	-	-	-	8	1	-	-	-	-	-	-	0

(c) Coding rate R = 3/4.

39	40	51	41	3	29	8	36	-	14	-	6	-	33	-	11	-	4	1	0	-	-	-	-
48	21	47	9	48	35	51	-	38	-	28	-	34	-	50	-	50	-	-	0	0	-	-	-
30	39	28	42	50	39	5	17	-	6	-	18	-	20	-	15	-	40	-	-	0	0	-	-
29	0	1	43	36	30	47	-	49	-	47	-	3	-	35	-	34	-	0	-	-	0	0	-
1	32	11	23	10	44	12	7	-	48	-	4	-	9	-	17	-	16	-	-	-	-	0	0
13	7	15	47	23	16	47	-	43	-	29	-	52	-	2	-	53	-	1	-	-	-	-	0

(d) Coding rate R = 5/6.

48	29	37	52	2	16	6	14	53	31	34	5	18	42	53	31	45	-	46	52	1	0	-	-
17	4	30	7	43	11	24	6	14	21	6	39	17	40	47	7	15	41	19	-	-	0	0	-
7	2	51	31	46	23	16	11	53	40	10	7	46	53	33	35	-	25	35	38	0	-	0	0
19	48	41	1	10	7	36	47	5	29	52	52	31	10	26	6	3	2	-	51	1	-	-	0

Table F-3 defines the matrix prototypes of the parity-check matrices for a codeword block length *n*=1944 bits, with a subblock size *Z*=81 bits.

Table F-3—Matrix prototypes for codeword block length *n*=1944 bits, subblock size is *Z* = 81 bits

(a) Coding rate R = 1/2.

57	-	-	-	50	-	11	-	50	-	79	-	1	0	-	-	-	-	-	-	-	-	-	-
3	-	28	-	0	-	-	-	55	7	-	-	-	0	0	-	-	-	-	-	-	-	-	-
30	-	-	-	24	37	-	-	56	14	-	-	-	-	0	0	-	-	-	-	-	-	-	-
62	53	-	-	53	-	-	3	35	-	-	-	-	-	-	0	0	-	-	-	-	-	-	-
40	-	-	20	66	-	-	22	28	-	-	-	-	-	-	-	0	0	-	-	-	-	-	-
0	-	-	-	8	-	42	-	50	-	-	8	-	-	-	-	-	0	0	-	-	-	-	-
69	79	79	-	-	-	56	-	52	-	-	-	0	-	-	-	-	-	0	0	-	-	-	-
65	-	-	-	38	57	-	-	72	-	27	-	-	-	-	-	-	-	-	0	0	-	-	-
64	-	-	-	14	52	-	-	30	-	-	32	-	-	-	-	-	-	-	-	0	0	-	-
-	45	-	70	0	-	-	-	77	9	-	-	-	-	-	-	-	-	-	-	-	0	0	-
2	56	-	57	35	-	-	-	-	-	12	-	-	-	-	-	-	-	-	-	-	-	0	0
24	-	61	-	60	-	-	27	51	-	-	16	1	-	-	-	-	-	-	-	-	-	-	0

(b) Coding rate R = 2/3.

61	75	4	63	56	-	-	-	-	-	-	8	-	2	17	25	1	0	-	-	-	-	-	-
56	74	77	20	-	-	-	64	24	4	67	-	7	-	-	-	-	0	0	-	-	-	-	-
28	21	68	10	7	14	65	-	-	-	23	-	-	-	75	-	-	-	0	0	-	-	-	-
48	38	43	78	76	-	-	-	-	5	36	-	15	72	-	-	-	-	-	0	0	-	-	-
40	2	53	25	-	52	62	-	20	-	-	44	-	-	-	-	0	-	-	-	0	0	-	-
69	23	64	10	22	-	21	-	-	-	-	-	68	23	29	-	-	-	-	-	-	0	0	-
12	0	68	20	55	61	-	40	-	-	-	52	-	-	-	44	-	-	-	-	-	-	0	0
58	8	34	64	78	-	-	11	78	24	-	-	-	-	-	58	1	-	-	-	-	-	-	0

(c) Coding rate R = 3/4.

48	29	28	39	9	61	-	-	-	63	45	80	-	-	-	37	32	22	1	0	-	-	-	-
4	49	42	48	11	30	-	-	-	49	17	41	37	15	-	54	-	-	-	0	0	-	-	-
35	76	78	51	37	35	21	-	17	64	-	-	-	59	7	-	-	32	-	-	0	0	-	-
9	65	44	9	54	56	73	34	42	-	-	-	35	-	-	-	46	39	0	-	-	0	0	-
3	62	7	80	68	26	-	80	55	-	36	-	26	-	9	-	72	-	-	-	-	-	0	0
26	75	33	21	69	59	3	38	-	-	-	35	-	62	36	26	-	-	1	-	-	-	-	0

(d) Coding rate R = 5/6.

13	48	80	66	4	74	7	30	76	52	37	60	-	49	73	31	74	73	23	-	1	0	-	-
69	63	74	56	64	77	57	65	6	16	51	-	64	-	68	9	48	62	54	27	-	0	0	-
51	15	0	80	24	25	42	54	44	71	71	9	67	35	-	58	-	29	-	53	0	-	0	0
16	29	36	41	44	56	59	37	50	24	-	65	4	65	52	-	4	-	73	52	1	-	-	0

Annex G

(normative)

Frame exchange sequences

G.1 General

The allowable frame exchange sequences are defined using an extension of the EBNF format as defined in ISO/IEC 14977 : 1996 [B46]. The elements of this syntax that are used here are as follows:

- [a] = a is optional.
- {a} = a is repeated zero or more times.
- n{a} = a is repeated n or more times. For example, 3{a} requires 3 or more "a". This notation is an extension to ISO/IEC 14977 and equivalent to n*a{a} as defined in that standard.
- a|b|c|... = selection between mutually exclusive alternatives, a, b, c ....
- ( ) = grouping, e.g., "a (b|c)" is equivalent to "a b | a c".
- (* a *) = “a” is a comment. Comments are placed before the text they relate to.
- < > = order of frames not relevant. For example, <a b> is either "a b" or "b a."
- A rule is terminated by a semicolon ";"
- The meaning of whitespace is changed from ISO/IEC 14977. Terminals do not contain whitespace, and the concatenate-symbol (comma in ISO/IEC 14977) is replaced by white space. Whitespace appearing between terminals indicates concatenation. Otherwise, whitespace is not significant and is used to highlight the nesting of grouped terms.

Two types of terminals are defined:

- **Frames.** A frame is shown in **bold** and identified by its type/subtype (e.g., **Beacon**, **Data**). Frames are shown with an initial capital letter.
- *Attributes.* Attributes are shown in *italic*. An attribute is introduced by the "+" character. The attribute specifies a condition that applies to the frame that precedes it. Where there are multiple attributes applied, they are generally ordered in the same order of the fields in the frame to which they refer. The syntax a+(b|c) where b and c are attributes is equivalent to (a+b) | (a+c).

Nonterminals of this syntax are shown in a normal font, i.e., a sequence of words joined by hyphens (e.g., cf-frame-exchange-sequence).

The attributes are defined in Table G-1.

Table G-1—Attributes applicable to frame exchange sequence definition

Attribute	Description
a-mpdu	Frame is part of an A-MPDU aggregate.
a-mpdu-end	Frame is the last frame in an A-MPDU aggregate.
block-ack	QoS data frame has ack policy equal to Block Ack.
broadcast	Frame RA is the broadcast address.
CF	Beacon contains a CFP element.

Table G-1—Attributes applicable to frame exchange sequence definition *(continued)*

Attribute	Description
CF-Ack	Data type CF-Ack subtype bit equal to 1 or CF-End+CF-Ack frame.
CF-Poll	Data type CF-Poll subtype bit equal to 1.
csi	An Action frame carrying channel state feedback (i.e., CSI, uncompressed beamforming, or compressed beamforming feedback matrices).
csi-request	A +HTC frame with the Feedback Request field equal to a value > 0.
delayed	BlockAck or BlockAckReq under a delayed policy.
delayed-no-ack	BlockAck or BlockAckReq frame has No Ack policy.
DTIM	Beacon is a DTIM.
frag	Frame has its More Fragments field equal to 1.
group	Frame RA has i/g bit equal to 1.
HTC	+HTC frame, i.e., a frame that contains the HT Control field, including the Control Wrapper frame. See NOTE.
implicit-bar	QoS data frame in an A-MPDU with Normal Ack policy.
individual	Frame RA has i/g bit equal to 0.
last	Frame has its More Fragments field equal to 0.
L-sig	L-sig duration not equal to PPDU duration.
action-no-ack	Management frame of subtype Action No Ack.
mfb	A +HTC frame with the MFB field is not equal to all ones.
more-psmp	A PSMP frame with the More PSMP field equal to 1.
mrq	A +HTC frame with the MRQ subfield equal to 1.
ndp-announce	A +HTC frame with the NDP Announcement subfield equal to 1.
no-ack	QoS Data frame has ack policy equal to No Ack.
no-more-psmp	A PSMP frame with the More PSMP field equal to 0.
normal-ack	QoS Data frame has ack policy equal to Normal Ack.
non-QAP	Frame is transmitted by a non-AP QoS STA.
non-stbc	PPDU TXVECTOR STBC parameter is equal to 0.
null	Data type Null Data subtype bit equal to 1.
pifs	Frame is transmitted following a PIFS.
psmp-ack	Ack Policy field of QoS data frame is equal to PSMP Ack.
QAP	Frame is transmitted by a QoS AP.
QoS	Data type QoS subtype bit equal to 1.
RD	Frame includes an HT Control field in which the RDG/More PPDU subfield is equal to 1.
self	Frame RA = TA.
sounding	PPDU TXVECTOR SOUNDING parameter is present and equal to SOUNDING.
stbc	PPDU TXVECTOR STBC parameter is equal to a value >0.

Table G-1—Attributes applicable to frame exchange sequence definition *(continued)*

Attribute	Description
to-ap	Frame is addressed to the AP.
trq	Frame is a +HTC frame with the TRQ field equal to 1.
NOTE—A control frame that contains the HT Control field is always transmitted using the control wrapper frame.	

G.2 Basic sequences

The allowable frame exchange sequence is defined by the rule frame sequence. Except where modified by the *pifs* attribute, frames are separated by a SIFS.

(* This rule defines all the allowable frame exchange sequences *)
frame-sequence =
([**CTS**] (**Management** +*broadcast* | **Data** +*group*)) |
([**CTS** | **RTS CTS** | **PS-Poll**] {frag-frame **ACK**} last-frame **ACK**) |
(**PS-Poll ACK**) |
([**Beacon** +*DTIM*] {cf-sequence} [**CF-End** [+*CF-Ack*]])|
hcf-sequence |
mcf-sequence;

(* A frag-frame is a nonfinal part of an individually addressed MSDU or MMPDU *)
frag-frame = (**Data** | **Management**) +*individual* +*frag*;

(* This is the last (or only) part of a an individually addressed MSDU or MMPDU *)
last-frame = (**Data** | **Management**) +*individual* +*last*;

(* A cf-sequence expresses all the sequences that may be generated within a contention-free period. The first frame in this sequence is sent by the AP. *)
cf-sequence =
(*Broadcast *)
Beacon | **Management** +*broadcast* | **Data** +*group* [+*QoS*] |

(* CF poll with data *)
(**Data**+*individual* +*CF-Poll* [+*CF-Ack*]
(**Data** +*individual* +*CF-Ack* [**Data** +*null* +*CF-Ack*] |
Data +*null* +*CF-Ack*)) |

(* CF poll without data *)
Data +*individual* +*null* +*CF-Poll* [+*CF-Ack*]
(**Data** +*null* |
(**Data** +*individual* (**Data** +*null* +*CF-Ack* | **ACK**)))|

(* individual management *)
(**Management** +*individual* **ACK**) |

(* All the sequences initiated by an HC *)
hcf-sequence;

G.3 EDCA and HCCA sequences

(* An hcf-sequence represents all the sequences that may be generated under HCCA. The sequence may be initiated by an HC within a CFP, or it may be initiated by a STA using EDCA channel access. *)
hcf-sequence =
([**CTS**] 1{(**Data** +*group* [+*QoS*]) | **Management** +*broadcast*) +pifs} |
([**CTS**] 1{txop-sequence}) |

(* HC only, polled TXOP delivery *)
([**RTS CTS**] non-cf-ack-piggybacked-qos-poll-sequence)

(* HC only, polled TXOP delivery *)
cf-ack-piggybacked-qos-poll-sequence |

(* HC only, self TXOP delivery or termination *)
Data +*self* +*null* +*CF-Poll* +*QoS*;

(* A cf-ack-piggybacked-qos-poll-sequence is the start of a polled TXOP that also delivers a CF-Ack. There are two main variants, polls that deliver data and, therefore, need acknowledgment and polls that do not. *)
cf-ack-piggybacked-qos-poll-sequence=
(qos-poll-requiring-no-ack +*CF-Ack* (
[**CTS** +*self*] polled-txop-content |
polled-txop-termination)) |
(qos-poll-requiring-ack +*CF-Ack* (
ACK (
polled-txop-content |
polled-txop-termination)) |
cf-ack-piggybacked-qos-data-sequence);

(* A non-cf-ack-piggybacked-qos-poll-sequence is the start of a polled TXOP that does not deliver a CF-Ack. Except for this, it is identical to the CF-Ack version. *)
non-cf-ack-piggybacked-qos-poll-sequence=
(qos-poll-requiring-no-ack (
[**CTS** +*self*] polled-txop-content |
polled-txop-termination)) |
(qos-poll-requiring-ack (
ACK (
polled-txop-content |
polled-txop-termination)) |
cf-ack-piggybacked-qos-data-sequence);

(* This sequence is the delivery of a single frame that is the TXOP poll frame that does not require acknowledgment either because the frame carries no data or because the frame carries data that do not

require immediate acknowledgment. *)
qos-poll-requiring-no-ack =

Data +*null* +*CF-Poll* +*QoS* |

Data +*individual* +*CF-Poll* +*QoS* +(*no-ack*|*block-ack*);

(* A qos-poll-requiring-ack is the delivery of a single frame that is a TXOP poll frame, but also carries data that require immediate acknowledgment. *)
qos-poll-requiring-ack =

Data +*individual* +*CF-Poll* [+*CF-Ack*] +*QoS* +*normal-ack*;

(* Polled-txop-content is what may occur after the delivery of a polled TXOP. A QoS STA transmits the first frame in this sequence *)
polled-txop-content =

1{txop-sequence} [polled-txop-termination];

(* A polled-txop-termination may be used by a QoS STA to terminate the polled TXOP. The data frame is addressed to the HC, which regains control of the medium and may reuse any unused polled TXOP duration. *)
polled-txop-termination =

Data +*individual* +*null* +*QoS* +*normal-ack* **ACK**;

(* A TXOP (either polled or EDCA) may be filled with txop-sequences, which are initiated by the TXOP holder. *)
txop-sequence =

(((**RTS CTS**) | **CTS** +*self*) **Data** +*individual* +*QoS* +(*block-ack* | *no-ack*)) |

[**RTS CTS**] (txop-part-requiring-ack txop-part-providing-ack)|

[**RTS CTS**] **(Management** | (**Data** +*QAP*)) +*individual* **ACK** |

[**RTS CTS**] **(BlockAckReq BlockAck**) |

ht-txop-sequence;

(* These frames require acknowledgment *)
txop-part-requiring-ack =

Data +*individual* [+*null*] |

Data +*individual* [+*null*] +*QoS* +*normal-ack* |

BlockAckReq +*delayed* |

BlockAck +*delayed*;

(* These frames provide acknowledgment to the txop-part-requiring-ack *)
txop-part-providing-ack=

ACK |

cf-ack-piggybacked-qos-poll-sequence | (* An HC responds with a new polled TXOP on expiry of current TXOP *)

cf-ack-piggybacked-qos-data-sequence | (* An HC responds with CF-Ack and its own data on expiry of TXOP *)

Data +*CF-Ack*;

(* An HC has received a frame requiring Ack with a duration value indicating the end of the TXOP. The HC continues the CAP by transmitting its own data. *)
cf-ack-piggybacked-qos-data-sequence =

```
        ( Data +individual +CF-Ack +QoS +(no-ack|block-ack) polled-txop-content ) |
        ( Data +individual +CF-Ack +QoS+normal-ack (
            ACK polled-txop-content |
            Data +CF-Ack |
            cf-ack-piggybacked-qos-poll-sequence ) ) ;
```

(* An mcf-sequence represents all the sequences that may be generated under MCF. The sequence may be initiated by a mesh STA using EDCA channel access or MCCA channel access. *)

```
mcf-sequence =
        ( [CTS] |{(Data+group+QoS ) | Management+broadcast} ) | ( [CTS] 1{txop-sequence} ) |
        group-mccaop-abandon;
```

(* A group-mccaop-abandon is the delivery of a single QoS Null frame by a mesh STA that has dot11MCCAActivated true. *)

```
group-mccaop-abandon =
        Data+broadcast+null+QoS
```

G.4 HT sequences

(* The ht-txop-sequence describes the additional sequences that may be initiated by an HT STA that is the holder of a TXOP *)

```
ht-txop-sequence =      L-sig-protected-sequence |
                        ht-nav-protected-sequence |
                        dual-cts-protected-sequence |
                        1{initiator-sequence};
```

(* an L-sig-protected-sequence is a sequence protected using the L-sig TXOP protection feature *)
L-sig-protected-sequence = L-sig-protection-set 1{initiator-sequence} resync-sequence;

(* an ht-nav-protected sequence consists of setting the NAV, performing one or more initiator-sequences and then resetting the NAV if time permits *)
ht-nav-protected-sequence = nav-set 1{initiator-sequence} [resync-sequence] ;

(* a dual-cts-protected-sequence is a sequence protected using the dual CTS protection feature *)
dual-cts-protected-sequence = dual-cts-nav-set 1{initiator-sequence} [dual-cts-nav-reset];

(* a dual-cts-nav-set is an initial exchange that establishes NAV protection using dual CTS protection. *)

```
dual-cts-nav-set =      (* A dual CTS initiated by a non-AP HT STA that is not STBC-capable,
                        preceded by an optional CTS frame addressed to the AP. *)
                        (
                                    [ CTS+to-ap+non-stbc+non-QAP ]
                                    RTS+non-stbc+non-QAP
                                    CTS+non-stbc+QAP
                                    [ CTS+stbc+pifs+QAP ]
                        ) |

                        (* A dual CTS initiated by a non-AP STA that is STBC-capable, preceded by an
                        optional CTS frame addressed to the AP. *)
                        (
                                    [ CTS+to-ap+stbc+non-QAP ]
                                    RTS+stbc+non-QAP
                                    CTS+stbc+QAP
```

CTS+*non-stbc*+*QAP*
) |

(* An STBC initiator-sequence (i.e., containing STBC PPDUs) transmitted by the AP is protected by non-STBC CTS to self *)
(**CTS**+*self*+*non-stbc*+*QAP*) |

(* A non-STBC initiator-sequence transmitted by the AP is protected by STBC CTS to self *)
(**CTS**+*self*+*stbc*+*QAP*);

(* a dual-cts-nav-reset resets the NAV in the vicinity of the transmitting non-AP STA, and resets the NAV of both STBC and non-STBC-capable STA in the vicinity of the AP *)
dual-cts-nav-reset = [**CF-End**+*non-QAP*] **CF-End**+*stbc*+*QAP* **CF-End**+*non-stbc*+*QAP*);

(* an ma-no-ack-htc represents an Action No Ack + HTC frame *)
ma-no-ack-htc = **Management**+*action-no-ack*+*HTC;*

(* This is the sequence of frames that establish protection using the L-sig TXOP protection method *)
L-sig-protection-set = **(RTS**+*L-sig*[+*HTC*] **CTS**+*L-sig*[+*HTC*]) |
(Data+*individual*+*L-sig* [+*HTC*][+*null*][+*QoS*+*normal-ack*] **ACK** [+*HTC*] +*L-sig*) |
(1{ **Data**+*L-sig*[+*HTC*]+*individual*+*QoS*+*implicit-bar*+*a-mpdu*}+*a-mpdu-end*
BlockAck+*L-sig*[+*HTC*]
) |
(**BlockAckReq**+*L-sig*[+*HTC*] (**BlockAck**[+*HTC*]|**ACK**[+*HTC*])+*L-sig*) |
(**BlockAck**+*L-sig*[+*HTC*] **ACK**[+*HTC*])+*L-sig*);

(* These are the series of frames that establish NAV protection for an HT sequence *)
nav-set = (**RTS**[+*HTC*] **CTS**[+*HTC*]) |
CTS+*self* |
(Data[+*HTC*]+*individual*[+*null*][+*QoS*+*normal-ack*] **ACK)** |
Data[+*HTC*]+*individual*[+*QoS*+(*block-ack*)] |
Data+*group*[+*null*][+*QoS*] |
(1{ **Data**[+*HTC*]+*individual*+*QoS*+*implicit-bar*+*a-mpdu*}+*a-mpdu-end*
BlockAck[+*HTC*]
) |
(**BlockAckReq**[+*HTC*] (**BlockAck**[+*HTC*]|**ACK**[+*HTC*])) |
(**BlockAck**[+*HTC*] **ACK**);

resync-sequence = **CF-End | (CF-End+***non-QAP* **CF-End+***QAP***)**;

(* This is an initiator sequence. The different forms arise from whether the initiator transmits a frame that requires a BlockAck, and whether it delivers an RDG. When an RDG is delivered, the response is distinguished according to whether it demands a BlockAck response from the initiator. *)
initiator-sequence = (* No BlockAck expected, no RDG *)
burst |

(* BlockAckReq delivered, BlockAck expected. No RD *)
(burst-bar (**BlockAck**|**ACK**) [+*HTC*]) |

(* No BlockAckReq delivered, RDG *)
(burst-rd (
burst |

```
                                    burst-bar initiator-sequence-ba
                                    )
                        ) |

                        (burst-rd-bar (BlockAck|ACK) [+HTC]) |
                        (burst-rd-bar       (
                                    burst-ba |
                                    burst-ba-bar initiator-sequence-ba
                                    )
                        ) |
                        ht-ack-sequence |
                        psmp-burst |
                        link-adaptation-exchange ;

(* This is the same as the initiator-sequence, except the initiator is constrained to generate a BlockAck
response because a previous RD response contained a BlockAckReq *)
initiator-sequence-ba = burst-ba |
                        (burst-ba-bar (BlockAck|ACK)[+HTC)) |
                        (burst-ba-rd        (
                                    burst |
                                    burst-bar initiator-sequence-ba
                                    )
                        ) |
                        (burst-ba-rd-bar (BlockAck|ACK)[+HTC]) |
                        (burst-ba-rd-bar (
                                    burst-ba |
                                    burst-ba-bar initiator-sequence-ba
                                    )
                        );

(* These are sequences that occur within an ht-txop-sequence that have an ack response *)
ht-ack-sequence =       (BlockAck+delayed[+HTC] ACK[+HTC]) |
                        (BlockAckReq+delayed[+HTC] ACK[+HTC]) |
                        (Data[+HTC]+individual[+null][+QoS+normal-ack] ACK[+HTC]);

(* A burst is a sequence of 1 or more packets, none of them requiring a response *)
burst =                 1{ppdu-not-requiring-response};

(* A burst containing a BlockAckReq *)
burst-bar =             {ppdu-not-requiring-response} ppdu-bar;

(* A burst containing a BlockAck *)
burst-ba =              ppdu-ba {ppdu-not-requiring-response};

(* A burst containing a BlockAck and BlockAckReq, either in the same packet, or in separate packets. *)
burst-ba-bar =          (ppdu-ba {ppdu-not-requiring-response} ppdu-bar) |
                        ppdu-ba-bar;

(* A burst delivering an RDG *)
burst-rd =              {ppdu-not-requiring-response} ppdu-rd;

(* A burst containing a BlockAckReq and delivering an RDG *)
burst-rd-bar = burst ppdu-rd-bar;
```

```
(* A burst containing a BlockAck and delivering an RDG *)
burst-ba-rd =               (ppdu-ba {ppdu-not-requiring-response} ppdu-rd) |
                            ppdu-ba-rd;

(* A burst containing a BlockAckReq and BlockAck and delivering an RDG *)
burst-ba-rd-bar =           (ppdu-ba {ppdu-not-requiring-response} ppdu-rd-bar) |
                            ppdu-ba-rd-bar;

(* A PPDU not requiring a response is either a single frame not requiring response, or an A-MPDU of such
frames.*)
ppdu-not-requiring-response =
                            frame-not-requiring-response-non-ampdu |
                            1{frame-not-requiring-response-ampdu+a-mpdu}+a-mpdu-end;

(* A frame-not-requiring-response-non-ampdu is a frame that does not require a response and that may be
sent outside A-MPDU. It includes frames that do not require a response and that are not allowed within an
A-MPDU. *)
frame-not-requiring-response-non-ampdu =
                            Data[+HTC]+QoS+no-ack |
                            frame-not-requiring-response-ampdu;

(* A frame-not-requiring-response-ampdu is a frame that does not require a response and can be sent within
an A-MPDU. It is one of the delayed Block Ack policy frames sent under No Ack policy, or Data that does
not require an immediate ack, or an Action No Ack frame. A frame-not-requiring-response may be included
with any of the following sequences in any position, except the initial position when this contains a
BlockAck or Multi-TID BlockAck: ppdu-bar, ppdu-ba-bar, ppdu-ba, ppdu-rd, ppdu-rd-bar, ppdu-ba-rd-bar,
psmp-ppdu *)
frame-not-requiring-response-ampdu =
                            BlockAck[+HTC]+delayed-no-ack |
                            BlockAckReq[+HTC]+delayed-no-ack |
                            Data[+HTC]+QoS+block-ack |
                            ma-no-ack-htc;

(* A PPDU containing a BlockAckReq is either a non-A-MPDU BlockAckReq, or an A-MPDU containing
Data carrying implicit Block Ack request*).
ppdu-bar=                   BlockAckReq[+HTC] |
                            (1{Data[+HTC]+QoS+implicit-bar+a-mpdu} + a-mpdu-end);

(* A PPDU containing both BlockAck and BlockAckReq is an A-MPDU that contains a BlockAck, plus
either a BlockAckReq frame, or 1 or more data frames carrying implicit Block Ack request. *)
ppdu-ba-bar=                BlockAck[+HTC]+a-mpdu
                            (
                                          BlockAckReq[+HTC]+a-mpdu |
                                          1{Data[+HTC]+QoS+implicit-bar+a-mpdu}
                            ) + a-mpdu-end;

(*A PPDU containing BlockAck is either a non-A-MPDU BlockAck, or an A-MPDU containing a
BlockAck, and also containing data that does not carry implicit Block Ack request. *)
ppdu-ba=                    BlockAck[+HTC] |
                            (
                                          BlockAck[+HTC]+a-mpdu
                                          1{Data[+HTC]+QoS+(no-ack|block-ack)+a-mpdu}
                            ) + a-mpdu-end;
```

(* A PPDU delivering an RDG, but not delivering a BlockAckReq is either a data frame, not requiring immediate acknowledgment, or a BlockAck or BlockAckReq, not requiring immediate acknowledgment *).
ppdu-rd= **Data**+*HTC*[+*null*]+*QoS*+(*no-ack*|*block-ack*)+*RD* |
(BlockAck|BlockAckReq)+*HTC*+*delayed-no-ack*+*RD* |
(
1{**Data**+*HTC*+*QoS*+*RD*+*a-mpdu*}
) + *a-mpdu-end*;

(* A PPDU containing a BlockAckReq and delivering an RDG is either an non-A-MPDU BlockAckReq frame, or an A-MPDU containing at least one data frame with RD and implicit-bar. *)
ppdu-rd-bar= **BlockAckReq**+*HTC*+*RD* |
(
1{**Data**+*HTC*+*QoS*+*implicit-bar*+*RD*+*a-mpdu*}
) + a-mpdu-end;

(* A PPDU containing a BlockAck and granting RD is either an unaggregated BlockAck or an A-MPDU that contains a BlockAck and at least one data frame containing RD, but not implicit Block Ack request. *)
ppdu-ba-rd= **BlockAck**+*HTC*+*RD* |
(
BlockAck+*a-mpdu* (
1{**Data**+*HTC*+*QoS*+(*no-ack*|*block-ack*)+*RD*+*a-mpdu*}
)
) + *a-mpdu-end;*

(* A PPDU containing a BlockAck, BlockAckReq and granting RD is an A-MPDU that contains a BlockAck and either an explicit BlockAckReq (and no data frames) or data frames carrying the implicit Block Ack request. The RD attribute is present in all frames carrying an HT Control field, and at least one of these frames is present. This constraint is not expressed in the syntax below. *)
ppdu-ba-rd-bar= (
BlockAck[+*HTC*+*RD*]+*a-mpdu*
BlockAckReq[+*HTC*+*RD*]+*a-mpdu*
) + *a-mpdu-end* |
(
BlockAck[+*HTC*+*RD*]+*a-mpdu*
1{**Data**[+*HTC*+*RD*]+*QoS*+*implicit-bar*+*a-mpdu*}
) + *a-mpdu-end*;

(* A PSMP burst is a sequence of PSMP sequence ending with a last-psmp-sequence *)
psmp-burst = {non-last-psmp-sequence} last-psmp-sequence;
non-last-psmp-sequence = **PSMP**+*more-psmp*+*QAP* downlink-phase uplink-phase;
last-psmp-sequence = **PSMP**+*no-more-psmp*+*QAP* downlink-phase uplink-phase;

(* The downlink phase is a sequence of allocations to STA as defined in the PSMP frame during which they may expect to receive. *)
downlink-phase = {psmp-allocated-time};

(* The uplink phase is a sequence of allocations to STA as defined in the PSMP frame during which they are allowed to transmit *)
uplink-phase = {psmp-allocated-time};

(* During a time allocation, one or more packets may be transmitted of contents defined by psmp-ppdu *)
psmp-allocated-time = 1{psmp-ppdu};

```
(* The packets that may be transmitted during PSMP are isolated Multi-TID BlockAck or Multi-TID
BlockAckReq frames (under an HT-immediate BlockAck policy), BlockAck or BlockAckReq frames
(under an HT-delayed or immediate BlockAck policy), isolated data frames, or an A-MPDU containing an
optional Multi-TID BlockAck frame and one or more data frames sent under the PSMP Ack Policy, or an A-
MPDU containing both Multi-TID BlockAck and Multi-TID BlockAckReq frames, but no data. Any
number of Action No Ack frames may be present in either A-MPDU. *)
psmp-ppdu =            Multi-TID BlockAck | (*HT-immediate*)
                       Multi-TID BlockAckReq | (*HT-immediate*)
                       BlockAck | (*HT-delayed or immediate*)
                       BlockAckReq | (*HT-delayed or immediate*)
                       Data[+HTC]+individual+QoS+psmp-ack |
                       (
                                 [Multi-TID BlockAck+a-mpdu]
                                 {Management+action-no-ack[+HTC] }
                                 1{Data[+HTC]+individual+QoS+psmp-ack+a-mpdu};
                       ) + a-mpdu-end |
                       (
                                 Multi-TID BlockAck+a-mpdu
                                 { Management+action-no-ack[+HTC] }
                                 Multi-TID BlockAckReq+a-mpdu
                       ) + a-mpdu-end;

(* A link adaptation exchange is a frame exchange sequence in which on-the-air signaling is used to control
or return the results of link measurements so that the initiator device can choose effective values for its
TXVECTOR parameters. *)
link-adaptation-exchange =
                       mcs-adaptation |
                       implict-txbf |
                       explicit-txbf;

(* An mcs-adaptation exchange includes an MCS measurement request and subsequent MFB. The MRQ
and MFB may be present in any +HTC frame. The exchange can occur either as a fast exchange, in which
the feedback is supplied in a response frame, an exchange in which the response is supplied along with some
other data frame within the same TXOP, or an exchange in which the response is supplied in a subsequent
TXOP won by the MCS responder. Only the fast response is shown in the syntax that follows. The
sequences shown below are representative examples only and are not exhaustive.*)
mcs-adaptation =
                   (* RTS/CTS *)
                   (RTS+HTC+mrq CTS+HTC+mfb) |

                   (* non-aggregated Data/ACK *)
                   (Data+HTC+QoS+mrq+normal-ack ACK+HTC+mfb) |

                   (* non-aggregated BlockAck *)
                   (BlockAckReq+HTC+mrq (BlockAck+HTC+mfb | ACK+HTC+mfb)) |

                   (* aggregated data with implicit Block Ack request and MRQ *)
                   (
                       (
                           1{Data[+HTC]+mrq [+rdg] +QoS+implicit-bar+a-mpdu}
                       ) + a-mpdu-end
                       (
                           (* Unaggregated BlockAck response *)
                           BlockAck+HTC +mfb |
```

(* Aggregated BlockAck response *)
(
BlockAck[+*HTC*+*mfb*] +*a-mpdu*
1{**Data**[+*HTC*+*mfb*]+*QoS*+(*no-ack*|*block-ack*)+*a-mpdu*}
) + *a-mpdu-end*
)
);

(* An implicit-txbf (implicit transmit beamforming) starts with the transmission of a request to sound the channel. The initiator measures the channel based on the sounding packet and updates its beamforming feedback matrices based on its observations of the sounding packet. No channel measurements are sent over the air.*)
implict-txbf =
(**RTS**+*HTC*+*trq* **(CTS**+*sounding* | **CTS**+*HTC*+*ndp-announce* **NDP**)) **|**
(**Data**+*HTC*+*trq*+*QoS*+*normal-ack*
(ACK+*sounding* | **ACK**+*HTC*+*ndp-announce* **NDP**)
) **|**
(**BlockAckReq**+*HTC*+*trq*
(BlockAck+*sounding* |
BlockAck+*HTC*+*ndp-announce* **NDP**
)
) **|**
(**BlockAck**+*HTC*+*trq*+*delayed*
(ACK+*sounding* |
ACK+*HTC*+*ndp-announce* **NDP**
)
)
(* The trq/sounding protocol also operates within aggregates. In this case the TRQ is carried in all +HTC frames (of which there has to be at least one) within the TRQ initiator's transmission. The response PPDU either is a sounding PPDU, or carries at least one +HTC frame with an ndp-announce, in which case the following PPDU is an NDP sounding PPDU. The following syntax is an simplified representation of this sequnce. *)
(**[BlockAck**+*HTC*+*trq*+*a-mpdu*] **{Data**+*HTC*+*trq*+*QoS*+*a-mpdu*}+*a-mpdu-end*)
(
(**[BlockAck**+*HTC*+*a-mpdu*]
{Data+*HTC*+*QoS*+*a-mpdu*}+*a-mpdu-end*+*sounding*)
) |
(
(**[BlockAck**+*HTC*+*ndp-announce*+*a-mpdu*]
{Data+*HTC*+*ndp-announce*+*QoS*+*a-mpdu*}+*a-mpdu-end*)
) **NDP** |
(**BlockAck**+*HTC*+*sounding*) |
(**BlockAck**+*HTC*+*ndp-announce NDP*);

(* During operation of explicit transmit beamforming (explicit-txbf), there are three encodings of feedback information. These are not distinguished here and are all identified by the *csi* attribute. The feedback position may be immediate, aggregate, or delayed. Immediate feedback follows a SIFS after the sounding PPDU (identified by the *sounding* attribute) or the NDP. Aggregate feedback occurs during an aggregate within the same TXOP and may accompany data frames in the same PPDU. Delayed feedback occurs during a subsequent TXOP during which the CSI responder is TXOP intiator. Only immediate feedback is described in the syntax below. The frame indicating any *csi-request* is carried in a sounding PPDU or

followed by an NDP. The CSI response is carried in an Action No Ack frame, which may be aggregated with the BlockAck or Ack response frame.
There are also two sets of sequences "staggered" and "NDP" that use either staggered sounding, or NDP sounding respectively.*)

```
explicit-txbf =                explicit-txbf-staggered | explicit-txbf-NDP;
```

(* Staggered sounding. In this case, the sounding request is present in a frame that also generates an immediate response. The response is aggregated with the feedback in an A-MPDU. *)

```
explicit-txbf-staggered =
                               (
                                         Data+HTC+csi-request+QoS+normal-ack+sounding
                                         (ACK+a-mpdu
                                         Management+action-no-ack +csi+a-mpdu-end)
                               ) |
                               (
                                         BlockAckReq+HTC+csi-request+sounding
                                         (BlockAck+a-mpdu
                                         Management+action-no-ack +csi+a-mpdu-end)
                               ) |
                               (
                                         BlockAckReq+HTC+csi-request+delayed+sounding
                                         (ACK+a-mpdu
                                         Management+action-no-ack+csi+a-mpdu-end)
                               ) ;
```

(* NDP sounding. In this case, the NDP announcement is present in a frame that also generates an immediate response. The beamformer transmits an NDP once the immediate response is received, and the beamforee transmits immediate feedback once it receives the NDP. *)

```
explicit-txbf-NDP =
                               (
                                         RTS+HTC+csi-request+ndp-announce
                                         CTS
                                         NDP
                                         Management+action-no-ack +csi
                               ) |
                               (
                                         Data+HTC+csi-request+QoS+normal-ack+ndp-announce
                                         ACK
                                         NDP
                                         Management+action-no-ack +csi
                               ) |
                               (
                                         BlockAckReq+HTC+csi-request+ndp-announce
                                         BlockAck
                                         NDP
                                         Management+action-no-ack +csi
                               ) |
                               (
                                         BlockAckReq+HTC+csi-request+delayed+ndp-announce
                                         ACK
                                         NDP
                                         Management+action-no-ack +csi
                               ) ;
```

Annex H

(normative)

Usage of Ethertype 89-0d

The Ethertype 89-0d frame body is specified in Figure H-1, omitting any possible security header and trailer.

	LLC	SNAP	Payload Type	Payload
Octets:	3	5	1	variable

Figure H-1—Ethertype 89-0d frame body

LLC is defined in ISO/IEC 8802-2:1998.

SNAP is defined in IEEE Std 802-2001. The formatting of the SNAP header is according to IETF RFC 1042. The Ethertype is set to 89-0d.

The Payload Type field is set to one of the values in Table H-1.

Table H-1—Payload Type field values

Protocol name	Payload type	Subclause
Remote Request/Response	1	12.10.3
TDLS	2	10.22.2
Reserved	3–255	

The Payload depends on the value inside the Payload Type field, and is defined in the subclauses listed in Table H-1.

Annex I

(informative)

Hopping sequences

The mechanisms described in this annex are obsolete. Consequently this annex may be removed in a future revision of this standard.

The following tables (Table I-1, Table I-2, and Table I-3) pertain to the hopping sequences for China, North America and ETSI.

Table I-1—Hopping sequence set 1

index	0	3	6	9	12	15	18	21	24	27	30	33	36
1	2	5	8	11	14	17	20	23	26	29	32	35	38
2	25	28	31	34	37	40	43	46	49	52	55	58	61
3	64	67	70	73	76	79	3	6	9	12	15	18	21
4	10	13	16	19	22	25	28	31	34	37	40	43	46
5	45	48	51	54	57	60	63	66	69	72	75	78	2
6	18	21	24	27	30	33	36	39	42	45	48	51	54
7	73	76	79	3	6	9	12	15	18	21	24	27	30
8	49	52	55	58	61	64	67	70	73	76	79	3	6
9	21	24	27	30	33	36	39	42	45	48	51	54	57
10	63	66	69	72	75	78	2	5	8	11	14	17	20
11	78	2	5	8	11	14	17	20	23	26	29	32	35
12	31	34	37	40	43	46	49	52	55	58	61	64	67
13	61	64	67	70	73	76	79	3	6	9	12	15	18
14	24	27	30	33	36	39	42	45	48	51	54	57	60
15	54	57	60	63	66	69	72	75	78	2	5	8	11
16	65	68	71	74	77	80	4	7	10	13	16	19	22
17	28	31	34	37	40	43	46	49	52	55	58	61	64
18	79	3	6	9	12	15	18	21	24	27	30	33	36
19	33	36	39	42	45	48	51	54	57	60	63	66	69
20	4	7	10	13	16	19	22	25	28	31	34	37	40
21	20	23	26	29	32	35	38	41	44	47	50	53	56
22	13	16	19	22	25	28	31	34	37	40	43	46	49
23	38	41	44	47	50	53	56	59	62	65	68	71	74
24	74	77	80	4	7	10	13	16	19	22	25	28	31
25	56	59	62	65	68	71	74	77	80	4	7	10	13
26	71	74	77	80	4	7	10	13	16	19	22	25	28
27	23	26	29	32	35	38	41	44	47	50	53	56	59
28	5	8	11	14	17	20	23	26	29	32	35	38	41
29	39	42	45	48	51	54	57	60	63	66	69	72	75
30	12	15	18	21	24	27	30	33	36	39	42	45	48
31	36	39	42	45	48	51	54	57	60	63	66	69	72
32	68	71	74	77	80	4	7	10	13	16	19	22	25
33	9	12	15	18	21	24	27	30	33	36	39	42	45
34	70	73	76	79	3	6	9	12	15	18	21	24	27
35	77	80	4	7	10	13	16	19	22	25	28	31	34
36	6	9	12	15	18	21	24	27	30	33	36	39	42
37	62	65	68	71	74	77	80	4	7	10	13	16	19
38	29	32	35	38	41	44	47	50	53	56	59	62	65
39	14	17	20	23	26	29	32	35	38	41	44	47	50

Table I-1—Hopping sequence set 1 *(continued)*

index	0	3	6	9	12	15	18	21	24	27	30	33	36
40	27	30	33	36	39	42	45	48	51	54	57	60	63
41	16	19	22	25	28	31	34	37	40	43	46	49	52
42	59	62	65	68	71	74	77	80	4	7	10	13	16
43	43	46	49	52	55	58	61	64	67	70	73	76	79
44	76	79	3	6	9	12	15	18	21	24	27	30	33
45	34	37	40	43	46	49	52	55	58	61	64	67	70
46	72	75	78	2	5	8	11	14	17	20	23	26	29
47	11	14	17	20	23	26	29	32	35	38	41	44	47
48	60	63	66	69	72	75	78	2	5	8	11	14	17
49	80	4	7	10	13	16	19	22	25	28	31	34	37
50	47	50	53	56	59	62	65	68	71	74	77	80	4
51	22	25	28	31	34	37	40	43	46	49	52	55	58
52	75	78	2	5	8	11	14	17	20	23	26	29	32
53	66	69	72	75	78	2	5	8	11	14	17	20	23
54	41	44	47	50	53	56	59	62	65	68	71	74	77
55	15	18	21	24	27	30	33	36	39	42	45	48	51
56	35	38	41	44	47	50	53	56	59	62	65	68	71
57	67	70	73	76	79	3	6	9	12	15	18	21	24
58	52	55	58	61	64	67	70	73	76	79	3	6	9
59	58	61	64	67	70	73	76	79	3	6	9	12	15
60	44	47	50	53	56	59	62	65	68	71	74	77	80
61	50	53	56	59	62	65	68	71	74	77	80	4	7
62	17	20	23	26	29	32	35	38	41	44	47	50	53
63	7	10	13	16	19	22	25	28	31	34	37	40	43
64	19	22	25	28	31	34	37	40	43	46	49	52	55
65	8	11	14	17	20	23	26	29	32	35	38	41	44
66	69	72	75	78	2	5	8	11	14	17	20	23	26
67	51	54	57	60	63	66	69	72	75	78	2	5	8
68	42	45	48	51	54	57	60	63	66	69	72	75	78
69	3	6	9	12	15	18	21	24	27	30	33	36	39
70	30	33	36	39	42	45	48	51	54	57	60	63	66
71	57	60	63	66	69	72	75	78	2	5	8	11	14
72	37	40	43	46	49	52	55	58	61	64	67	70	73
73	55	58	61	64	67	70	73	76	79	3	6	9	12
74	26	29	32	35	38	41	44	47	50	53	56	59	62
75	46	49	52	55	58	61	64	67	70	73	76	79	3
76	53	56	59	62	65	68	71	74	77	80	4	7	10
77	40	43	46	49	52	55	58	61	64	67	70	73	76
78	32	35	38	41	44	47	50	53	56	59	62	65	68
79	48	51	54	57	60	63	66	69	72	75	78	2	5

Table I-1—Hopping sequence set 1 *(continued)*

index	39	42	45	48	51	54	57	60	63	66	69	72	75
1	41	44	47	50	53	56	59	62	65	68	71	74	77
2	64	67	70	73	76	79	3	6	9	12	15	18	21
3	24	27	30	33	36	39	42	45	48	51	54	57	60
4	49	52	55	58	61	64	67	70	73	76	79	3	6
5	5	8	11	14	17	20	23	26	29	32	35	38	41
6	57	60	63	66	69	72	75	78	2	5	8	11	14
7	33	36	39	42	45	48	51	54	57	60	63	66	69
8	9	12	15	18	21	24	27	30	33	36	39	42	45
9	60	63	66	69	72	75	78	2	5	8	11	14	17
10	23	26	29	32	35	38	41	44	47	50	53	56	59
11	38	41	44	47	50	53	56	59	62	65	68	71	74
12	70	73	76	79	3	6	9	12	15	18	21	24	27
13	21	24	27	30	33	36	39	42	45	48	51	54	57
14	63	66	69	72	75	78	2	5	8	11	14	17	20
15	14	17	20	23	26	29	32	35	38	41	44	47	50
16	25	28	31	34	37	40	43	46	49	52	55	58	61
17	67	70	73	76	79	3	6	9	12	15	18	21	24
18	39	42	45	48	51	54	57	60	63	66	69	72	75
19	72	75	78	2	5	8	11	14	17	20	23	26	29
20	43	46	49	52	55	58	61	64	67	70	73	76	79
21	59	62	65	68	71	74	77	80	4	7	10	13	16
22	52	55	58	61	64	67	70	73	76	79	3	6	9
23	77	80	4	7	10	13	16	19	22	25	28	31	34
24	34	37	40	43	46	49	52	55	58	61	64	67	70
25	16	19	22	25	28	31	34	37	40	43	46	49	52
26	31	34	37	40	43	46	49	52	55	58	61	64	67
27	62	65	68	71	74	77	80	4	7	10	13	16	19
28	44	47	50	53	56	59	62	65	68	71	74	77	80
29	78	2	5	8	11	14	17	20	23	26	29	32	35
30	51	54	57	60	63	66	69	72	75	78	2	5	8
31	75	78	2	5	8	11	14	17	20	23	26	29	32
32	28	31	34	37	40	43	46	49	52	55	58	61	64
33	48	51	54	57	60	63	66	69	72	75	78	2	5
34	30	33	36	39	42	45	48	51	54	57	60	63	66
35	37	40	43	46	49	52	55	58	61	64	67	70	73
36	45	48	51	54	57	60	63	66	69	72	75	78	2
37	22	25	28	31	34	37	40	43	46	49	52	55	58
38	68	71	74	77	80	4	7	10	13	16	19	22	25
39	53	56	59	62	65	68	71	74	77	80	4	7	10

Table I-1—Hopping sequence set 1 *(continued)*

index	39	42	45	48	51	54	57	60	63	66	69	72	75
40	66	69	72	75	78	2	5	8	11	14	17	20	23
41	55	58	61	64	67	70	73	76	79	3	6	9	12
42	19	22	25	28	31	34	37	40	43	46	49	52	55
43	3	6	9	12	15	18	21	24	27	30	33	36	39
44	36	39	42	45	48	51	54	57	60	63	66	69	72
45	73	76	79	3	6	9	12	15	18	21	24	27	30
46	32	35	38	41	44	47	50	53	56	59	62	65	68
47	50	53	56	59	62	65	68	71	74	77	80	4	7
48	20	23	26	29	32	35	38	41	44	47	50	53	56
49	40	43	46	49	52	55	58	61	64	67	70	73	76
50	7	10	13	16	19	22	25	28	31	34	37	40	43
51	61	64	67	70	73	76	79	3	6	9	12	15	18
52	35	38	41	44	47	50	53	56	59	62	65	68	71
53	26	29	32	35	38	41	44	47	50	53	56	59	62
54	80	4	7	10	13	16	19	22	25	28	31	34	37
55	54	57	60	63	66	69	72	75	78	2	5	8	11
56	74	77	80	4	7	10	13	16	19	22	25	28	31
57	27	30	33	36	39	42	45	48	51	54	57	60	63
58	12	15	18	21	24	27	30	33	36	39	42	45	48
59	18	21	24	27	30	33	36	39	42	45	48	51	54
60	4	7	10	13	16	19	22	25	28	31	34	37	40
61	10	13	16	19	22	25	28	31	34	37	40	43	46
62	56	59	62	65	68	71	74	77	80	4	7	10	13
63	46	49	52	55	58	61	64	67	70	73	76	79	3
64	58	61	64	67	70	73	76	79	3	6	9	12	15
65	47	50	53	56	59	62	65	68	71	74	77	80	4
66	29	32	35	38	41	44	47	50	53	56	59	62	65
67	11	14	17	20	23	26	29	32	35	38	41	44	47
68	2	5	8	11	14	17	20	23	26	29	32	35	38
69	42	45	48	51	54	57	60	63	66	69	72	75	78
70	69	72	75	78	2	5	8	11	14	17	20	23	26
71	17	20	23	26	29	32	35	38	41	44	47	50	53
72	76	79	3	6	9	12	15	18	21	24	27	30	33
73	15	18	21	24	27	30	33	36	39	42	45	48	51
74	65	68	71	74	77	80	4	7	10	13	16	19	22
75	6	9	12	15	18	21	24	27	30	33	36	39	42
76	13	16	19	22	25	28	31	34	37	40	43	46	49
77	79	3	6	9	12	15	18	21	24	27	30	33	36
78	71	74	77	80	4	7	10	13	16	19	22	25	28
79	8	11	14	17	20	23	26	29	32	35	38	41	44

Table I-2—Hopping sequence set 2

index	1	4	7	10	13	16	19	22	25	28	31	34	37
1	3	6	9	12	15	18	21	24	27	30	33	36	39
2	26	29	32	35	38	41	44	47	50	53	56	59	62
3	65	68	71	74	77	80	4	7	10	13	16	19	22
4	11	14	17	20	23	26	29	32	35	38	41	44	47
5	46	49	52	55	58	61	64	67	70	73	76	79	3
6	19	22	25	28	31	34	37	40	43	46	49	52	55
7	74	77	80	4	7	10	13	16	19	22	25	28	31
8	50	53	56	59	62	65	68	71	74	77	80	4	7
9	22	25	28	31	34	37	40	43	46	49	52	55	58
10	64	67	70	73	76	79	3	6	9	12	15	18	21
11	79	3	6	9	12	15	18	21	24	27	30	33	36
12	32	35	38	41	44	47	50	53	56	59	62	65	68
13	62	65	68	71	74	77	80	4	7	10	13	16	19
14	25	28	31	34	37	40	43	46	49	52	55	58	61
15	55	58	61	64	67	70	73	76	79	3	6	9	12
16	66	69	72	75	78	2	5	8	11	14	17	20	23
17	29	32	35	38	41	44	47	50	53	56	59	62	65
18	80	4	7	10	13	16	19	22	25	28	31	34	37
19	34	37	40	43	46	49	52	55	58	61	64	67	70
20	5	8	11	14	17	20	23	26	29	32	35	38	41
21	21	24	27	30	33	36	39	42	45	48	51	54	57
22	14	17	20	23	26	29	32	35	38	41	44	47	50
23	39	42	45	48	51	54	57	60	63	66	69	72	75
24	75	78	2	5	8	11	14	17	20	23	26	29	32
25	57	60	63	66	69	72	75	78	2	5	8	11	14
26	72	75	78	2	5	8	11	14	17	20	23	26	29
27	24	27	30	33	36	39	42	45	48	51	54	57	60
28	6	9	12	15	18	21	24	27	30	33	36	39	42
29	40	43	46	49	52	55	58	61	64	67	70	73	76
30	13	16	19	22	25	28	31	34	37	40	43	46	49
31	37	40	43	46	49	52	55	58	61	64	67	70	73
32	69	72	75	78	2	5	8	11	14	17	20	23	26
33	10	13	16	19	22	25	28	31	34	37	40	43	46
34	71	74	77	80	4	7	10	13	16	19	22	25	28
35	78	2	5	8	11	14	17	20	23	26	29	32	35
36	7	10	13	16	19	22	25	28	31	34	37	40	43
37	63	66	69	72	75	78	2	5	8	11	14	17	20
38	30	33	36	39	42	45	48	51	54	57	60	63	66
39	15	18	21	24	27	30	33	36	39	42	45	48	51

Table I-2—Hopping sequence set 2 *(continued)*

index	1	4	7	10	13	16	19	22	25	28	31	34	37
40	28	31	34	37	40	43	46	49	52	55	58	61	64
41	17	20	23	26	29	32	35	38	41	44	47	50	53
42	60	63	66	69	72	75	78	2	5	8	11	14	17
43	44	47	50	53	56	59	62	65	68	71	74	77	80
44	77	80	4	7	10	13	16	19	22	25	28	31	34
45	35	38	41	44	47	50	53	56	59	62	65	68	71
46	73	76	79	3	6	9	12	15	18	21	24	27	30
47	12	15	18	21	24	27	30	33	36	39	42	45	48
48	61	64	67	70	73	76	79	3	6	9	12	15	18
49	2	5	8	11	14	17	20	23	26	29	32	35	38
50	48	51	54	57	60	63	66	69	72	75	78	2	5
51	23	26	29	32	35	38	41	44	47	50	53	56	59
52	76	79	3	6	9	12	15	18	21	24	27	30	33
53	67	70	73	76	79	3	6	9	12	15	18	21	24
54	42	45	48	51	54	57	60	63	66	69	72	75	78
55	16	19	22	25	28	31	34	37	40	43	46	49	52
56	36	39	42	45	48	51	54	57	60	63	66	69	72
57	68	71	74	77	80	4	7	10	13	16	19	22	25
58	53	56	59	62	65	68	71	74	77	80	4	7	10
59	59	62	65	68	71	74	77	80	4	7	10	13	16
60	45	48	51	54	57	60	63	66	69	72	75	78	2
61	51	54	57	60	63	66	69	72	75	78	2	5	8
62	18	21	24	27	30	33	36	39	42	45	48	51	54
63	8	11	14	17	20	23	26	29	32	35	38	41	44
64	20	23	26	29	32	35	38	41	44	47	50	53	56
65	9	12	15	18	21	24	27	30	33	36	39	42	45
66	70	73	76	79	3	6	9	12	15	18	21	24	27
67	52	55	58	61	64	67	70	73	76	79	3	6	9
68	43	46	49	52	55	58	61	64	67	70	73	76	79
69	4	7	10	13	16	19	22	25	28	31	34	37	40
70	31	34	37	40	43	46	49	52	55	58	61	64	67
71	58	61	64	67	70	73	76	79	3	6	9	12	15
72	38	41	44	47	50	53	56	59	62	65	68	71	74
73	56	59	62	65	68	71	74	77	80	4	7	10	13
74	27	30	33	36	39	42	45	48	51	54	57	60	63
75	47	50	53	56	59	62	65	68	71	74	77	80	4
76	54	57	60	63	66	69	72	75	78	2	5	8	11
77	41	44	47	50	53	56	59	62	65	68	71	74	77
78	33	36	39	42	45	48	51	54	57	60	63	66	69
79	49	52	55	58	61	64	67	70	73	76	79	3	6

Table I-2—Hopping sequence set 2 *(continued)*

index	40	43	46	49	52	55	58	61	64	67	70	73	76
1	42	45	48	51	54	57	60	63	66	69	72	75	78
2	65	68	71	74	77	80	4	7	10	13	16	19	22
3	25	28	31	34	37	40	43	46	49	52	55	58	61
4	50	53	56	59	62	65	68	71	74	77	80	4	7
5	6	9	12	15	18	21	24	27	30	33	36	39	42
6	58	61	64	67	70	73	76	79	3	6	9	12	15
7	34	37	40	43	46	49	52	55	58	61	64	67	70
8	10	13	16	19	22	25	28	31	34	37	40	43	46
9	61	64	67	70	73	76	79	3	6	9	12	15	18
10	24	27	30	33	36	39	42	45	48	51	54	57	60
11	39	42	45	48	51	54	57	60	63	66	69	72	75
12	71	74	77	80	4	7	10	13	16	19	22	25	28
13	22	25	28	31	34	37	40	43	46	49	52	55	58
14	64	67	70	73	76	79	3	6	9	12	15	18	21
15	15	18	21	24	27	30	33	36	39	42	45	48	51
16	26	29	32	35	38	41	44	47	50	53	56	59	62
17	68	71	74	77	80	4	7	10	13	16	19	22	25
18	40	43	46	49	52	55	58	61	64	67	70	73	76
19	73	76	79	3	6	9	12	15	18	21	24	27	30
20	44	47	50	53	56	59	62	65	68	71	74	77	80
21	60	63	66	69	72	75	78	2	5	8	11	14	17
22	53	56	59	62	65	68	71	74	77	80	4	7	10
23	78	2	5	8	11	14	17	20	23	26	29	32	35
24	35	38	41	44	47	50	53	56	59	62	65	68	71
25	17	20	23	26	29	32	35	38	41	44	47	50	53
26	32	35	38	41	44	47	50	53	56	59	62	65	68
27	63	66	69	72	75	78	2	5	8	11	14	17	20
28	45	48	51	54	57	60	63	66	69	72	75	78	2
29	79	3	6	9	12	15	18	21	24	27	30	33	36
30	52	55	58	61	64	67	70	73	76	79	3	6	9
31	76	79	3	6	9	12	15	18	21	24	27	30	33
32	29	32	35	38	41	44	47	50	53	56	59	62	65
33	49	52	55	58	61	64	67	70	73	76	79	3	6
34	31	34	37	40	43	46	49	52	55	58	61	64	67
35	38	41	44	47	50	53	56	59	62	65	68	71	74
36	46	49	52	55	58	61	64	67	70	73	76	79	3
37	23	26	29	32	35	38	41	44	47	50	53	56	59
38	69	72	75	78	2	5	8	11	14	17	20	23	26
39	54	57	60	63	66	69	72	75	78	2	5	8	11

Table I-2—Hopping sequence set 2 *(continued)*

index	40	43	46	49	52	55	58	61	64	67	70	73	76
40	67	70	73	76	79	3	6	9	12	15	18	21	24
41	56	59	62	65	68	71	74	77	80	4	7	10	13
42	20	23	26	29	32	35	38	41	44	47	50	53	56
43	4	7	10	13	16	19	22	25	28	31	34	37	40
44	37	40	43	46	49	52	55	58	61	64	67	70	73
45	74	77	80	4	7	10	13	16	19	22	25	28	31
46	33	36	39	42	45	48	51	54	57	60	63	66	69
47	51	54	57	60	63	66	69	72	75	78	2	5	8
48	21	24	27	30	33	36	39	42	45	48	51	54	57
49	41	44	47	50	53	56	59	62	65	68	71	74	77
50	8	11	14	17	20	23	26	29	32	35	38	41	44
51	62	65	68	71	74	77	80	4	7	10	13	16	19
52	36	39	42	45	48	51	54	57	60	63	66	69	72
53	27	30	33	36	39	42	45	48	51	54	57	60	63
54	2	5	8	11	14	17	20	23	26	29	32	35	38
55	55	58	61	64	67	70	73	76	79	3	6	9	12
56	75	78	2	5	8	11	14	17	20	23	26	29	32
57	28	31	34	37	40	43	46	49	52	55	58	61	64
58	13	16	19	22	25	28	31	34	37	40	43	46	49
59	19	22	25	28	31	34	37	40	43	46	49	52	55
60	5	8	11	14	17	20	23	26	29	32	35	38	41
61	11	14	17	20	23	26	29	32	35	38	41	44	47
62	57	60	63	66	69	72	75	78	2	5	8	11	14
63	47	50	53	56	59	62	65	68	71	74	77	80	4
64	59	62	65	68	71	74	77	80	4	7	10	13	16
65	48	51	54	57	60	63	66	69	72	75	78	2	5
66	30	33	36	39	42	45	48	51	54	57	60	63	66
67	12	15	18	21	24	27	30	33	36	39	42	45	48
68	3	6	9	12	15	18	21	24	27	30	33	36	39
69	43	46	49	52	55	58	61	64	67	70	73	76	79
70	70	73	76	79	3	6	9	12	15	18	21	24	27
71	18	21	24	27	30	33	36	39	42	45	48	51	54
72	77	80	4	7	10	13	16	19	22	25	28	31	34
73	16	19	22	25	28	31	34	37	40	43	46	49	52
74	66	69	72	75	78	2	5	8	11	14	17	20	23
75	7	10	13	16	19	22	25	28	31	34	37	40	43
76	14	17	20	23	26	29	32	35	38	41	44	47	50
77	80	4	7	10	13	16	19	22	25	28	31	34	37
78	72	75	78	2	5	8	11	14	17	20	23	26	29
79	9	12	15	18	21	24	27	30	33	36	39	42	45

Table I-3—Hopping sequence set 3

index	2	5	8	11	14	17	20	23	26	29	32	35	38
1	4	7	10	13	16	19	22	25	28	31	34	37	40
2	27	30	33	36	39	42	45	48	51	54	57	60	63
3	66	69	72	75	78	2	5	8	11	14	17	20	23
4	12	15	18	21	24	27	30	33	36	39	42	45	48
5	47	50	53	56	59	62	65	68	71	74	77	80	4
6	20	23	26	29	32	35	38	41	44	47	50	53	56
7	75	78	2	5	8	11	14	17	20	23	26	29	32
8	51	54	57	60	63	66	69	72	75	78	2	5	8
9	23	26	29	32	35	38	41	44	47	50	53	56	59
10	65	68	71	74	77	80	4	7	10	13	16	19	22
11	80	4	7	10	13	16	19	22	25	28	31	34	37
12	33	36	39	42	45	48	51	54	57	60	63	66	69
13	63	66	69	72	75	78	2	5	8	11	14	17	20
14	26	29	32	35	38	41	44	47	50	53	56	59	62
15	56	59	62	65	68	71	74	77	80	4	7	10	13
16	67	70	73	76	79	3	6	9	12	15	18	21	24
17	30	33	36	39	42	45	48	51	54	57	60	63	66
18	2	5	8	11	14	17	20	23	26	29	32	35	38
19	35	38	41	44	47	50	53	56	59	62	65	68	71
20	6	9	12	15	18	21	24	27	30	33	36	39	42
21	22	25	28	31	34	37	40	43	46	49	52	55	58
22	15	18	21	24	27	30	33	36	39	42	45	48	51
23	40	43	46	49	52	55	58	61	64	67	70	73	76
24	76	79	3	6	9	12	15	18	21	24	27	30	33
25	58	61	64	67	70	73	76	79	3	6	9	12	15
26	73	76	79	3	6	9	12	15	18	21	24	27	30
27	25	28	31	34	37	40	43	46	49	52	55	58	61
28	7	10	13	16	19	22	25	28	31	34	37	40	43
29	41	44	47	50	53	56	59	62	65	68	71	74	77
30	14	17	20	23	26	29	32	35	38	41	44	47	50
31	38	41	44	47	50	53	56	59	62	65	68	71	74
32	70	73	76	79	3	6	9	12	15	18	21	24	27
33	11	14	17	20	23	26	29	32	35	38	41	44	47
34	72	75	78	2	5	8	11	14	17	20	23	26	29
35	79	3	6	9	12	15	18	21	24	27	30	33	36
36	8	11	14	17	20	23	26	29	32	35	38	41	44
37	64	67	70	73	76	79	3	6	9	12	15	18	21
38	31	34	37	40	43	46	49	52	55	58	61	64	67
39	16	19	22	25	28	31	34	37	40	43	46	49	52

Table I-3—Hopping sequence set 3 *(continued)*

index	2	5	8	11	14	17	20	23	26	29	32	35	38
40	29	32	35	38	41	44	47	50	53	56	59	62	65
41	18	21	24	27	30	33	36	39	42	45	48	51	54
42	61	64	67	70	73	76	79	3	6	9	12	15	18
43	45	48	51	54	57	60	63	66	69	72	75	78	2
44	78	2	5	8	11	14	17	20	23	26	29	32	35
45	36	39	42	45	48	51	54	57	60	63	66	69	72
46	74	77	80	4	7	10	13	16	19	22	25	28	31
47	13	16	19	22	25	28	31	34	37	40	43	46	49
48	62	65	68	71	74	77	80	4	7	10	13	16	19
49	3	6	9	12	15	18	21	24	27	30	33	36	39
50	49	52	55	58	61	64	67	70	73	76	79	3	6
51	24	27	30	33	36	39	42	45	48	51	54	57	60
52	77	80	4	7	10	13	16	19	22	25	28	31	34
53	68	71	74	77	80	4	7	10	13	16	19	22	25
54	43	46	49	52	55	58	61	64	67	70	73	76	79
55	17	20	23	26	29	32	35	38	41	44	47	50	53
56	37	40	43	46	49	52	55	58	61	64	67	70	73
57	69	72	75	78	2	5	8	11	14	17	20	23	26
58	54	57	60	63	66	69	72	75	78	2	5	8	11
59	60	63	66	69	72	75	78	2	5	8	11	14	17
60	46	49	52	55	58	61	64	67	70	73	76	79	3
61	52	55	58	61	64	67	70	73	76	79	3	6	9
62	19	22	25	28	31	34	37	40	43	46	49	52	55
63	9	12	15	18	21	24	27	30	33	36	39	42	45
64	21	24	27	30	33	36	39	42	45	48	51	54	57
65	10	13	16	19	22	25	28	31	34	37	40	43	46
66	71	74	77	80	4	7	10	13	16	19	22	25	28
67	53	56	59	62	65	68	71	74	77	80	4	7	10
68	44	47	50	53	56	59	62	65	68	71	74	77	80
69	5	8	11	14	17	20	23	26	29	32	35	38	41
70	32	35	38	41	44	47	50	53	56	59	62	65	68
71	59	62	65	68	71	74	77	80	4	7	10	13	16
72	39	42	45	48	51	54	57	60	63	66	69	72	75
73	57	60	63	66	69	72	75	78	2	5	8	11	14
74	28	31	34	37	40	43	46	49	52	55	58	61	64
75	48	51	54	57	60	63	66	69	72	75	78	2	5
76	55	58	61	64	67	70	73	76	79	3	6	9	12
77	42	45	48	51	54	57	60	63	66	69	72	75	78
78	34	37	40	43	46	49	52	55	58	61	64	67	70
79	50	53	56	59	62	65	68	71	74	77	80	4	7

Table I-3—Hopping sequence set 3 *(continued)*

index	41	44	47	50	53	56	59	62	65	68	71	74	77
1	43	46	49	52	55	58	61	64	67	70	73	76	79
2	66	69	72	75	78	2	5	8	11	14	17	20	23
3	26	29	32	35	38	41	44	47	50	53	56	59	62
4	51	54	57	60	63	66	69	72	75	78	2	5	8
5	7	10	13	16	19	22	25	28	31	34	37	40	43
6	59	62	65	68	71	74	77	80	4	7	10	13	16
7	35	38	41	44	47	50	53	56	59	62	65	68	71
8	11	14	17	20	23	26	29	32	35	38	41	44	47
9	62	65	68	71	74	77	80	4	7	10	13	16	19
10	25	28	31	34	37	40	43	46	49	52	55	58	61
11	40	43	46	49	52	55	58	61	64	67	70	73	76
12	72	75	78	2	5	8	11	14	17	20	23	26	29
13	23	26	29	32	35	38	41	44	47	50	53	56	59
14	65	68	71	74	77	80	4	7	10	13	16	19	22
15	16	19	22	25	28	31	34	37	40	43	46	49	52
16	27	30	33	36	39	42	45	48	51	54	57	60	63
17	69	72	75	78	2	5	8	11	14	17	20	23	26
18	41	44	47	50	53	56	59	62	65	68	71	74	77
19	74	77	80	4	7	10	13	16	19	22	25	28	31
20	45	48	51	54	57	60	63	66	69	72	75	78	2
21	61	64	67	70	73	76	79	3	6	9	12	15	18
22	54	57	60	63	66	69	72	75	78	2	5	8	11
23	79	3	6	9	12	15	18	21	24	27	30	33	36
24	36	39	42	45	48	51	54	57	60	63	66	69	72
25	18	21	24	27	30	33	36	39	42	45	48	51	54
26	33	36	39	42	45	48	51	54	57	60	63	66	69
27	64	67	70	73	76	79	3	6	9	12	15	18	21
28	46	49	52	55	58	61	64	67	70	73	76	79	3
29	80	4	7	10	13	16	19	22	25	28	31	34	37
30	53	56	59	62	65	68	71	74	77	80	4	7	10
31	77	80	4	7	10	13	16	19	22	25	28	31	34
32	30	33	36	39	42	45	48	51	54	57	60	63	66
33	50	53	56	59	62	65	68	71	74	77	80	4	7
34	32	35	38	41	44	47	50	53	56	59	62	65	68
35	39	42	45	48	51	54	57	60	63	66	69	72	75
36	47	50	53	56	59	62	65	68	71	74	77	80	4
37	24	27	30	33	36	39	42	45	48	51	54	57	60
38	70	73	76	79	3	6	9	12	15	18	21	24	27
39	55	58	61	64	67	70	73	76	79	3	6	9	12

Table I-3—Hopping sequence set 3 *(continued)*

index	41	44	47	50	53	56	59	62	65	68	71	74	77
40	68	71	74	77	80	4	7	10	13	16	19	22	25
41	57	60	63	66	69	72	75	78	2	5	8	11	14
42	21	24	27	30	33	36	39	42	45	48	51	54	57
43	5	8	11	14	17	20	23	26	29	32	35	38	41
44	38	41	44	47	50	53	56	59	62	65	68	71	74
45	75	78	2	5	8	11	14	17	20	23	26	29	32
46	34	37	40	43	46	49	52	55	58	61	64	67	70
47	52	55	58	61	64	67	70	73	76	79	3	6	9
48	22	25	28	31	34	37	40	43	46	49	52	55	58
49	42	45	48	51	54	57	60	63	66	69	72	75	78
50	9	12	15	18	21	24	27	30	33	36	39	42	45
51	63	66	69	72	75	78	2	5	8	11	14	17	20
52	37	40	43	46	49	52	55	58	61	64	67	70	73
53	28	31	34	37	40	43	46	49	52	55	58	61	64
54	3	6	9	12	15	18	21	24	27	30	33	36	39
55	56	59	62	65	68	71	74	77	80	4	7	10	13
56	76	79	3	6	9	12	15	18	21	24	27	30	33
57	29	32	35	38	41	44	47	50	53	56	59	62	65
58	14	17	20	23	26	29	32	35	38	41	44	47	50
59	20	23	26	29	32	35	38	41	44	47	50	53	56
60	6	9	12	15	18	21	24	27	30	33	36	39	42
61	12	15	18	21	24	27	30	33	36	39	42	45	48
62	58	61	64	67	70	73	76	79	3	6	9	12	15
63	48	51	54	57	60	63	66	69	72	75	78	2	5
64	60	63	66	69	72	75	78	2	5	8	11	14	17
65	49	52	55	58	61	64	67	70	73	76	79	3	6
66	31	34	37	40	43	46	49	52	55	58	61	64	67
67	13	16	19	22	25	28	31	34	37	40	43	46	49
68	4	7	10	13	16	19	22	25	28	31	34	37	40
69	44	47	50	53	56	59	62	65	68	71	74	77	80
70	71	74	77	80	4	7	10	13	16	19	22	25	28
71	19	22	25	28	31	34	37	40	43	46	49	52	55
72	78	2	5	8	11	14	17	20	23	26	29	32	35
73	17	20	23	26	29	32	35	38	41	44	47	50	53
74	67	70	73	76	79	3	6	9	12	15	18	21	24
75	8	11	14	17	20	23	26	29	32	35	38	41	44
76	15	18	21	24	27	30	33	36	39	42	45	48	51
77	2	5	8	11	14	17	20	23	26	29	32	35	38
78	73	76	79	3	6	9	12	15	18	21	24	27	30
79	10	13	16	19	22	25	28	31	34	37	40	43	46

Annex J

(informative)

Formal description of a subset of MAC operation

J.1 Status of this annex

This annex is obsolete. Consequently this annex may be removed in a future revision of this standard. This clause is no longer maintained and may not be compatible with or describe all features of this standard.

J.2 Overview

This annex contains formal descriptions of the behavior of a subset of MAC STA and AP entities. These descriptions also describe the frame formats and the generation and interpretation of information encoded in MAC frames, in the parameters of service primitives supported by the MAC, and in MIB attributes used or generated by the MAC. The MAC is described using the 1992 version of the ITU Specification and Description Language (SDL-92). SDL-92 is defined in ITU-T Recommendation Z.100 (03/93). An update to ITU-T Recommendation Z.100 was approved in 1996 (SDL-96), but none of the SDL facilities used in this annex were modified. An introduction to the MAC formal description is provided in J.3. Definitions of the data types and operators used by the MAC state machines are provided in J.4. An SDL system describing MAC operation at an IEEE 802.11 STA is contained in J.5. Finally, a subset of an SDL system describing the aspects of MAC operation at an IEEE 802.11 AP that differ from operation at a non-AP STA is provided in J.6.

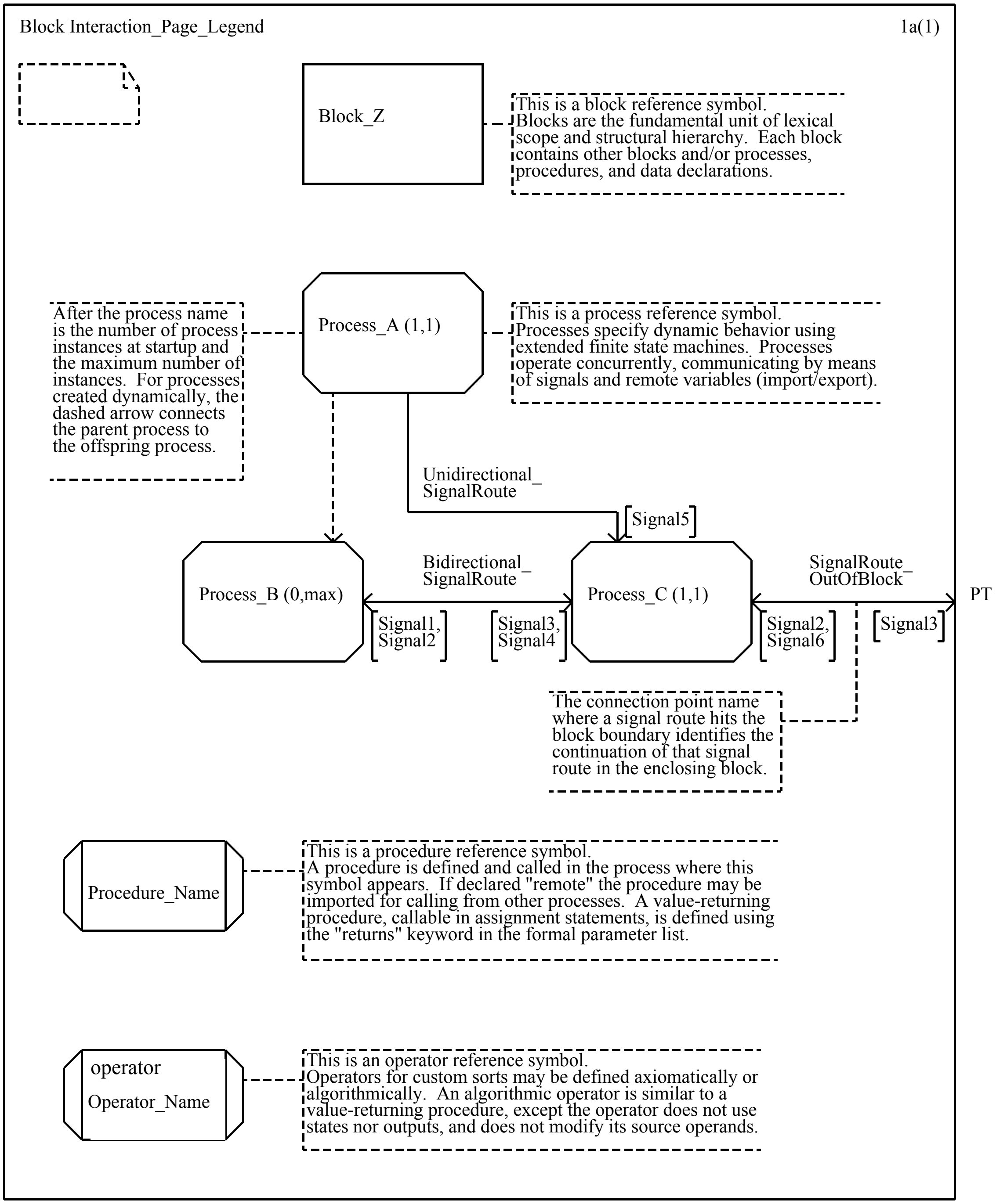
In Annex C, the MAC and PHY MIBs are described in Abstract Syntax Notation One (ASN.1), defined in ISO/IEC 8824-1:1995, ISO/IEC 8824-2:1995, ISO/IEC 8824-3:1995, ISO/IEC 8824-4:1995, ISO/IEC 8825-1:1995, and ISO/IEC 8825-2:1996. ITU-T Recommendation Z.105 (03/95) defines the use of SDL in conjunction with ASN.1, allowing system behavior to be defined using SDL and data types to be defined using ASN.1. Incomplete tool support precluded the use of ITU-T Recommendation Z.105 in this annex. However, within the limits of ITU-T Recommendation Z.100 (referred to subsequently as Z.100), the data types in J.4 are defined in a similar manner to ITU-T Recommendation Z.105 (referred to subsequently as Z.105). Annex A contains a listing of available documentation.

NOTE 1—The SDL definitions in this annex are expected to be usable with any SDL tool that supports the 1993 version or 1996 update of ITU-T Recommendation Z.100. Software for generating, analyzing, verifying, and simulating SDL system descriptions is available from several sources.

NOTE 2—The SDL code in this annex was generated using *SDT/PC version 3.02*; from Telelogic AB, Malmo, Sweden (+46-40-174700; internet: telelogic.se); U.S. office in Princeton, NJ (+1-609-520-1935; internet: telelogic.com). Telelogic offers SDT for several workstation platforms in addition to SDT/PC.

NOTE 3— The use of Telelogic's product to prepare this annex does not constitute an endorsement of SDT by the IEEE LAN MAN Standards Committee or by the IEEE.

NOTE 4—The diagrams on the next two pages show most of the symbols of SDL graphical syntax (SDL-GR) used in the MAC formal description. The symbols in these diagrams have labels and comments that explain their meanings. These diagrams are intended to serve as a legend for the SDL-GR symbols that comprise most of the process interaction and state transition diagrams. These diagrams are neither a complete SDL system, nor a complete presentation of SDL-GR symbology. Also, this state machine fragment exists to illustrate the SDL graphical syntax and does not describe any useful behavior.



This is a block reference symbol. Blocks are the fundamental unit of lexical scope and structural hierarchy. Each block contains other blocks and/or processes, procedures, and data declarations.

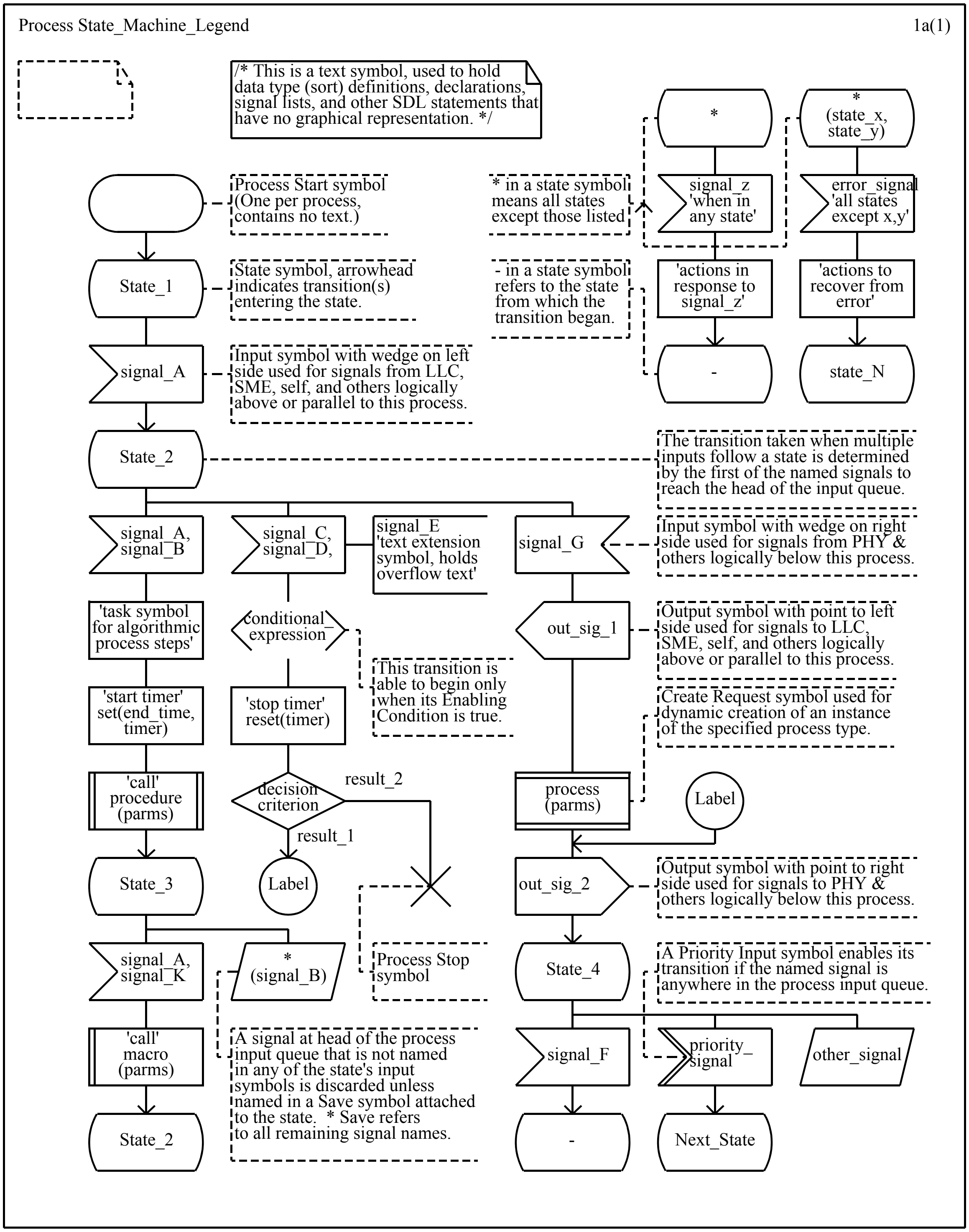
After the process name is the number of process instances at startup and the maximum number of instances. For processes created dynamically, the dashed arrow connects the parent process to the offspring process.

This is a process reference symbol. Processes specify dynamic behavior using extended finite state machines. Processes operate concurrently, communicating by means of signals and remote variables (import/export).

The connection point name where a signal route hits the block boundary identifies the continuation of that signal route in the enclosing block.

This is a procedure reference symbol. A procedure is defined and called in the process where this symbol appears. If declared "remote" the procedure may be imported for calling from other processes. A value-returning procedure, callable in assignment statements, is defined using the "returns" keyword in the formal parameter list.

This is an operator reference symbol. Operators for custom sorts may be defined axiomatically or algorithmically. An algorithmic operator is similar to a value-returning procedure, except the operator does not use states nor outputs, and does not modify its source operands.



/* This is a text symbol, used to hold data type (sort) definitions, declarations, signal lists, and other SDL statements that have no graphical representation. */

Process Start symbol (One per process, contains no text.)

State symbol, arrowhead indicates transition(s) entering the state.

Input symbol with wedge on left side used for signals from LLC, SME, self, and others logically above or parallel to this process.

* in a state symbol means all states except those listed

- in a state symbol refers to the state from which the transition began.

The transition taken when multiple inputs follow a state is determined by the first of the named signals to reach the head of the input queue.

Input symbol with wedge on right side used for signals from PHY & others logically below this process.

Output symbol with point to left side used for signals to LLC, SME, self, and others logically above or parallel to this process.

This transition is able to begin only when its Enabling Condition is true.

Create Request symbol used for dynamic creation of an instance of the specified process type.

Output symbol with point to right side used for signals to PHY & others logically below this process.

Process Stop symbol

A Priority Input symbol enables its transition if the named signal is anywhere in the process input queue.

A signal at head of the process input queue that is not named in any of the state's input symbols is discarded unless named in a Save symbol attached to the state. * Save refers to all remaining signal names.

J.3 Introduction to the MAC formal description

This formal description defines the behavior of IEEE 802.11 MAC entities. The MAC protocol functional decomposition used herein facilitates explicit description of the reference points and durations of the various timed intervals; the bases for generation and/or validation of header fields, service parameters, and MIB attributes; and the interpretation of each value in cases where enumerated data types are used in service parameters.

J.3.1 Fundamental assumptions

The MAC protocol is described as an SDL system, which is a set of extended finite state machines. Each state machine is a set of independent processes, all of which operate concurrently. All variable data-holding entities and procedures exist solely within the context of a single process. In SDL all interprocess communication is done with signals (there are no global variables). Signals may be sent and received explicitly, using SDL’s output and input symbols, or implicitly, using SDL’s export/import mechanism (only if the variables or procedures are declared “remote”). By default, signals incur delays when traversing channels between blocks; however, only nondelaying channels and signal routes are used in the MAC state machines, and all remote variables and procedures are declared with the “nodelay” property.

State transitions, procedure calls, and tasks (assignment statements and other algorithmic processing steps) are assumed to require zero time. This permits the time intervals that are part of the normative MAC behavior to be defined explicitly, using SDL timers. One unit of system time (a 1.0 change in the value of “now”) is assumed to represent 1 µs of real time. Usec (microsecond) and TU data types are defined, with operators to convert Usec and TU values to SDL time or duration when necessary.

The SDL system boundary encloses the MAC entities. The LLC, SME, PHY, and DS are part of the environment. SDL generally assumes that entities in the environment operate as specified; however, the MAC state machines that communicate with the various SAPs attempt to validate inputs from the environment, and to handle cases where a pair of communicating entities, one within the system and the other outside the system boundary, have different local views of the medium, STA, or service state. All STAs in an IEEE 802.11 service set are assumed to exhibit the behaviors described herein. Nevertheless, because of the open nature of the WM, the MAC state machines check for error cases that can arise only when an entity on the WM is transmitting IEEE 802.11 protocol data units (PDUs), but is not obeying the communication protocols specified by this standard.

J.3.2 Notation conventions

When practical, names used in the clauses of this standard are spelled identically in this annex. The principal exceptions are those names that conflict with one of SDL’s reserved words (such as power management mode “active,” which is renamed “sta_active” in SDL). To help fit the SDL text into the graphic symbols, acronyms with multiple, sequential capital letters are written with only the first letter capitalized (e.g., “MSDU” is written “Msdu” and “MLMEJoin.request” is written “MlmeJoin.request”).

SDL reserved words and the names of variables and synonyms (named constants) begin with lowercase letters. The names of sorts (data types), signals, signal routes, channels, blocks, and processes begin with uppercase letters. The names of certain groups of variables and/or synonyms begin with a particular lowercase letter, followed by the remainder of the name, beginning with an uppercase letter. These groups are

“aNameOfAttribute”PHY operational parameters.
“cNameOfCapability”Capability bits, also used for internal values exported as MIB counters.
“dNameOfDuration”Duration (relative time) values, declared as Usec, TU, or Duration.
“dot11NameOfAttribute”MIB attributes.

“eNameOfElement”Element ID values.

“mNameOfVariable”Remote variables used for intra-MAC communication, but not part of the MIB. Most of these variables are exported from the MLME block.

“sNameOfStaticValue”Synonyms for static data values used within the MAC.

“tNameOfTime”Time (absolute time) values, declared as Usec, TU, or Time. The names of timers begin with “T.”

J.3.3 Modeling techniques

State machines are grouped according to defined function sets that are visible, directly or indirectly, at an exposed interface. The emphasis in the organization of the state machines is explicitly to show initiation of and response to events at the exposed interfaces, and time-related actions, including those dependent on the absence of external events (e.g., response timeouts) and intervals measured in derived units (e.g., backoff “time” in units of slots during which the WM is idle). The operations associated with the various state transitions emphasize communication functions. Most of the details regarding insertion, extraction, and encoding of information in fields of the PDUs is encapsulated with the definitions of those fields. This approach, which relies heavily on SDL’s abstract data type and inheritance mechanisms, permits the behavior of the data-holding entities to be precisely defined, without obscuring process flow by adding in-line complexity to the individual state transitions.

The modeling of PDUs and service data units (SDUs) requires sorts such as octet strings, and operators such as bitwise boolean functions, which are not predefined in SDL. These sorts and operators are defined in Package macsorts, which appears in J.4.

PDU and SDU sorts are based on the **Bit** sort. Bit is a subtype of SDL’s predefined Boolean sort. As a result, Bit literals 0 and 1 are alternative names for “false” and “true” and have no numeric significance. To use 0 or 1 as integer values requires a conversion operation. Items of the **Bitstring** sort are 0-origin, variable-length strings of Bits. With Bitstring operands, operators “and,” “or,” “xor,” and “not” operate bitwise, with the length of the result equal to the length of the longest (or only) source string. The **Octet** sort is a subtype of Bitstring that adds conversion operators to and from Integer. Each item of the Octet sort has length=8 {by usage convention in Z.100, enforced in Z.105}. Items of the **Octetstring** sort are 0-origin, variable-length strings of Octets. The **Frame** sort is a subtype of Octetstring that adds operators to extract and to modify all MAC header fields and most other MAC frame fields and elements. Most MAC fields and elements that contain named values with specific value assignments or enumerations are defined as subtypes of Frame, Octetstring, or Bitstring with the names added as literals or synonyms, so that the state machines can refer to the names without introducing ambiguity about the value encodings.

Where communication at a SAP or between processes is strictly first in first out (FIFO), the (implicit) input queue of the SDL processes is used. When more sophisticated queue management is needed, a queue whose entries are instances of one, specified sort is created using the **Queue** generator. Entries on Queue sorts may be added and removed at either the tail or the head, and the number of queue entries may be determined. The contents of a Queue may also be searched to locate entries with particular parameter values.

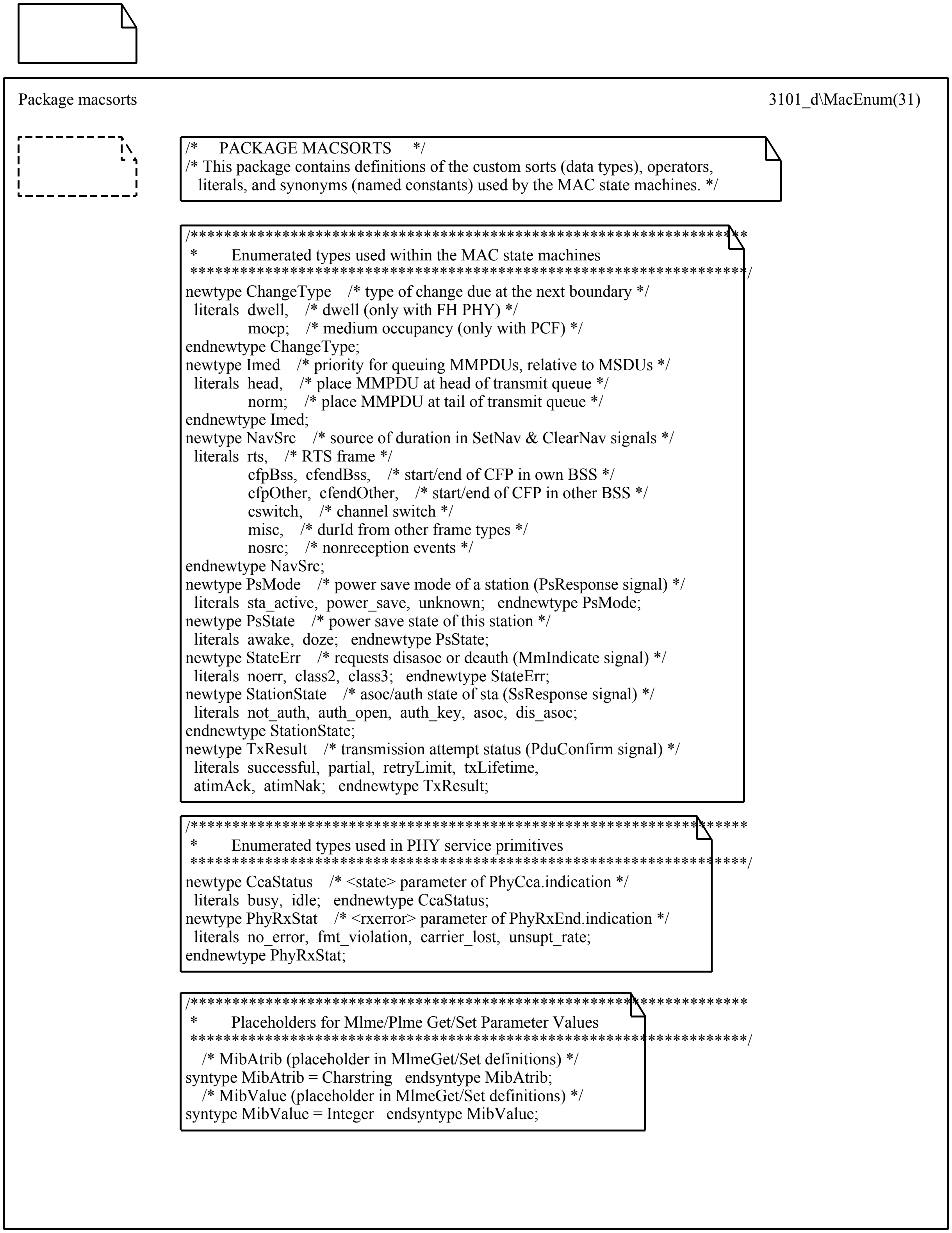
In J.4 is an SDL-92 Package (a named collection of SDL definitions that can be included by reference into an SDL System specification), which is a formal description of the formats and data encodings used in IEEE 802.11 SDUs, PDUs, and the parameters of the service primitives used at each of the SAPs supported by the IEEE 802.11 MAC. This package also contains definitions for some data structures and operators used internally by one or more of the MAC state machines.

The behaviors of many intra-MAC operators are part of the normative description of the MAC protocol because results of the specified operations are visible, directly or indirectly, at exposed interfaces. For example, custom operators are used to define the generation of the CRC-32 value used in the FCS field (operator crc32, page 424), the calculation of frame transmission time used as part of the value in the

Duration/ID field in certain types of frames (operator calcDur, page 440), the comparison of the values of particular fields of a received MAC header with cached data values as part of the procedure for detecting duplicate frames (operator searchTupleCache, page 412), and numerous other aspects of frame formats and information encoding. On the other hand, data structures used solely for intra-MAC storage or for transferring of information between different state machines of a single STA or AP, are only normative to the extent that they define items of internal state and the temporal sequence necessary for proper operation of the MAC protocol. The specific structures and encodings used for internal data storage and communication functions in this formal description do *not* constrain MAC implementations, provided those implementations exhibit the specified behaviors at the defined SAPs and, in conjunction with an appropriate PHY, on the WM.

J.4 Data type and operator definitions for the MAC state machines

This clause is in SDL/PR (phrase notation), with the exception of procedural operators, which are defined in SDL/GR (graphic notation). Package macsorts contains the definitions of the sorts (data types with associated operators and literals) and synonyms (named constants) used by the MAC state machines. Package macmib defines data types for attributes in the MAC MIB, and portions of the PHY MIB, accessed by the MAC state machines. Package macmib exists solely to satisfy SDL’s strong type checking in the absence of an SDL tool that fully supports Z.105 (the combined use of SDL with ASN.1).



```
/*     PACKAGE MACSORTS      */
/* This package contains definitions of the custom sorts (data types), operators,
   literals, and synonyms (named constants) used by the MAC state machines. */
```

```
/*******************************************************************
 *       Enumerated types used within the MAC state machines
 ******************************************************************/
newtype ChangeType    /* type of change due at the next boundary */
  literals  dwell,    /* dwell (only with FH PHY) */
            mocp;    /* medium occupancy (only with PCF) */
endnewtype ChangeType;
newtype Imed    /* priority for queuing MMPDUs, relative to MSDUs */
  literals  head,    /* place MMPDU at head of transmit queue */
            norm;    /* place MMPDU at tail of transmit queue */
endnewtype Imed;
newtype NavSrc    /* source of duration in SetNav & ClearNav signals */
  literals  rts,    /* RTS frame */
            cfpBss,  cfendBss,    /* start/end of CFP in own BSS */
            cfpOther,  cfendOther,    /* start/end of CFP in other BSS */
            cswitch,    /* channel switch */
            misc,    /* durId from other frame types */
            nosrc;    /* nonreception events */
endnewtype NavSrc;
newtype PsMode    /* power save mode of a station (PsResponse signal) */
  literals  sta_active,  power_save,  unknown;   endnewtype PsMode;
newtype PsState    /* power save state of this station */
  literals  awake,  doze;   endnewtype PsState;
newtype StateErr    /* requests disasoc or deauth (MmIndicate signal) */
  literals  noerr,  class2,  class3;   endnewtype StateErr;
newtype StationState    /* asoc/auth state of sta (SsResponse signal) */
  literals  not_auth,  auth_open,  auth_key,  asoc,  dis_asoc;
endnewtype StationState;
newtype TxResult    /* transmission attempt status (PduConfirm signal) */
  literals  successful,  partial,  retryLimit,  txLifetime,
  atimAck,  atimNak;   endnewtype TxResult;
```

```
/*******************************************************************
 *       Enumerated types used in PHY service primitives
 ******************************************************************/
newtype CcaStatus    /* <state> parameter of PhyCca.indication */
  literals  busy,  idle;   endnewtype CcaStatus;
newtype PhyRxStat    /* <rxerror> parameter of PhyRxEnd.indication */
  literals  no_error,  fmt_violation,  carrier_lost,  unsupt_rate;
endnewtype PhyRxStat;
```

```
/*******************************************************************
 *       Placeholders for Mlme/Plme Get/Set Parameter Values
 ******************************************************************/
   /* MibAtrib (placeholder in MlmeGet/Set definitions) */
syntype MibAtrib = Charstring   endsyntype MibAtrib;
   /* MibValue (placeholder in MlmeGet/Set definitions) */
syntype MibValue = Integer   endsyntype MibValue;
```

Package macsorts

3102_d\LmeEnum(31)

```
/*******************************************************************
 *       Enumerated types used in Mac and Mlme service primitives
 *******************************************************************/
newtype AuthType    /* <authentication type> parm in Mlme primitives */
  inherits Octetstring  operators all;
  adding  literals open_system,  shared_key;
  axioms  open_system == mkOS(0, 2);  shared_key == mkOS(1, 2);
endnewtype AuthType;
newtype AuthTypeSet  powerset( AuthType);   endnewtype AuthTypeSet;
newtype BssType    /* <BSS type> parameter & BSS description element */
  literals infrastructure,  independent,  any_bss;   endnewtype BssType;
newtype BssTypeSet  powerset( BssType);   endnewtype BssTypeSet;
newtype CfPriority    /* <priority> parameter of various requests */
  literals  contention,  contentionFree;   endnewtype CfPriority;
newtype MibStatus    /* <status> parm of Mlme/Plme Get/Set.confirm */
  literals  success,   invalid,   write_only,   read_only;
endnewtype MibStatus;
newtype MlmeStatus    /* <status> parm of Mlme operation confirm */
  literals  success,  invalid,  timeout,  refused,
            tomany_req,  already_bss;   endnewtype MlmeStatus;
newtype PwrSave    /* <power save mode> parameter of MlmePowerMgt */
  literals  sta_active,  power_save;   endnewtype PwrSave;
newtype Routing    /* <routing info> parameter for MAC data service */
  literals  null_rt;   endnewtype Routing;
newtype RxStatus    /* <reception status> parm of MaUnitdata indication */
  literals  rx_success,  rx_failure;   endnewtype RxStatus;
newtype ScanType    /* <scan type> parameter of MlmeScan.request */
  literals  active_scan,  passive_scan;   endnewtype ScanType;
newtype ServiceClass    /* <service class> parameter for MaUnitdata */
  literals  reorderable,  strictlyOrdered;   endnewtype ServiceClass;
newtype TxStatus    /* <transmission status> parm of MaUnitdataStatus */
  literals  successful,   retryLimit,   txLifetime,   noBss,
      excessiveDataLength,  nonNullSourceRouting,
      unsupportedPriority,  unavailablePriority,
      unsupportedServiceClass,  unavailableServiceClass,
      unavailableKeyMapping;  endnewtype TxStatus;
```

Package macsorts 3103_e\IntraMac(31)

```
/*******************************************************************
 *       Intra-MAC remote variables (names of form mXYZ)
 ******************************************************************/
remote mActingAsAp Boolean nodelay;    /* =true if STA started BSS */
remote mAId  AsocId nodelay;    /* AID assigned to STA by AP */
remote mAssoc  Boolean nodelay;    /* =true if STA associated w/BSS */
remote mAtimW  Boolean nodelay;    /* =true if ATIM window in prog */
remote mBkIP  Boolean nodelay;    /* =true if backoff in prog */
remote mBrates Ratestring nodelay;    /* basic rate set for this sta */
remote mBssId  MacAddr nodelay;    /* identifier of current (I)BSS */
remote mCap  Octetstring nodelay;    /* capability info from MlmeJoin */
remote mCfp  Boolean nodelay;    /* =true if CF period in progress */
remote mDisable  Boolean nodelay;    /* =true if not in any BSS; then */
     /* TX only sends probe_req; RX only accepts beacon, probe_rsp */
remote mDtimCount  Integer nodelay;    /* =0 at Tbtt of Beacon with DTIM */
remote mFxIP  Boolean nodelay;    /* =true during frame exchange seq */
remote mIbss  Boolean nodelay;    /* =true if STA is member of IBSS */
remote mListenInt  Integer nodelay;    /* beacons between wake up @TBTT */
remote mNavEnd  Time nodelay;    /* NAV end Time, <=now when idle */
remote mNextBdry  Time nodelay;    /* next boundary Time; =0 if none */
remote mNextTbtt  Time nodelay;    /* Time next beacon due to occur */
remote mPcAvail  Boolean nodelay;    /* =true if point coord in BSS */
remote mPcDlvr  Boolean nodelay;    /* =true if CF delivery only */
remote mPcPoll  Boolean nodelay;    /* =true if CF delivery & polling */
remote mPdly  Usec nodelay;     /* probe delay from start or join */
remote mPss  PsState nodelay;    /* power save state of STA */
remote mReceiveDTIMs Boolean nodelay;  /* =true if DTIMs received */
remote mRxA  Boolean nodelay;    /* =true if RX indicated by PHY */
remote mSsId  Octetstring nodelay;    /* name of the current (I)BSS */
remote procedure TSF nodelay;    /* read & update 64-bit TSF timer */
   fpar Integer, Boolean;   returns Integer;
```

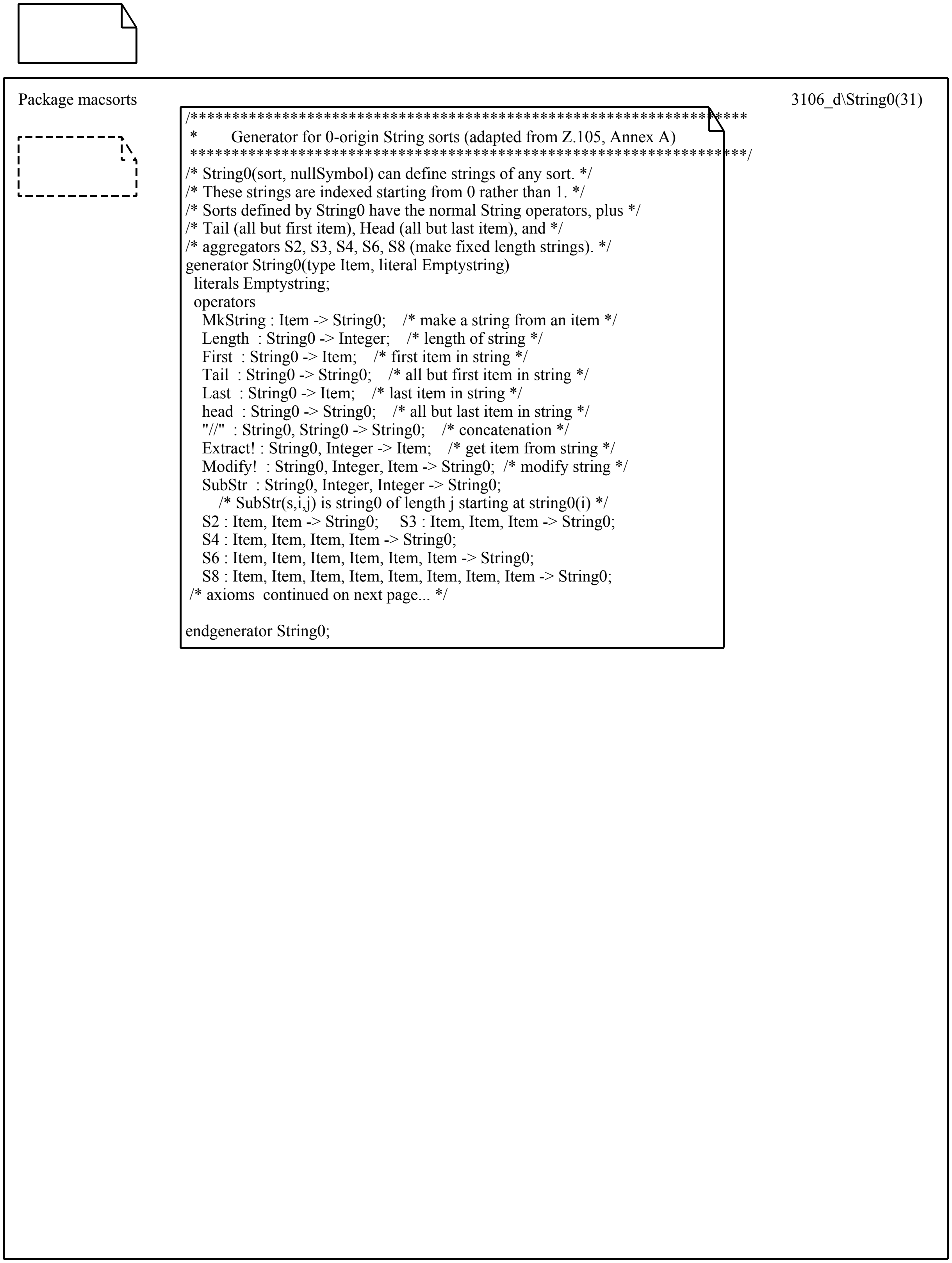
Package macsorts 3104_d\StaticData(31)

```
/*******************************************************************
 *       Named static data values    (names of form sXYZ)
 *******************************************************************/
synonym sMaxMsduLng Integer = 2304;    /* max octets in an MSDU */
synonym sMacHdrLng  Integer = 24;    /* octets in data header, no WEP */
synonym sWepHdrLng  Integer = 28;    /* octets in data header with WEP */
synonym sWepAddLng  Integer = 8;    /* octets added for WEP */
synonym sWdsAddLng  Integer = 6;    /* octets added for WDS (addr4) */
synonym sCrcLng  Integer = 4;    /* octets for crc32 (FCS, ICV) */
synonym sMaxMpduLng  Integer =        /* max octets in an MPDU */
   (sMaxMsduLng + sMacHdrLng + sWdsAddLng + sWepAddLng + sCrcLng);
syntype FrameIndexRange = Integer    /* index range for octets in MPDU */
   constants 0 : sMaxMpduLng   endsyntype FrameIndexRange;
synonym sTsOctet  Integer = 24;    /* first octet of Timestamp field */
synonym sMinFragLng Integer = 256;    /* min value for aMpduMaxLength */
synonym sMaxFragNum Integer =        /* maximum fragment number */
   (sMaxMsduLng / (sMinFragLng - sMacHdrLng - sCrcLng));
synonym sAckCtsLng Integer = 112;  /* bits in ACK and CTS frames */
```

```
/*******************************************************************
 *       Station configuration flags (static, supplementary to MIB)
 *******************************************************************/
synonym sVersion  Integer = 0;    /* supported Protocol Version */
synonym sCanBeAp  Boolean = false;    /* =true if STA can operate as AP */
synonym sCanBePc  Boolean = false;    /* =true if AP can be Point Coord */
synonym sCfPollable Boolean =true;    /* =true if responds to CF-polls */
```

Package macsorts 3105_d\Usec_TU(31)

```
/*******************************************************************
 *       Discrete microsecond and Time Unit sorts
 *******************************************************************/
/* SDL does not define the relationship between its concept */
/* of Time and physical time in the system being described. */
/* An abstraction is needed to establish this relationship, */
/* because Time in SDL uses the semantics of Real, whereas */
/* time in the MAC protocol is discrete, with the semantics */
/* of Natural and a step size (resolution) of 1 micosecond. */
/* Most MAC times are defined using the subtypes of Integer */
/* Usec and TU.  These have operators for explicit conversion */
/* to SDL Time (tUsec, tTU), SDL Duration (dUsec, dTU), and */
/* from SDL Time (uTime, tuTime) as needed to comply with SDL's */
/* strong type checking.  Where the MAC state machines need to */
/* access the contents of the TSF timer, SDL's 'now' (current */
/* time) is used.  This yields readable time-dependent code, */
/* but the value of 'now' cannot be modified by an SDL program, */
/* so adopting the TSF time from timestamps in received Beacons */
/* or Probe Responses is shown as an informal task symbol. */
/* Microsecond sort -- also has operators tmin and tmax */
newtype Usec   inherits Integer   operators all;
  adding  operators
    dUsec : Usec -> Duration;
    tUsec : Usec -> Time;
    uTime : Time -> Usec;
    tmax  : Usec, Usec -> Usec;
    tmin  : Usec, Usec -> Usec;
  axioms    for all u, w in Usec(
      u >= w ==> tmax(u, w) == u;      u < w ==> tmax(u, w) == w;
      u >= w ==> tmin(u, w) == w;      u < w ==> tmin(u, w) == u;
      for all t in Time(     for all r in Real(
          r = float(u) ==> tUsec(u) == Time!(Duration!(r));
          t = Time!(Duration!(r)) and u = fix(r) ==> u == uTime(t);));
      for all d in Duration(     for all r in Real(
          r = float(u) ==> dUsec(u) == Duration!(r); )));
  constants >= 0 /* constrain value range to be non-negative */
endnewtype Usec;
/* Time Unit sort -- (1 * TU) = (1024 * Usec) */
newtype TU  inherits Integer  operators all;
  adding  operators
    dTU    : TU -> Duration;
    tTU    : TU -> Time;
    tuTime : Time -> TU;
    u2TU   : Usec -> TU;
    tu2U   : TU -> Usec;
  axioms    for all k in TU(   for all t in Time(   for all r in Real(
          r = float(k) ==> tTU(k) == Time!(Duration!(1024 * r));
          t = Time!(Duration!(r)) and k = (fix(r) / 1024) ==> k == tuTime(t);));
      for all d in Duration(   for all r in Real(
          r = float(k) ==> dTU(k) == Duration!(1024 * r);));
      for all u in Usec(   u2TU(u) == u / 1024;   tu2U(k) == k * 1024; ));
  constants >= 0 /* constrain value range to be non-negative */
endnewtype TU;
```



```
/*******************************************************************
 *       Generator for 0-origin String sorts (adapted from Z.105, Annex A)
 *******************************************************************/
/* String0(sort, nullSymbol) can define strings of any sort. */
/* These strings are indexed starting from 0 rather than 1. */
/* Sorts defined by String0 have the normal String operators, plus */
/* Tail (all but first item), Head (all but last item), and */
/* aggregators S2, S3, S4, S6, S8 (make fixed length strings). */
generator String0(type Item, literal Emptystring)
  literals Emptystring;
  operators
    MkString : Item -> String0;    /* make a string from an item */
    Length  : String0 -> Integer;    /* length of string */
    First  : String0 -> Item;    /* first item in string */
    Tail  : String0 -> String0;    /* all but first item in string */
    Last  : String0 -> Item;    /* last item in string */
    head  : String0 -> String0;    /* all but last item in string */
    "//"  : String0, String0 -> String0;    /* concatenation */
    Extract! : String0, Integer -> Item;    /* get item from string */
    Modify!  : String0, Integer, Item -> String0;  /* modify string */
    SubStr  : String0, Integer, Integer -> String0;
        /* SubStr(s,i,j) is string0 of length j starting at string0(i) */
    S2 : Item, Item -> String0;     S3 : Item, Item, Item -> String0;
    S4 : Item, Item, Item, Item -> String0;
    S6 : Item, Item, Item, Item, Item, Item -> String0;
    S8 : Item, Item, Item, Item, Item, Item, Item, Item -> String0;
 /* axioms  continued on next page... */

endgenerator String0;
```

Package macsorts 3107_a\String0(31)

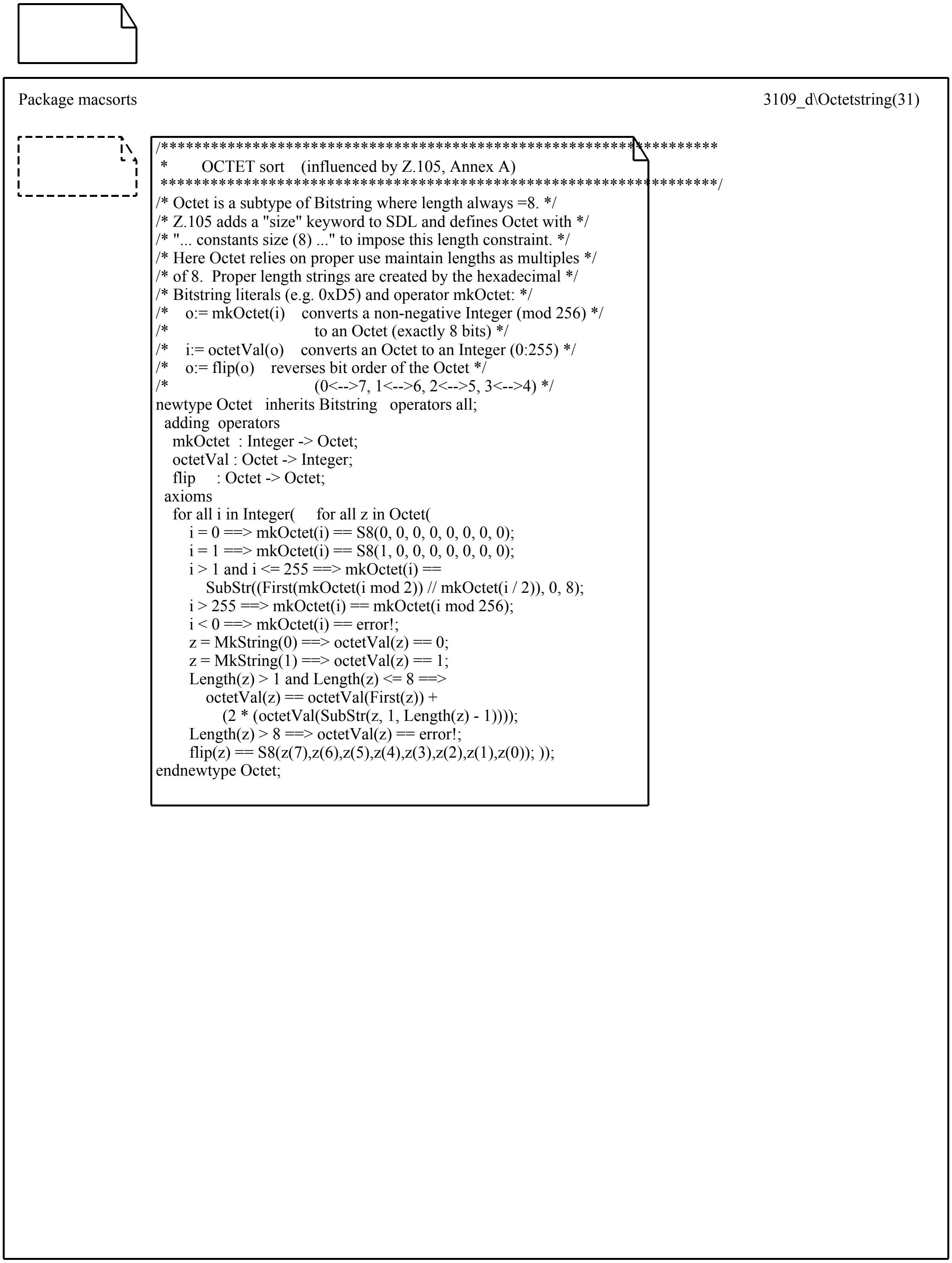
```
/* String0 axioms  */
/*   for all item0,item1,item2,item3,item4,item5,item6,item7 in Item(
     for all s, s1, S2, S3 in String0(     for all i, j in Integer(
     constructors are Emptystring, MkString, and "//";
      equalities between constructor terms
        s // Emptystring == s;         Emptystring // s == s;
        (s1 // S2) // S3 == s1 // (S2 // S3);
      definition of Length by applying it to all constructors
         type String Length(Emptystring) == 0;
         type String Length(MkString(item0)) == 1;
         type String Length(s1 // S2) == Length(s1) + Length(S2);
      definition of Extract! by applying it to all constructors,
        Extract!(MkString(item0), 0) == item0;
        i < Length(s1) ==> Extract!(s1 // S2, i) == Extract!(s1, i);
        i >= Length(s1) ==> Extract!(s1 // S2, i) == Extract!(S2, i - Length(s1));
        i < 0 or i >= Length(s) ==> Extract!(s, i) == error!;
      definition of First and Last by other operations
        First(s) == Extract!(s, 0);
        Last(s) == Extract!(s, Length(s) - 1);
      definition of substr(s,i,j) by induction on j,
        i >= 0 and i <= Length(s) ==> SubStr(s, i, 0) == Emptystring;
        i >= 0 and j > 0 and i + j <= Length(s) ==> SubStr(s, i, j) ==
                 SubStr(s, i, j - 1) // MkString(Extract!(s, i + j - 1));
        i < 0 or j < 0 or i + j > Length(s) ==> SubStr(s, i, j) == error!;
      definition of Modify!, Head, Tail, Sx by other operations
        Modify!(s, i, item0) == SubStr(s, 0, i) // MkString(item0) //
           SubStr(s, i + 1, Length(s) - i - 1);
        head(s) == SubStr(s, 0, Length(s) - 1);
        Tail(s) == SubStr(s, 1, Length(s) - 1);
        S2(item0, item1) == MkString(item0) // MkString(item1);
        S3(item0, item1, item2) ==
          MkString(item0) // MkString(item1) // MkString(item2);
        S4(item0, item1, item2, item3) ==
          MkString(item0) // MkString(item1) // MkString(item2) //
            MkString(item3);
        S6(item0, item1, item2, item3, item4, item5) ==
          MkString(item0) // MkString(item1) // MkString(item2) //
            MkString(item3) // MkString(item4) // MkString(item5);
        S8(item0, item1, item2, item3, item4, item5, item6, item7) ==
          MkString(item0) // MkString(item1) // MkString(item2) //
            MkString(item3) // MkString(item4) // MkString(item5) //
            MkString(item6) // MkString(item7); )));
 */
```

Package macsorts

3108_d\Bitstring(31)

```
/*******************************************************************
 *       ASN.1-style BIT sort    (from Z.105, Annex A)
 *******************************************************************/
   /* Bit is a subtype of Boolean -- bit values 0 and 1 are
   /* not numerals and cannot be used with Integer operators */
newtype Bit   inherits Boolean
   literals  0 = false,  1 = true;   operators all;   endnewtype Bit;
```

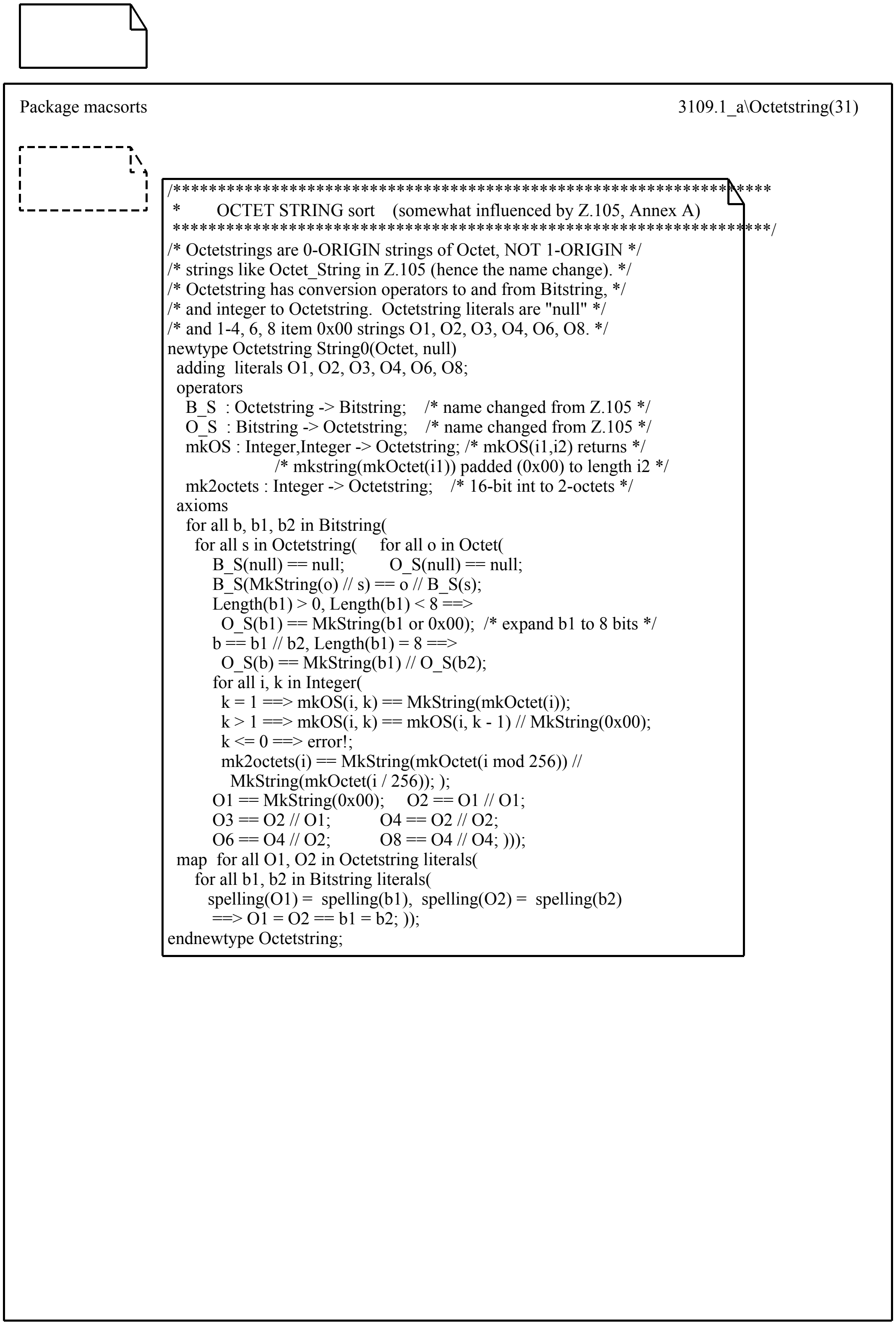
```
/*******************************************************************
 *       ASN.1-style BIT STRING sort    (adapted from Z.105, Annex A)
 *******************************************************************/
/* Bitstrings are 0-origin strings of Bit.  Z.105 uses ASN.1-style */
/* literals in binary ('1011'B) or hexadecimal ('D3'H), but this */
/* syntax is not accepted for Z.100 string literals.  Therefore, */
/* this version provides only hexadecimal literals 0x00-0xFF. */
/* Bitstring operators '=>', 'not', 'and', 'or', and 'xor' act */
/* bitwise, with the length of the result string equal to the */
/* length of the longest (or only) source string. */
newtype Bitstring String0(Bit, ")
  adding  literals     macro Hex_Literals;
  operators
    "not" : Bitstring -> Bitstring;
    "and" : Bitstring, Bitstring -> Bitstring;
    "or"  : Bitstring, Bitstring -> Bitstring;
    "xor" : Bitstring, Bitstring -> Bitstring;
    "=>"  : Bitstring, Bitstring -> Bitstring;     noequality;
  axioms     macro Hex_Axioms;
    for all s, s1, S2, S3 in Bitstring(
      s = s == true;      s1 = S2 == S2 = s1;
      s1 /= S2 == not (s1 = S2);      s1 = S2 == true ==> s1 == S2;
      ((s1 = S2) and (S2 = S3)) ==> s1 = S3 == true;
      ((s1 = S2) and (S2 /= S3)) ==> s1 = S3 == false;
      for all b, b1, b2 in Bit(
        not (") == ";
        not (MkString(b) // s) == MkString(not (b)) // not (s);
        " and " == ";
        Length(s) > 0 ==> " and s == MkString(0) and s;
        Length(s) > 0 ==> s and " == s and MkString(0);
        (MkString(b1) // s1) and (MkString(b2) // S2) ==
          MkString(b1 and b2) // (s1 and S2);
        s1 or S2 == not (not s1 and not S2);
        s1 xor S2 == (s1 or S2) and not (s1 and S2);
        s1 => S2 == not (not s1 and S2);));
  map  for all b1, b2 in Bitstring literals(
      for all bs1, bs2 in Charstring literals(
/* connection to the String generator */
        for all b in Bit literals(
            spelling(b1) = "" // bs1 // bs2 // "",
          spelling(b2) = "" // bs2 // "",  spelling(b) = bs1
            ==> b1 == MkString(b) // b2; )));
endnewtype Bitstring;
```



Package macsorts

3109_d\Octetstring(31)

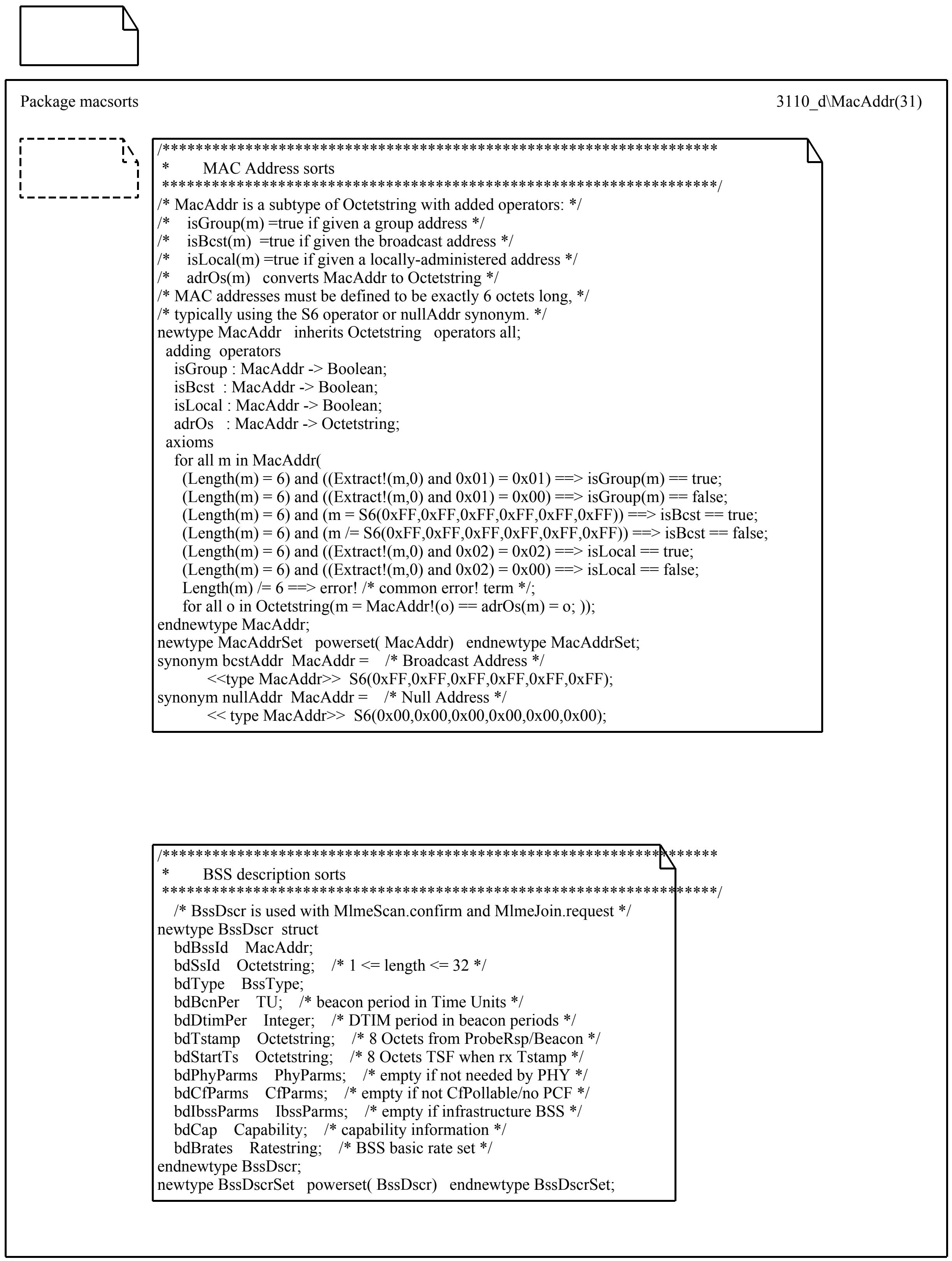
```
/*****************************************************************
 *      OCTET sort    (influenced by Z.105, Annex A)
 *****************************************************************/
/* Octet is a subtype of Bitstring where length always =8. */
/* Z.105 adds a "size" keyword to SDL and defines Octet with */
/* "... constants size (8) ..." to impose this length constraint. */
/* Here Octet relies on proper use maintain lengths as multiples */
/* of 8.  Proper length strings are created by the hexadecimal */
/* Bitstring literals (e.g. 0xD5) and operator mkOctet: */
/*    o:= mkOctet(i)    converts a non-negative Integer (mod 256) */
/*                             to an Octet (exactly 8 bits) */
/*    i:= octetVal(o)    converts an Octet to an Integer (0:255) */
/*    o:= flip(o)    reverses bit order of the Octet */
/*                             (0<-->7, 1<-->6, 2<-->5, 3<-->4) */
newtype Octet   inherits Bitstring   operators all;
  adding  operators
    mkOctet  : Integer -> Octet;
    octetVal : Octet -> Integer;
    flip     : Octet -> Octet;
  axioms
    for all i in Integer(     for all z in Octet(
        i = 0 ==> mkOctet(i) == S8(0, 0, 0, 0, 0, 0, 0, 0);
        i = 1 ==> mkOctet(i) == S8(1, 0, 0, 0, 0, 0, 0, 0);
        i > 1 and i <= 255 ==> mkOctet(i) ==
            SubStr((First(mkOctet(i mod 2)) // mkOctet(i / 2)), 0, 8);
        i > 255 ==> mkOctet(i) == mkOctet(i mod 256);
        i < 0 ==> mkOctet(i) == error!;
        z = MkString(0) ==> octetVal(z) == 0;
        z = MkString(1) ==> octetVal(z) == 1;
        Length(z) > 1 and Length(z) <= 8 ==>
            octetVal(z) == octetVal(First(z)) +
                (2 * (octetVal(SubStr(z, 1, Length(z) - 1))));
        Length(z) > 8 ==> octetVal(z) == error!;
        flip(z) == S8(z(7),z(6),z(5),z(4),z(3),z(2),z(1),z(0)); ));
endnewtype Octet;
```



Package macsorts

3109.1_a\Octetstring(31)

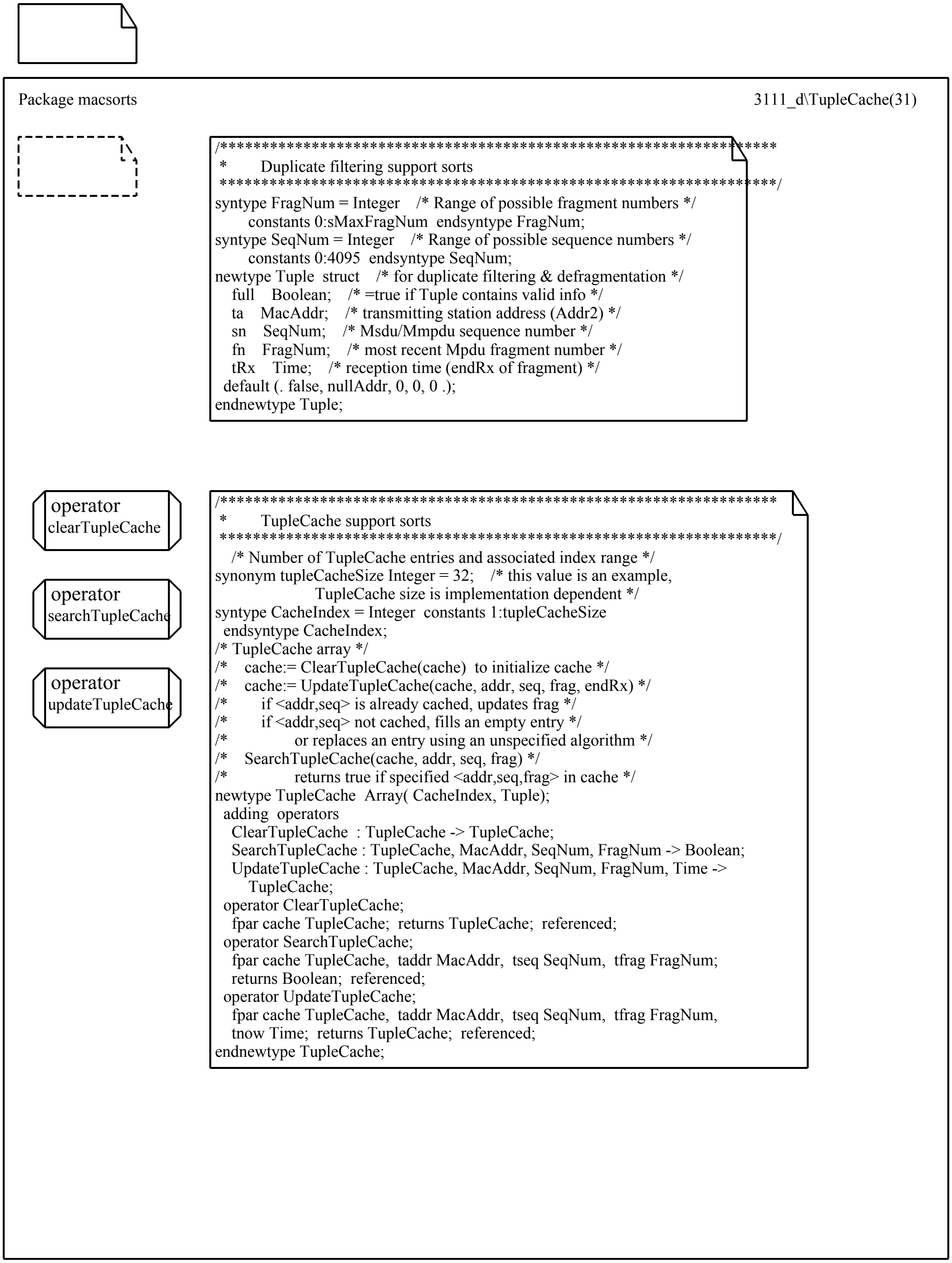
```
/*********************************************************************
 *       OCTET STRING sort    (somewhat influenced by Z.105, Annex A)
 *********************************************************************/
/* Octetstrings are 0-ORIGIN strings of Octet, NOT 1-ORIGIN */
/* strings like Octet_String in Z.105 (hence the name change). */
/* Octetstring has conversion operators to and from Bitstring, */
/* and integer to Octetstring.  Octetstring literals are "null" */
/* and 1-4, 6, 8 item 0x00 strings O1, O2, O3, O4, O6, O8. */
newtype Octetstring String0(Octet, null)
  adding  literals O1, O2, O3, O4, O6, O8;
  operators
    B_S  : Octetstring -> Bitstring;    /* name changed from Z.105 */
    O_S  : Bitstring -> Octetstring;    /* name changed from Z.105 */
    mkOS : Integer,Integer -> Octetstring; /* mkOS(i1,i2) returns */
                    /* mkstring(mkOctet(i1)) padded (0x00) to length i2 */
    mk2octets : Integer -> Octetstring;    /* 16-bit int to 2-octets */
  axioms
    for all b, b1, b2 in Bitstring(
      for all s in Octetstring(     for all o in Octet(
         B_S(null) == null;          O_S(null) == null;
         B_S(MkString(o) // s) == o // B_S(s);
         Length(b1) > 0, Length(b1) < 8 ==>
           O_S(b1) == MkString(b1 or 0x00);  /* expand b1 to 8 bits */
         b == b1 // b2, Length(b1) = 8 ==>
           O_S(b) == MkString(b1) // O_S(b2);
         for all i, k in Integer(
           k = 1 ==> mkOS(i, k) == MkString(mkOctet(i));
           k > 1 ==> mkOS(i, k) == mkOS(i, k - 1) // MkString(0x00);
           k <= 0 ==> error!;
           mk2octets(i) == MkString(mkOctet(i mod 256)) //
             MkString(mkOctet(i / 256)); );
         O1 == MkString(0x00);     O2 == O1 // O1;
         O3 == O2 // O1;           O4 == O2 // O2;
         O6 == O4 // O2;           O8 == O4 // O4; )));
  map  for all O1, O2 in Octetstring literals(
      for all b1, b2 in Bitstring literals(
        spelling(O1) =  spelling(b1),  spelling(O2) =  spelling(b2)
         ==> O1 = O2 == b1 = b2; ));
endnewtype Octetstring;
```



Package macsorts 3110_d\MacAddr(31)

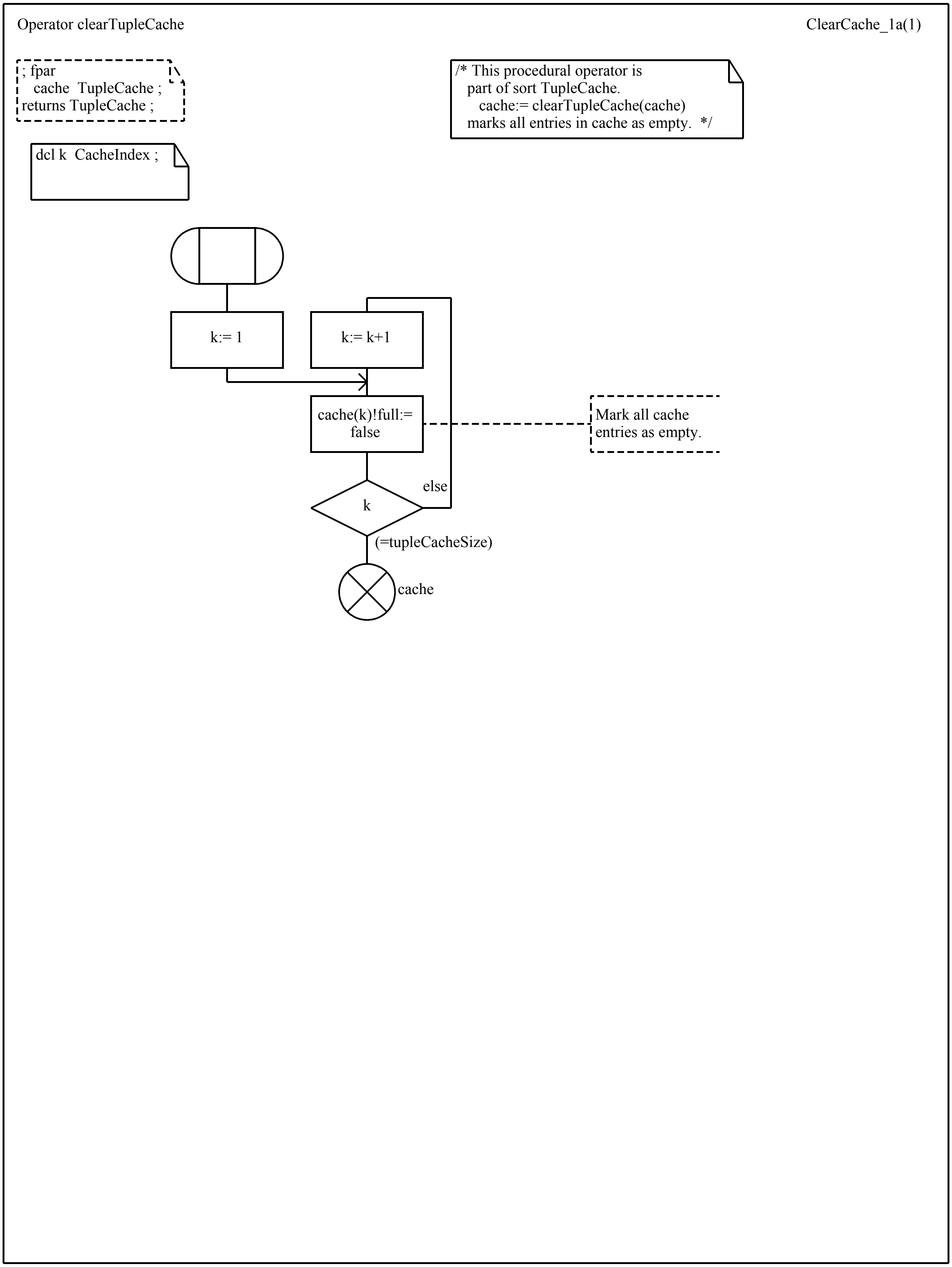
```
/******************************************************************
 *       MAC Address sorts
 ******************************************************************/
/* MacAddr is a subtype of Octetstring with added operators: */
/*    isGroup(m) =true if given a group address */
/*    isBcst(m)  =true if given the broadcast address */
/*    isLocal(m) =true if given a locally-administered address */
/*    adrOs(m)   converts MacAddr to Octetstring */
/* MAC addresses must be defined to be exactly 6 octets long, */
/* typically using the S6 operator or nullAddr synonym. */
newtype MacAddr   inherits Octetstring   operators all;
  adding  operators
    isGroup : MacAddr -> Boolean;
    isBcst  : MacAddr -> Boolean;
    isLocal : MacAddr -> Boolean;
    adrOs   : MacAddr -> Octetstring;
  axioms
    for all m in MacAddr(
      (Length(m) = 6) and ((Extract!(m,0) and 0x01) = 0x01) ==> isGroup(m) == true;
      (Length(m) = 6) and ((Extract!(m,0) and 0x01) = 0x00) ==> isGroup(m) == false;
      (Length(m) = 6) and (m = S6(0xFF,0xFF,0xFF,0xFF,0xFF,0xFF)) ==> isBcst == true;
      (Length(m) = 6) and (m /= S6(0xFF,0xFF,0xFF,0xFF,0xFF,0xFF)) ==> isBcst == false;
      (Length(m) = 6) and ((Extract!(m,0) and 0x02) = 0x02) ==> isLocal == true;
      (Length(m) = 6) and ((Extract!(m,0) and 0x02) = 0x00) ==> isLocal == false;
      Length(m) /= 6 ==> error! /* common error! term */;
      for all o in Octetstring(m = MacAddr!(o) == adrOs(m) = o; ));
endnewtype MacAddr;
newtype MacAddrSet   powerset( MacAddr)   endnewtype MacAddrSet;
synonym bcstAddr  MacAddr =    /* Broadcast Address */
          <<type MacAddr>>  S6(0xFF,0xFF,0xFF,0xFF,0xFF,0xFF);
synonym nullAddr  MacAddr =    /* Null Address */
          << type MacAddr>>  S6(0x00,0x00,0x00,0x00,0x00,0x00);
```

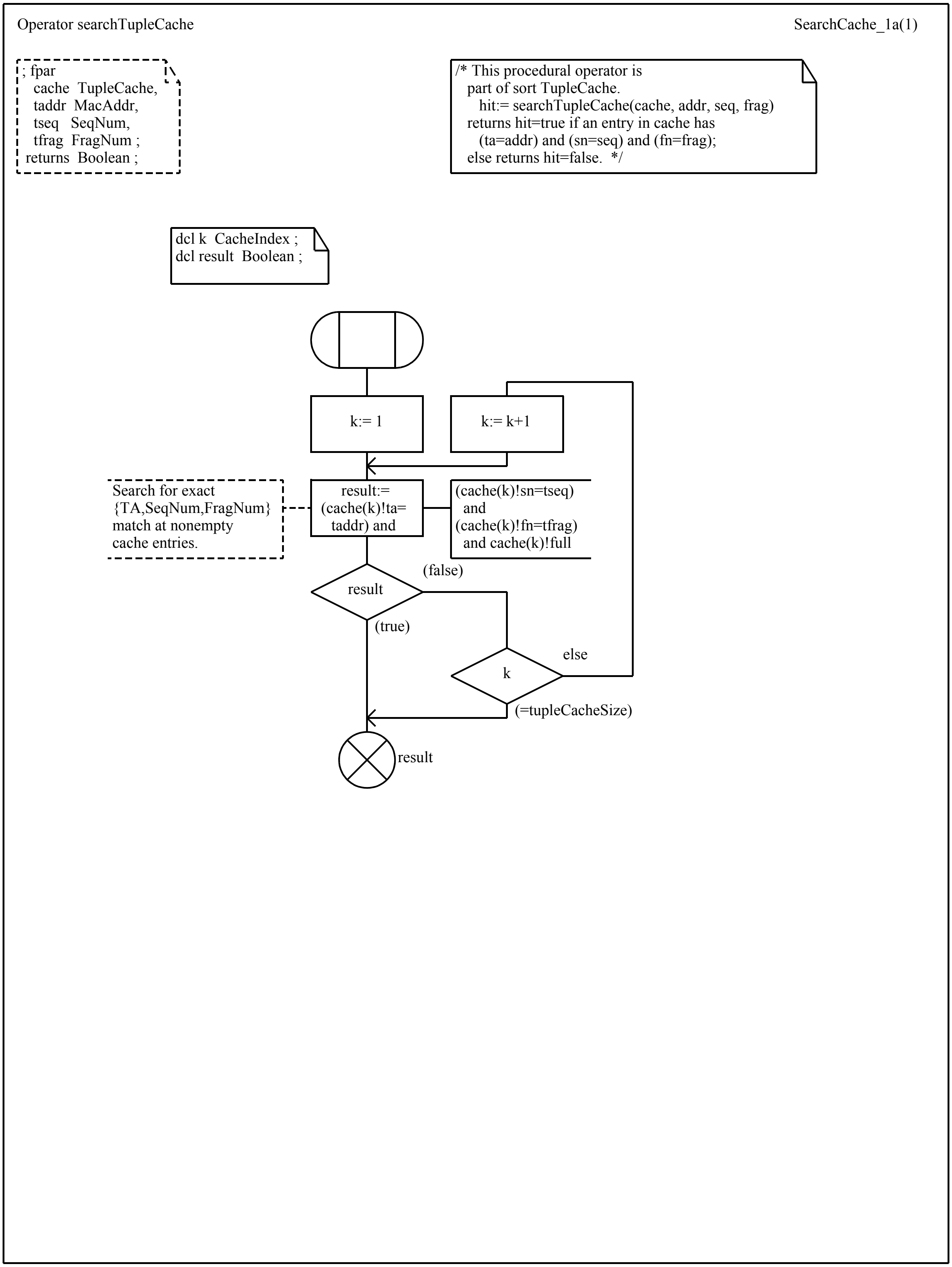
```
/******************************************************************
 *       BSS description sorts
 ******************************************************************/
    /* BssDscr is used with MlmeScan.confirm and MlmeJoin.request */
newtype BssDscr  struct
    bdBssId    MacAddr;
    bdSsId    Octetstring;    /* 1 <= length <= 32 */
    bdType    BssType;
    bdBcnPer    TU;    /* beacon period in Time Units */
    bdDtimPer    Integer;    /* DTIM period in beacon periods */
    bdTstamp    Octetstring;    /* 8 Octets from ProbeRsp/Beacon */
    bdStartTs    Octetstring;    /* 8 Octets TSF when rx Tstamp */
    bdPhyParms    PhyParms;    /* empty if not needed by PHY */
    bdCfParms    CfParms;    /* empty if not CfPollable/no PCF */
    bdIbssParms    IbssParms;    /* empty if infrastructure BSS */
    bdCap    Capability;    /* capability information */
    bdBrates    Ratestring;    /* BSS basic rate set */
endnewtype BssDscr;
newtype BssDscrSet   powerset( BssDscr)   endnewtype BssDscrSet;
```

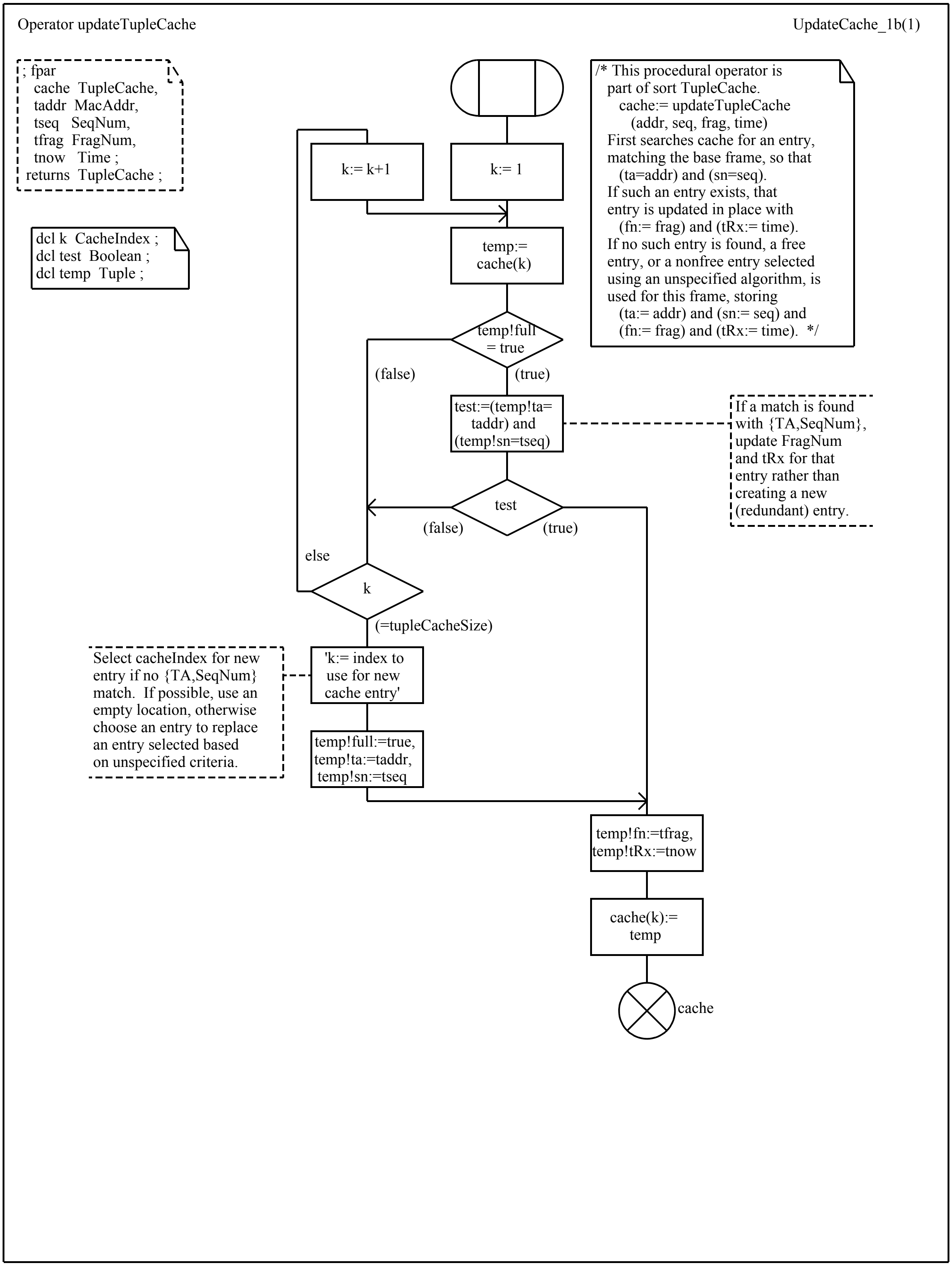


```
/*******************************************************************
 *       Duplicate filtering support sorts
 *******************************************************************/
syntype FragNum = Integer    /* Range of possible fragment numbers */
       constants 0:sMaxFragNum  endsyntype FragNum;
syntype SeqNum = Integer    /* Range of possible sequence numbers */
       constants 0:4095  endsyntype SeqNum;
newtype Tuple  struct    /* for duplicate filtering & defragmentation */
    full    Boolean;    /* =true if Tuple contains valid info */
    ta    MacAddr;    /* transmitting station address (Addr2) */
    sn    SeqNum;    /* Msdu/Mmpdu sequence number */
    fn    FragNum;    /* most recent Mpdu fragment number */
    tRx    Time;    /* reception time (endRx of fragment) */
  default (. false, nullAddr, 0, 0, 0 .);
endnewtype Tuple;
```

```
/*******************************************************************
 *       TupleCache support sorts
 *******************************************************************/
    /* Number of TupleCache entries and associated index range */
synonym tupleCacheSize Integer = 32;    /* this value is an example,
                    TupleCache size is implementation dependent */
syntype CacheIndex = Integer  constants 1:tupleCacheSize
  endsyntype CacheIndex;
/* TupleCache array */
/*    cache:= ClearTupleCache(cache)  to initialize cache */
/*    cache:= UpdateTupleCache(cache, addr, seq, frag, endRx) */
/*        if <addr,seq> is already cached, updates frag */
/*        if <addr,seq> not cached, fills an empty entry */
/*                or replaces an entry using an unspecified algorithm */
/*    SearchTupleCache(cache, addr, seq, frag) */
/*                returns true if specified <addr,seq,frag> in cache */
newtype TupleCache  Array( CacheIndex, Tuple);
  adding  operators
    ClearTupleCache  : TupleCache -> TupleCache;
    SearchTupleCache : TupleCache, MacAddr, SeqNum, FragNum -> Boolean;
    UpdateTupleCache : TupleCache, MacAddr, SeqNum, FragNum, Time ->
        TupleCache;
  operator ClearTupleCache;
    fpar cache TupleCache;  returns TupleCache;  referenced;
  operator SearchTupleCache;
    fpar cache TupleCache,  taddr MacAddr,  tseq SeqNum,  tfrag FragNum;
    returns Boolean;  referenced;
  operator UpdateTupleCache;
    fpar cache TupleCache,  taddr MacAddr,  tseq SeqNum,  tfrag FragNum,
    tnow Time;  returns TupleCache;  referenced;
endnewtype TupleCache;
```


Operator clearTupleCache
ClearCache_1a(1)
; fpar
cache TupleCache ;
returns TupleCache ;
/* This procedural operator is
part of sort TupleCache.
cache:= clearTupleCache(cache)
marks all entries in cache as empty. */
dcl k CacheIndex ;
k:= 1
k:= k+1
cache(k)!full:=
false
Mark all cache
entries as empty.
k
else
(=tupleCacheSize)
cache

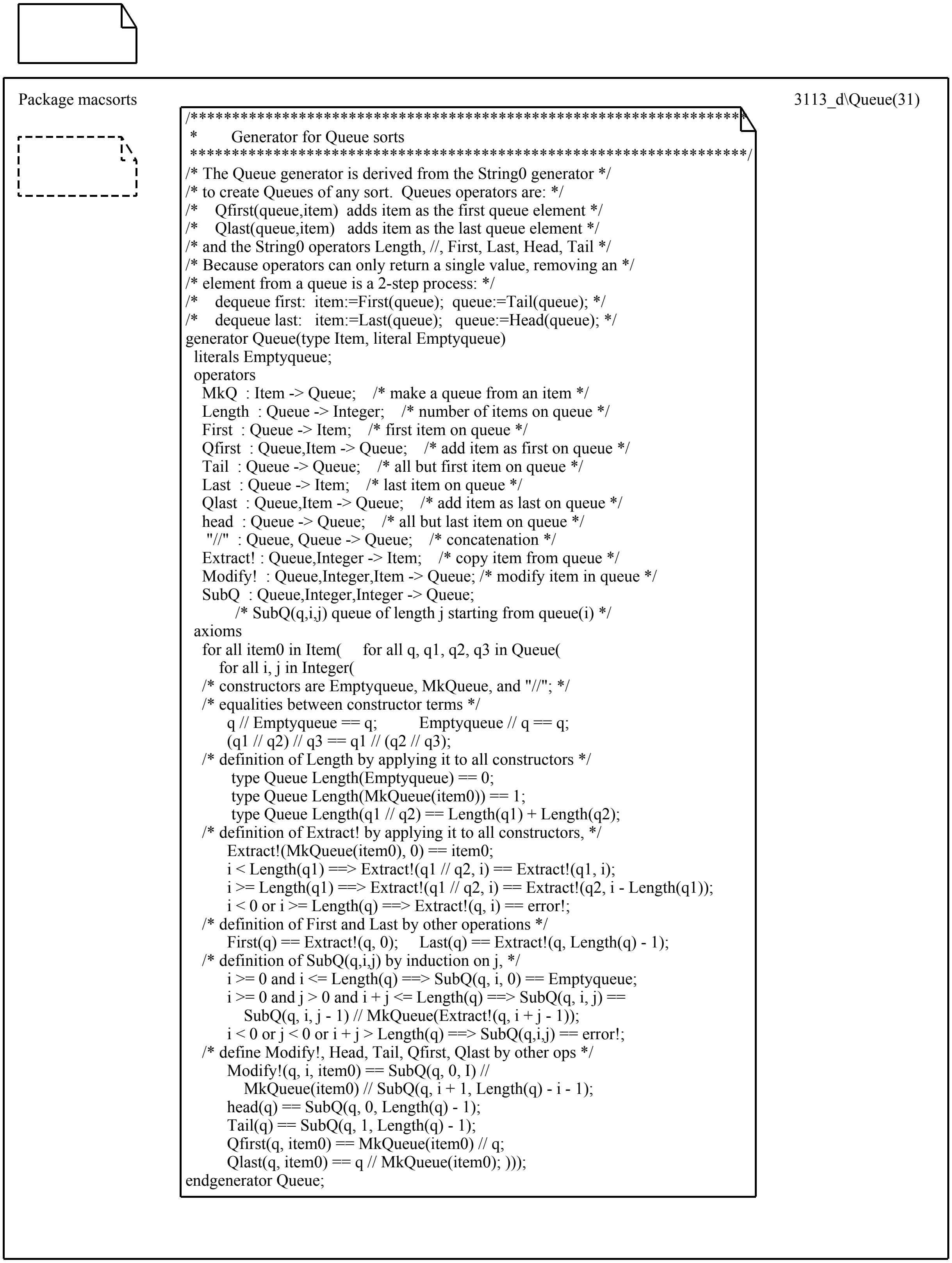

Operator searchTupleCache
SearchCache_1a(1)
; fpar
cache TupleCache,
taddr MacAddr,
tseq SeqNum,
tfrag FragNum ;
returns Boolean ;
/* This procedural operator is
part of sort TupleCache.
hit:= searchTupleCache(cache, addr, seq, frag)
returns hit=true if an entry in cache has
(ta=addr) and (sn=seq) and (fn=frag);
else returns hit=false. */
dcl k CacheIndex ;
dcl result Boolean ;
k:= 1
k:= k+1
Search for exact
{TA,SeqNum,FragNum}
match at nonempty
cache entries.
result:=
(cache(k)!ta=
taddr) and
(cache(k)!sn=tseq)
and
(cache(k)!fn=tfrag)
and cache(k)!full
result
(false)
(true)
k
else
(=tupleCacheSize)
result


Operator updateTupleCache
UpdateCache_1b(1)
; fpar
cache TupleCache,
taddr MacAddr,
tseq SeqNum,
tfrag FragNum,
tnow Time ;
returns TupleCache ;
dcl k CacheIndex ;
dcl test Boolean ;
dcl temp Tuple ;
/* This procedural operator is
part of sort TupleCache.
cache:= updateTupleCache
(addr, seq, frag, time)
First searches cache for an entry,
matching the base frame, so that
(ta=addr) and (sn=seq).
If such an entry exists, that
entry is updated in place with
(fn:= frag) and (tRx:= time).
If no such entry is found, a free
entry, or a nonfree entry selected
using an unspecified algorithm, is
used for this frame, storing
(ta:= addr) and (sn:= seq) and
(fn:= frag) and (tRx:= time). */
k:= 1
k:= k+1
temp:=
cache(k)
temp!full
= true
(false)
(true)
test:=(temp!ta=
taddr) and
(temp!sn=tseq)
If a match is found
with {TA,SeqNum},
update FragNum
and tRx for that
entry rather than
creating a new
(redundant) entry.
test
(false)
(true)
else
k
(=tupleCacheSize)
'k:= index to
use for new
cache entry'
Select cacheIndex for new
entry if no {TA,SeqNum}
match. If possible, use an
empty location, otherwise
choose an entry to replace
an entry selected based
on unspecified criteria.
temp!full:=true,
temp!ta:=taddr,
temp!sn:=tseq
temp!fn:=tfrag,
temp!tRx:=tnow
cache(k):=
temp
cache

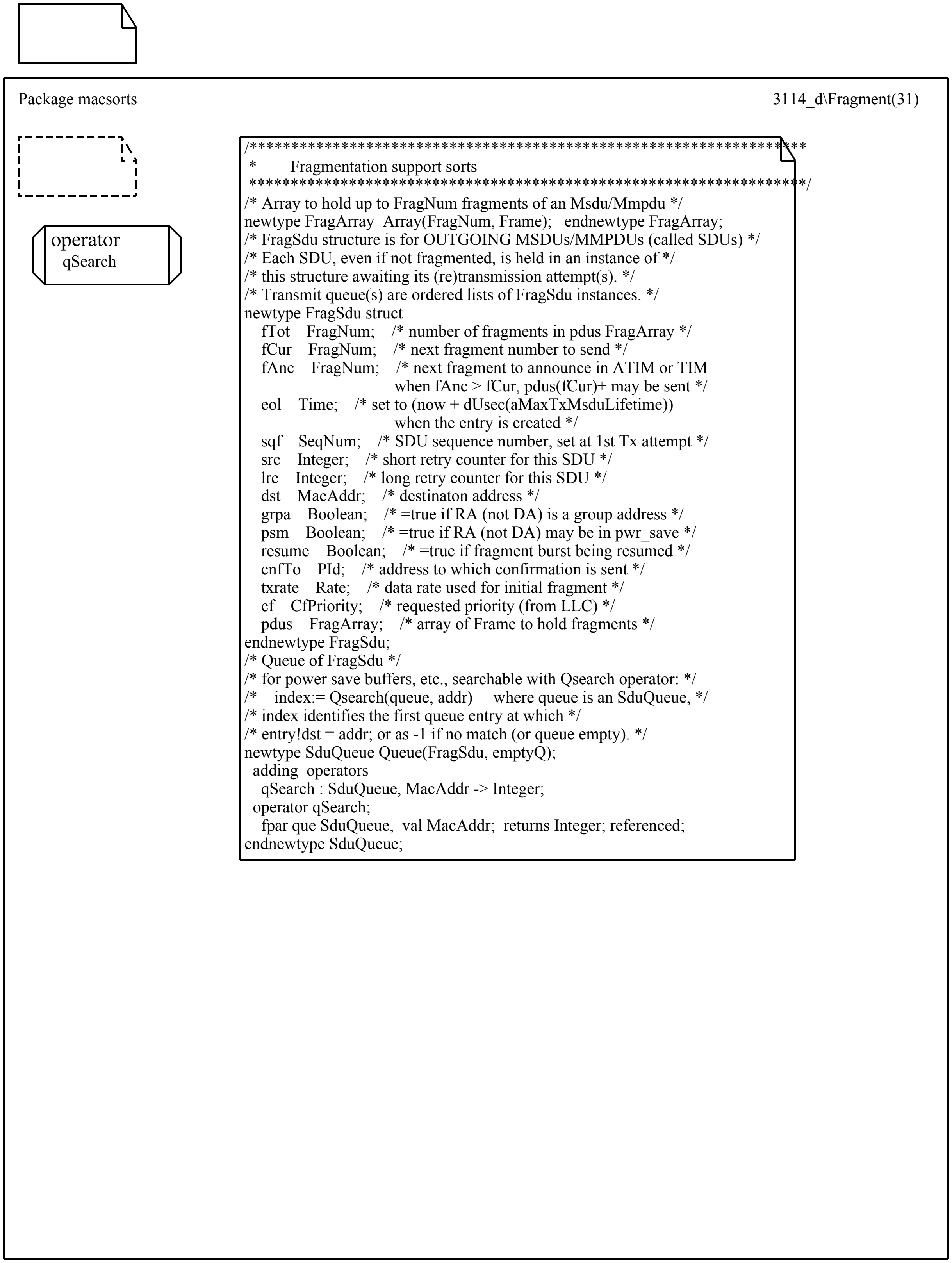
Package macsorts

3112_d\Counter(31)

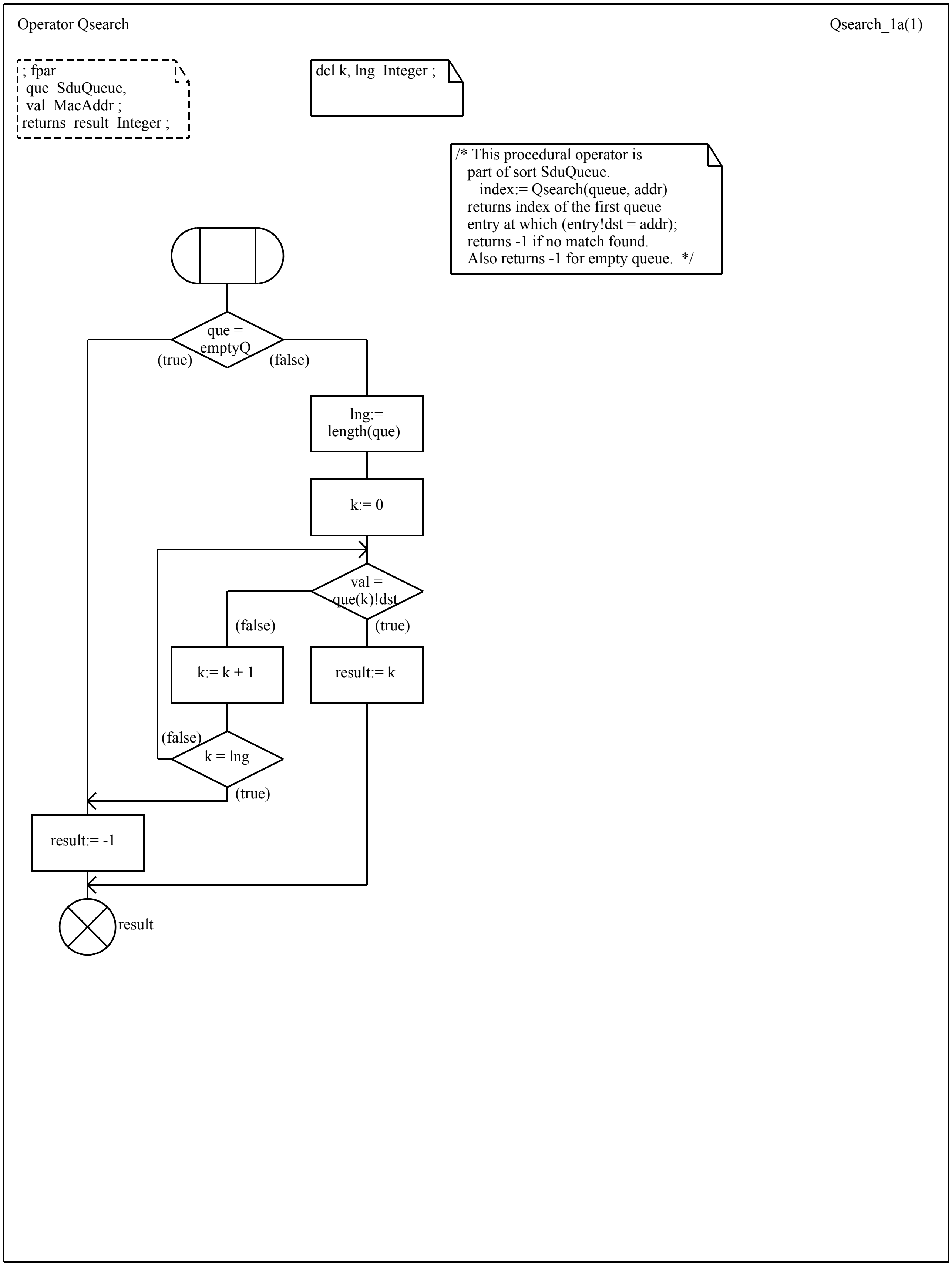
```
/*******************************************************************
 *        32-bit Counter sort and Integer string sort
 *******************************************************************/
/* This sort used for MIB counters, needed because SDL Integers */
/* have no specified maximum value.  inc(counter) increments the */
/* counter value by 1, with wraparound from (2^32)-1 to 0. */

newtype Counter32   inherits Integer   operators all;
  adding  operators
    inc : Counter32 -> Counter32;
  axioms
    for all c in Counter32 (
      c < 4294967295 ==> inc(c) == c + 1;
      c >= 4294967295 ==> inc(c) == 0; );
endnewtype Counter32;
        /* String (1-origin) of Integer */
newtype Intstring   String( Integer, noInt);   endnewtype Intstring;
```

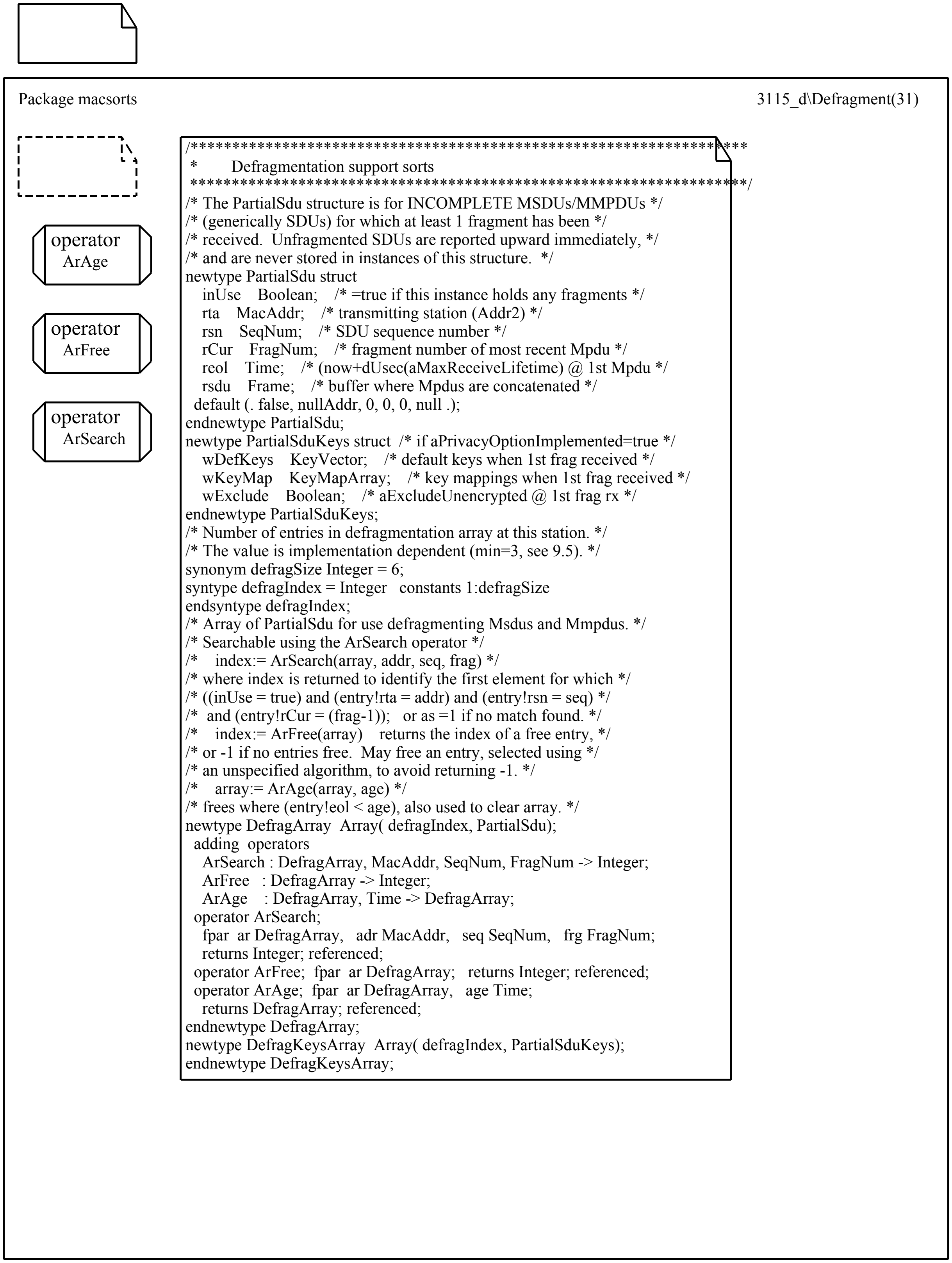


```
/*****************************************************************
 *       Generator for Queue sorts
 *****************************************************************/
/* The Queue generator is derived from the String0 generator */
/* to create Queues of any sort.  Queues operators are: */
/*    Qfirst(queue,item)  adds item as the first queue element */
/*    Qlast(queue,item)   adds item as the last queue element */
/* and the String0 operators Length, //, First, Last, Head, Tail */
/* Because operators can only return a single value, removing an */
/* element from a queue is a 2-step process: */
/*    dequeue first:  item:=First(queue);  queue:=Tail(queue); */
/*    dequeue last:   item:=Last(queue);   queue:=Head(queue); */
generator Queue(type Item, literal Emptyqueue)
  literals Emptyqueue;
  operators
    MkQ  : Item -> Queue;    /* make a queue from an item */
    Length  : Queue -> Integer;    /* number of items on queue */
    First  : Queue -> Item;    /* first item on queue */
    Qfirst  : Queue,Item -> Queue;    /* add item as first on queue */
    Tail  : Queue -> Queue;    /* all but first item on queue */
    Last  : Queue -> Item;    /* last item on queue */
    Qlast  : Queue,Item -> Queue;    /* add item as last on queue */
    head  : Queue -> Queue;    /* all but last item on queue */
     "//"  : Queue, Queue -> Queue;    /* concatenation */
    Extract! : Queue,Integer -> Item;    /* copy item from queue */
    Modify!  : Queue,Integer,Item -> Queue; /* modify item in queue */
    SubQ  : Queue,Integer,Integer -> Queue;
            /* SubQ(q,i,j) queue of length j starting from queue(i) */
  axioms
    for all item0 in Item(      for all q, q1, q2, q3 in Queue(
        for all i, j in Integer(
    /* constructors are Emptyqueue, MkQueue, and "//"; */
    /* equalities between constructor terms */
          q // Emptyqueue == q;          Emptyqueue // q == q;
          (q1 // q2) // q3 == q1 // (q2 // q3);
    /* definition of Length by applying it to all constructors */
           type Queue Length(Emptyqueue) == 0;
           type Queue Length(MkQueue(item0)) == 1;
           type Queue Length(q1 // q2) == Length(q1) + Length(q2);
    /* definition of Extract! by applying it to all constructors, */
          Extract!(MkQueue(item0), 0) == item0;
          i < Length(q1) ==> Extract!(q1 // q2, i) == Extract!(q1, i);
          i >= Length(q1) ==> Extract!(q1 // q2, i) == Extract!(q2, i - Length(q1));
          i < 0 or i >= Length(q) ==> Extract!(q, i) == error!;
    /* definition of First and Last by other operations */
          First(q) == Extract!(q, 0);     Last(q) == Extract!(q, Length(q) - 1);
    /* definition of SubQ(q,i,j) by induction on j, */
          i >= 0 and i <= Length(q) ==> SubQ(q, i, 0) == Emptyqueue;
          i >= 0 and j > 0 and i + j <= Length(q) ==> SubQ(q, i, j) ==
             SubQ(q, i, j - 1) // MkQueue(Extract!(q, i + j - 1));
          i < 0 or j < 0 or i + j > Length(q) ==> SubQ(q,i,j) == error!;
    /* define Modify!, Head, Tail, Qfirst, Qlast by other ops */
          Modify!(q, i, item0) == SubQ(q, 0, I) //
             MkQueue(item0) // SubQ(q, i + 1, Length(q) - i - 1);
          head(q) == SubQ(q, 0, Length(q) - 1);
          Tail(q) == SubQ(q, 1, Length(q) - 1);
          Qfirst(q, item0) == MkQueue(item0) // q;
          Qlast(q, item0) == q // MkQueue(item0); )));
endgenerator Queue;
```

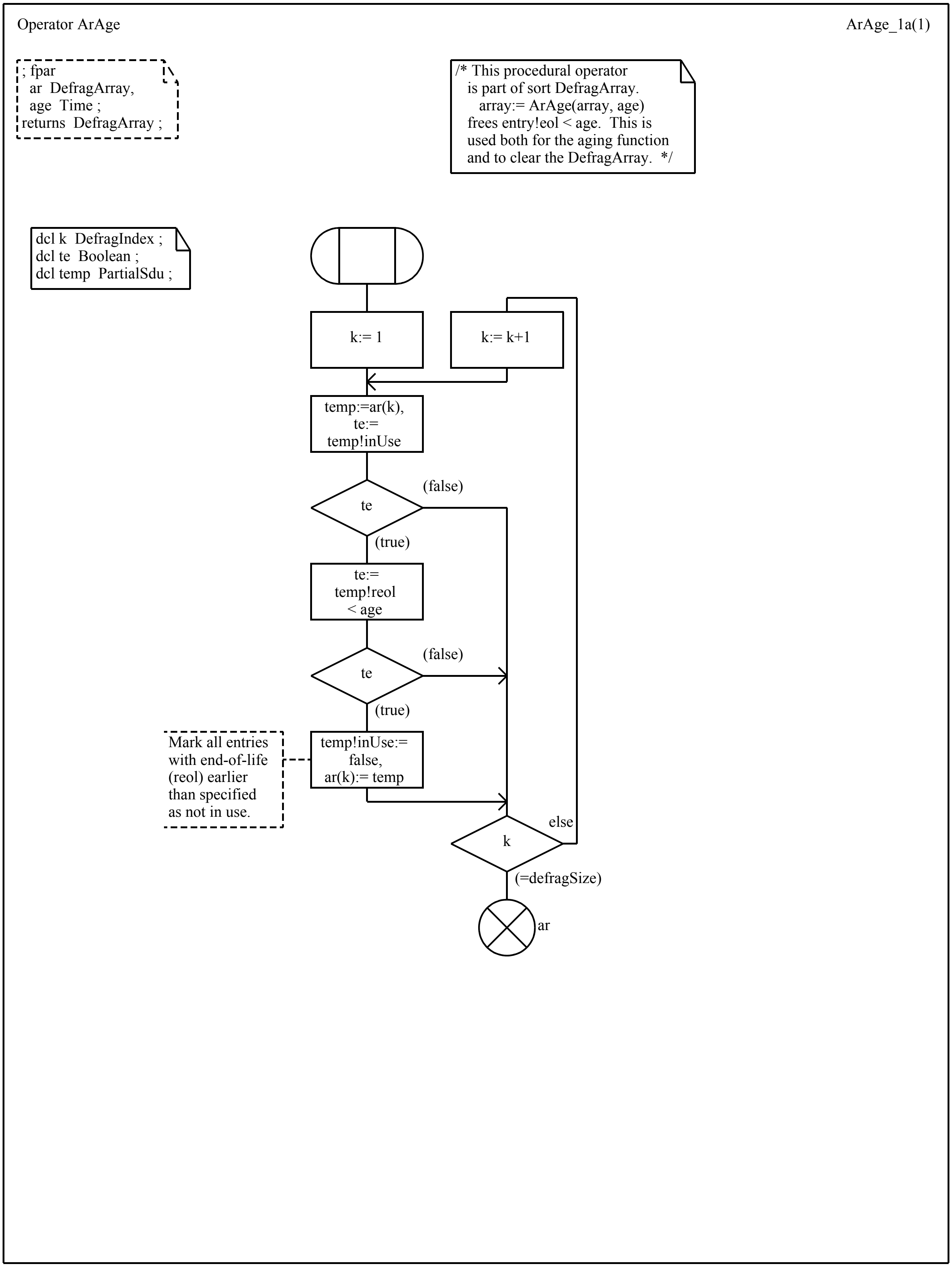


```
/******************************************************************
 *       Fragmentation support sorts
 ******************************************************************/
/* Array to hold up to FragNum fragments of an Msdu/Mmpdu */
newtype FragArray  Array(FragNum, Frame);   endnewtype FragArray;
/* FragSdu structure is for OUTGOING MSDUs/MMPDUs (called SDUs) */
/* Each SDU, even if not fragmented, is held in an instance of */
/* this structure awaiting its (re)transmission attempt(s). */
/* Transmit queue(s) are ordered lists of FragSdu instances. */
newtype FragSdu struct
   fTot    FragNum;    /* number of fragments in pdus FragArray */
   fCur    FragNum;    /* next fragment number to send */
   fAnc    FragNum;    /* next fragment to announce in ATIM or TIM
                           when fAnc > fCur, pdus(fCur)+ may be sent */
   eol    Time;    /* set to (now + dUsec(aMaxTxMsduLifetime))
                           when the entry is created */
   sqf    SeqNum;    /* SDU sequence number, set at 1st Tx attempt */
   src    Integer;    /* short retry counter for this SDU */
   lrc    Integer;    /* long retry counter for this SDU */
   dst    MacAddr;    /* destinaton address */
   grpa    Boolean;    /* =true if RA (not DA) is a group address */
   psm    Boolean;    /* =true if RA (not DA) may be in pwr_save */
   resume    Boolean;    /* =true if fragment burst being resumed */
   cnfTo    PId;    /* address to which confirmation is sent */
   txrate    Rate;    /* data rate used for initial fragment */
   cf    CfPriority;    /* requested priority (from LLC) */
   pdus    FragArray;    /* array of Frame to hold fragments */
endnewtype FragSdu;
/* Queue of FragSdu */
/* for power save buffers, etc., searchable with Qsearch operator: */
/*    index:= Qsearch(queue, addr)     where queue is an SduQueue, */
/* index identifies the first queue entry at which */
/* entry!dst = addr; or as -1 if no match (or queue empty). */
newtype SduQueue Queue(FragSdu, emptyQ);
  adding  operators
   qSearch : SduQueue, MacAddr -> Integer;
  operator qSearch;
   fpar que SduQueue,  val MacAddr;  returns Integer; referenced;
endnewtype SduQueue;
```


Operator Qsearch
Qsearch_1a(1)
; fpar
que SduQueue,
val MacAddr ;
returns result Integer ;
dcl k, lng Integer ;
/* This procedural operator is
part of sort SduQueue.
index:= Qsearch(queue, addr)
returns index of the first queue
entry at which (entry!dst = addr);
returns -1 if no match found.
Also returns -1 for empty queue. */
que =
emptyQ
(true)
(false)
lng:=
length(que)
k:= 0
val =
que(k)!dst
(false)
(true)
k:= k + 1
result:= k
(false)
k = lng
(true)
result:= -1
result



```
/*******************************************************************
 *      Defragmentation support sorts
 ******************************************************************/
/* The PartialSdu structure is for INCOMPLETE MSDUs/MMPDUs */
/* (generically SDUs) for which at least 1 fragment has been */
/* received.  Unfragmented SDUs are reported upward immediately, */
/* and are never stored in instances of this structure.  */
newtype PartialSdu struct
   inUse   Boolean;   /* =true if this instance holds any fragments */
   rta   MacAddr;   /* transmitting station (Addr2) */
   rsn   SeqNum;   /* SDU sequence number */
   rCur   FragNum;   /* fragment number of most recent Mpdu */
   reol   Time;   /* (now+dUsec(aMaxReceiveLifetime) @ 1st Mpdu */
   rsdu   Frame;   /* buffer where Mpdus are concatenated */
  default (. false, nullAddr, 0, 0, 0, null .);
endnewtype PartialSdu;
newtype PartialSduKeys struct  /* if aPrivacyOptionImplemented=true */
   wDefKeys   KeyVector;   /* default keys when 1st frag received */
   wKeyMap   KeyMapArray;   /* key mappings when 1st frag received */
   wExclude   Boolean;   /* aExcludeUnencrypted @ 1st frag rx */
endnewtype PartialSduKeys;
/* Number of entries in defragmentation array at this station. */
/* The value is implementation dependent (min=3, see 9.5). */
synonym defragSize Integer = 6;
syntype defragIndex = Integer   constants 1:defragSize
endsyntype defragIndex;
/* Array of PartialSdu for use defragmenting Msdus and Mmpdus. */
/* Searchable using the ArSearch operator */
/*    index:= ArSearch(array, addr, seq, frag) */
/* where index is returned to identify the first element for which */
/* ((inUse = true) and (entry!rta = addr) and (entry!rsn = seq) */
/*  and (entry!rCur = (frag-1));   or as =1 if no match found. */
/*    index:= ArFree(array)    returns the index of a free entry, */
/* or -1 if no entries free.  May free an entry, selected using */
/* an unspecified algorithm, to avoid returning -1. */
/*    array:= ArAge(array, age) */
/* frees where (entry!eol < age), also used to clear array. */
newtype DefragArray  Array( defragIndex, PartialSdu);
  adding  operators
   ArSearch : DefragArray, MacAddr, SeqNum, FragNum -> Integer;
   ArFree   : DefragArray -> Integer;
   ArAge    : DefragArray, Time -> DefragArray;
  operator ArSearch;
   fpar  ar DefragArray,   adr MacAddr,   seq SeqNum,   frg FragNum;
   returns Integer; referenced;
  operator ArFree;  fpar  ar DefragArray;   returns Integer; referenced;
  operator ArAge;  fpar  ar DefragArray,   age Time;
   returns DefragArray; referenced;
endnewtype DefragArray;
newtype DefragKeysArray  Array( defragIndex, PartialSduKeys);
endnewtype DefragKeysArray;
```

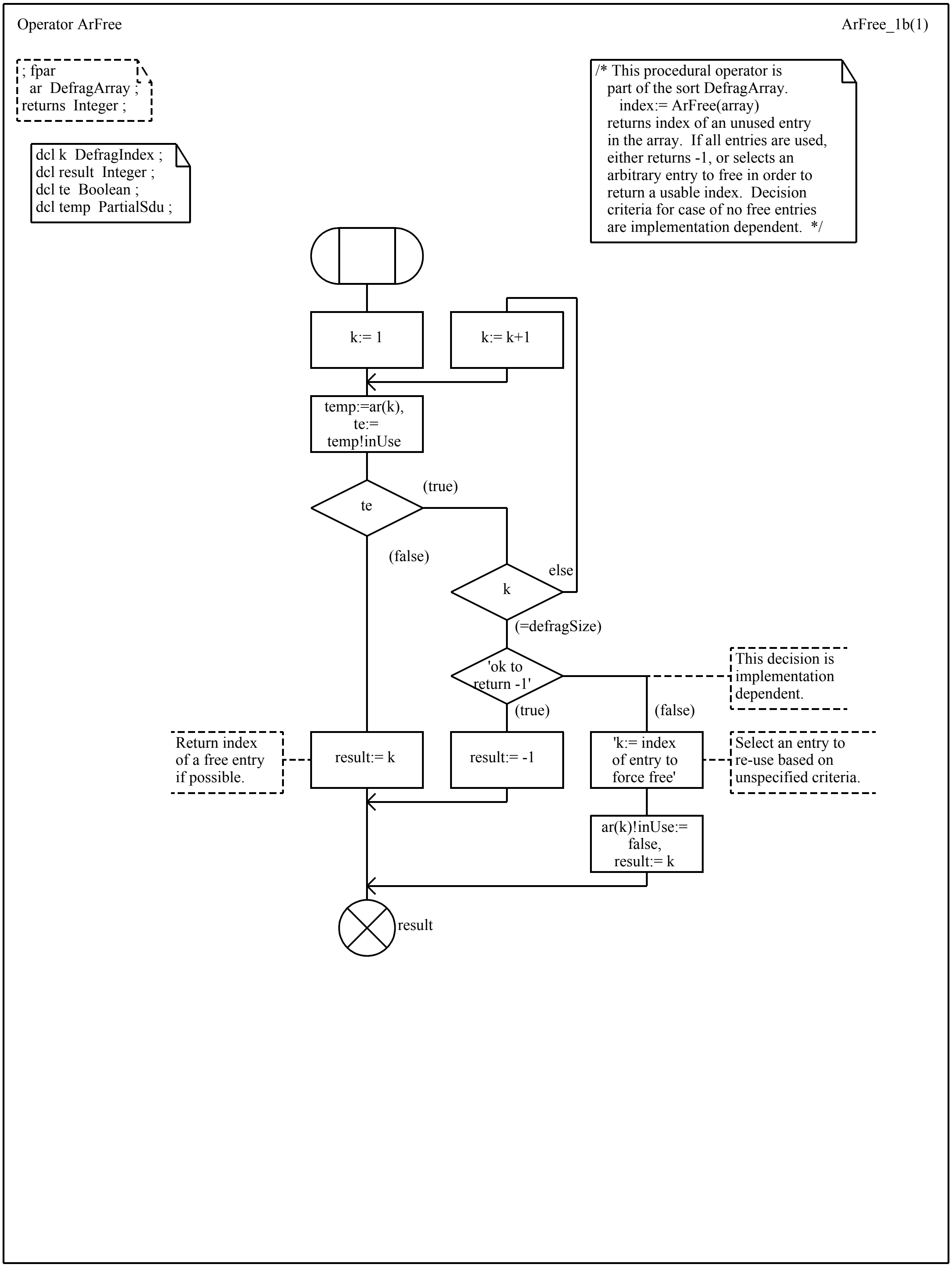


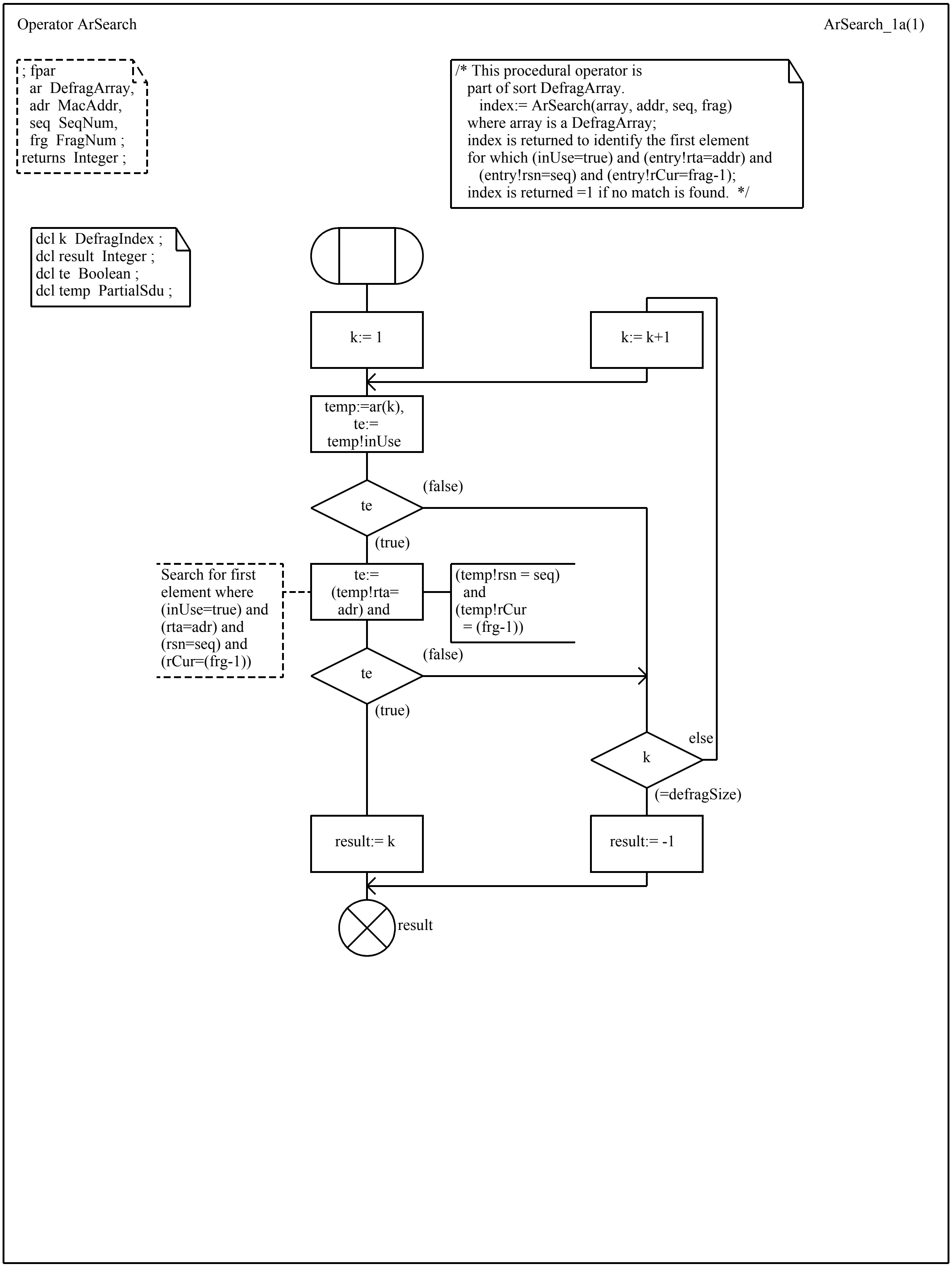
```
; fpar
  ar  DefragArray,
  age  Time ;
returns  DefragArray ;
```

```
/* This procedural operator
   is part of sort DefragArray.
     array:= ArAge(array, age)
   frees entry!eol < age.  This is
   used both for the aging function
   and to clear the DefragArray.  */
```

```
dcl k  DefragIndex ;
dcl te  Boolean ;
dcl temp  PartialSdu ;
```

Mark all entries with end-of-life (reol) earlier than specified as not in use.


Operator ArFree
ArFree_1b(1)
; fpar
ar DefragArray ;
returns Integer ;
dcl k DefragIndex ;
dcl result Integer ;
dcl te Boolean ;
dcl temp PartialSdu ;
/* This procedural operator is
part of the sort DefragArray.
index:= ArFree(array)
returns index of an unused entry
in the array. If all entries are used,
either returns -1, or selects an
arbitrary entry to free in order to
return a usable index. Decision
criteria for case of no free entries
are implementation dependent. */
k:= 1
k:= k+1
temp:=ar(k),
te:=
temp!inUse
te
(true)
(false)
k
else
(=defragSize)
'ok to
return -1'
This decision is
implementation
dependent.
(true)
(false)
Return index
of a free entry
if possible.
result:= k
result:= -1
'k:= index
of entry to
force free'
Select an entry to
re-use based on
unspecified criteria.
ar(k)!inUse:=
false,
result:= k
result


Operator ArSearch
ArSearch_1a(1)
; fpar
ar DefragArray,
adr MacAddr,
seq SeqNum,
frg FragNum ;
returns Integer ;
/* This procedural operator is
part of sort DefragArray.
index:= ArSearch(array, addr, seq, frag)
where array is a DefragArray;
index is returned to identify the first element
for which (inUse=true) and (entry!rta=addr) and
(entry!rsn=seq) and (entry!rCur=frag-1);
index is returned =1 if no match is found. */
dcl k DefragIndex ;
dcl result Integer ;
dcl te Boolean ;
dcl temp PartialSdu ;
k:= 1
k:= k+1
temp:=ar(k),
te:=
temp!inUse
te
(false)
(true)
Search for first
element where
(inUse=true) and
(rta=adr) and
(rsn=seq) and
(rCur=(frg-1))
te:=
(temp!rta=
adr) and
(temp!rsn = seq)
and
(temp!rCur
= (frg-1))
te
(false)
(true)
k
else
(=defragSize)
result:= k
result:= -1
result

Package macsorts

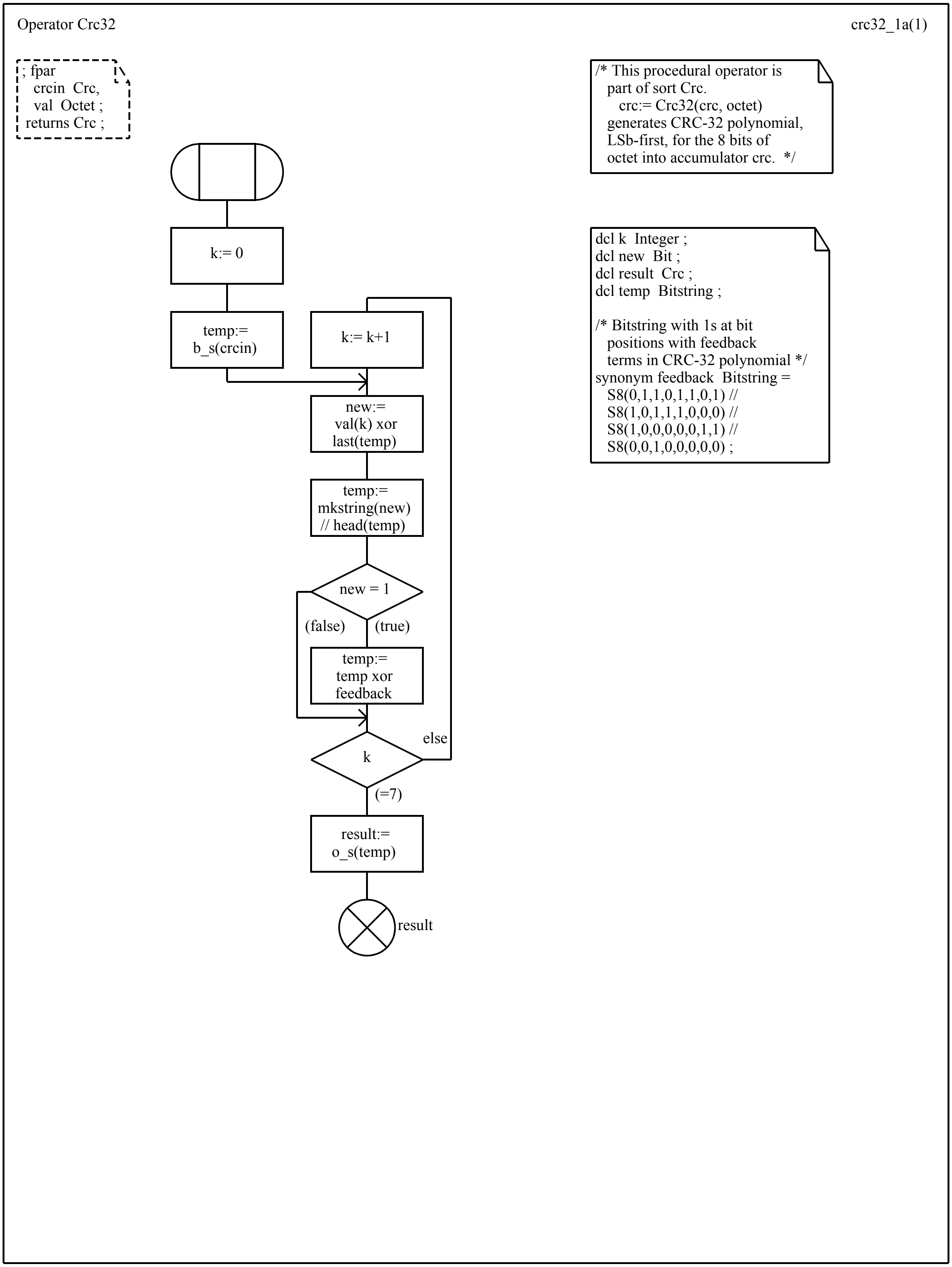
3116_d\Crc_Wep(31)

operator
crc32

```
/*****************************************************************
 *       CRC-32 sorts (for FCS and ICV)
 *****************************************************************/
/* Crc is a subtype of Octetstring with added operators: */
/*    crc:= Crc32(crc,octet) */
/* updates the crc value to include the new octet, and */
/*    Mirror(crc), which returns a Crc value with the order */
/* of the octets, and of the bits in each octet, reversed for */
/* MSb-first transmission (see 7.1.1).  Crc variables must have */
/* exactly 4 octets, which is done using initCrc or S4. */
newtype Crc   inherits Octetstring   operators all;
  adding  operators
    Crc32  : Crc, Octet -> Crc;
    mirror : Crc -> Octetstring;
  operator Crc32;  fpar crcin Crc, val Octet;  returns Crc;  referenced;
  axioms     for all c in Crc(
      mirror(c) == S4(flip(c(3)),flip(c(2)),flip(c(1)), flip(c(0))); );
endnewtype Crc;
synonym initCrc Crc =    /* Initial Crc value (all 1s) */
          << type Crc>> S4(0xFF,0xFF,0xFF,0xFF);
synonym goodCrc Crc =    /* Unique remainder for valid CRC-32 */
          << type Crc>> S4(0x7B,0xDD,0x04,0xC7);
```

operator
keyLookup

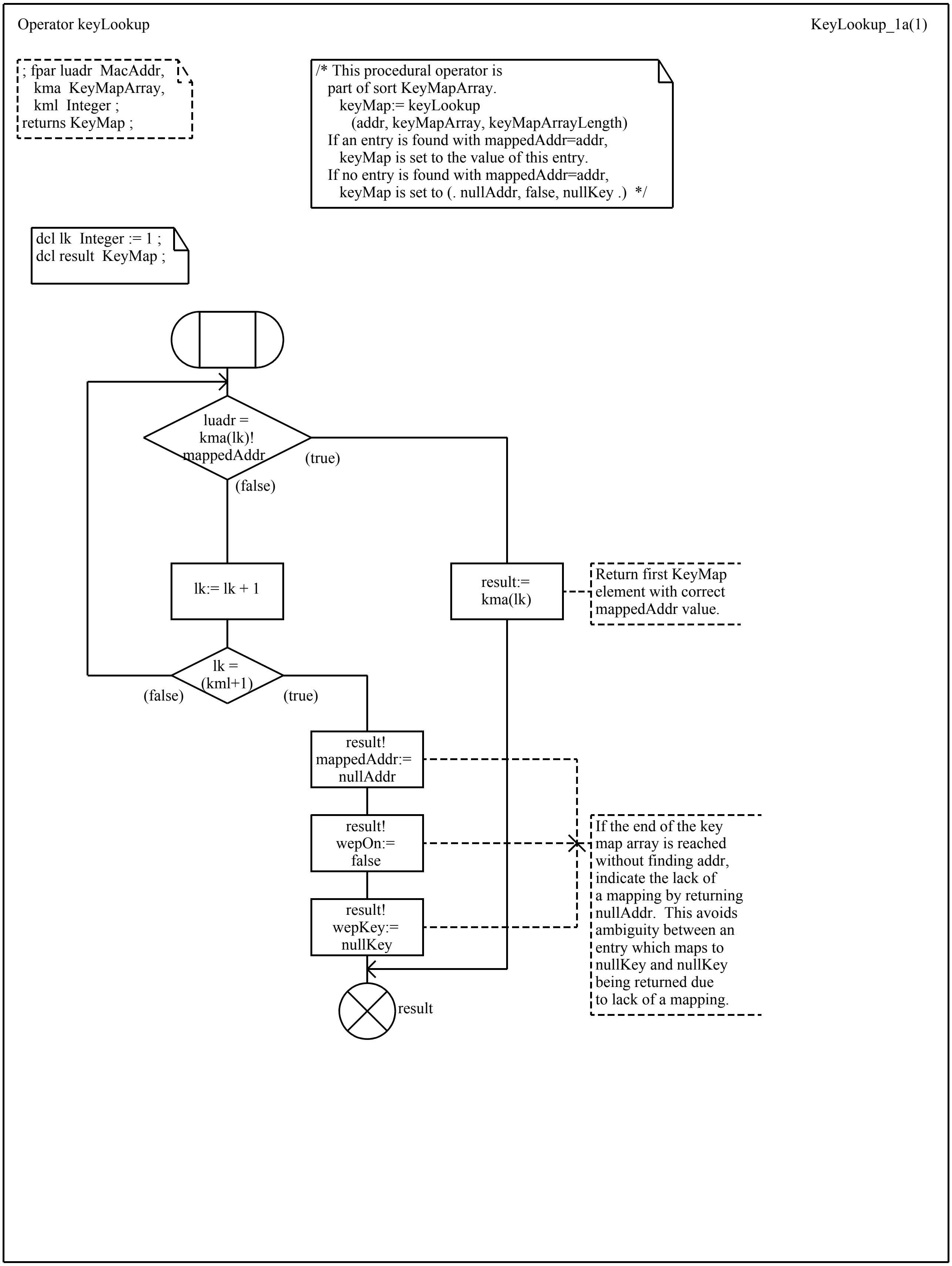
```
/*****************************************************************
 *    WEP support sorts
 *****************************************************************/
syntype KeyIndex = Integer   constants 0:3   endsyntype KeyIndex;
newtype PrngKey inherits Octetstring   operators all;
  adding literals nullKey;  /* nullKey is not any of 2^40 key values */
  axioms nullKey == null;   default nullKey;   endnewtype PrngKey;
newtype KeyVector  /* vector of default WEP keys */
  Array( KeyIndex, PrngKey);   endnewtype KeyVector;
    /* Number of entries in aWepKeyMappings array at this station.
    /* implementation dependent value, minimum=10 (see 8.3.2). */
synonym sWepKeyMappingLength Integer = 10;
syntype KeyMappingRange = Integer
    constants 1:sWepKeyMappingLength   endsyntype KeyMappingRange;
newtype KeyMap  struct    /* structure used for entries in KeyMapArray */
    mappedAddr    MacAddr;
    wepOn    Boolean;
    wepKey    PrngKey;
endnewtype KeyMap;
/* KeyMapArray -- used for aWepKeyMapping table; */
/* an array of KeyMap indexed by KeyMappingRange, with operator */
/*    KeyMap := keyLookup(addr, keyMapArray, keyMapArrayLength) */
/* returns the KeyMap entry for the specified addr, or */
/* (. nullAddr, false, nullKey .) if no mapping for addr. */
newtype KeyMapArray  Array( KeyMappingRange, KeyMap);
  adding  operators
    keyLookup : MacAddr, KeyMapArray, Integer -> KeyMap;
  operator keyLookup;
    fpar  luadr MacAddr,   kma KeyMapArray,   kml Integer;
    returns KeyMap; referenced;
endnewtype KeyMapArray;
```



```
; fpar
  crcin  Crc,
  val  Octet ;
returns Crc ;
```

```
/* This procedural operator is
   part of sort Crc.
     crc:= Crc32(crc, octet)
   generates CRC-32 polynomial,
   LSb-first, for the 8 bits of
   octet into accumulator crc.  */
```

```
dcl k  Integer ;
dcl new  Bit ;
dcl result  Crc ;
dcl temp  Bitstring ;

/* Bitstring with 1s at bit
   positions with feedback
   terms in CRC-32 polynomial */
synonym feedback  Bitstring =
   S8(0,1,1,0,1,1,0,1) //
   S8(1,0,1,1,1,0,0,0) //
   S8(1,0,0,0,0,0,1,1) //
   S8(0,0,1,0,0,0,0,0) ;
```



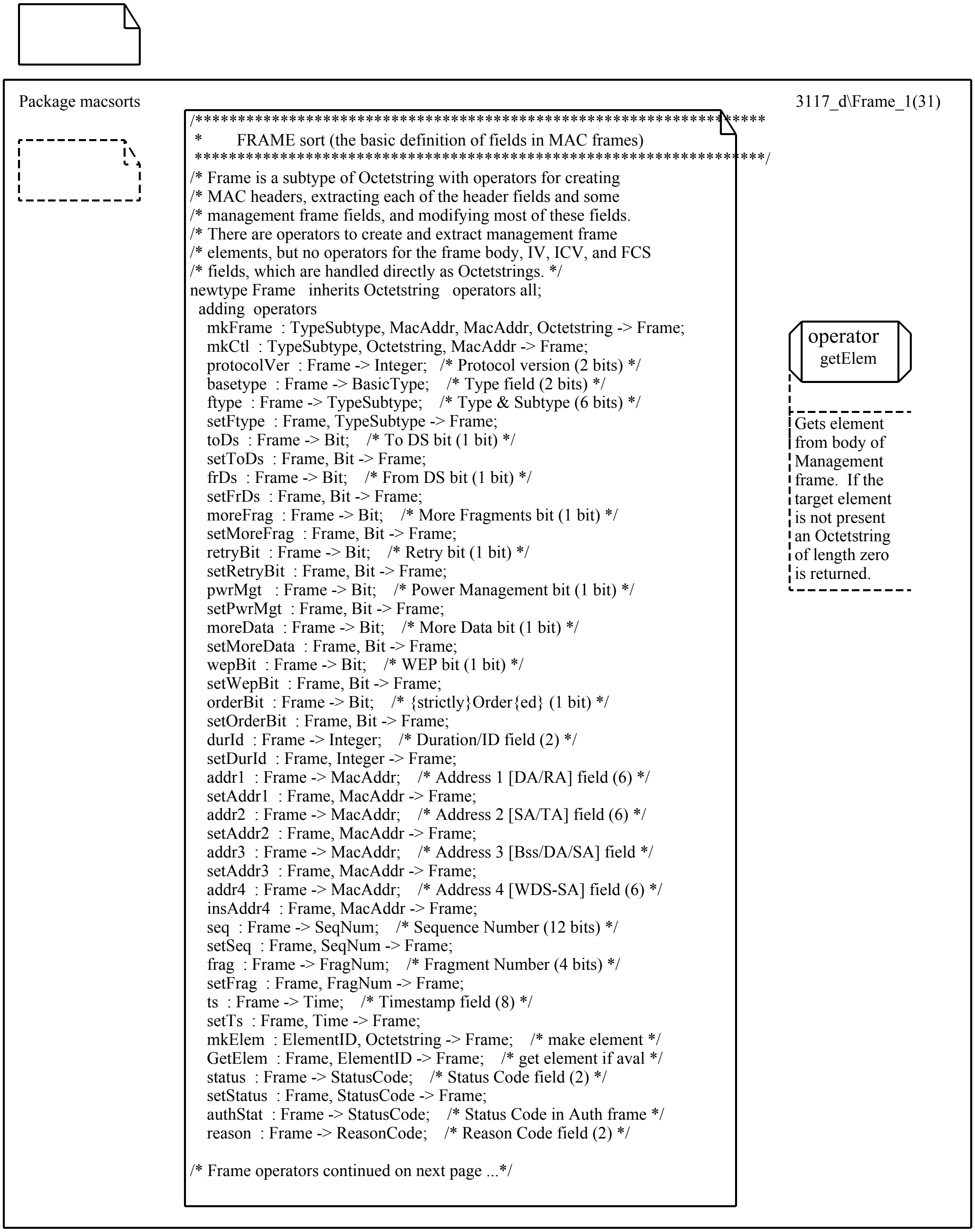
```
; fpar luadr  MacAddr,
  kma  KeyMapArray,
  kml  Integer ;
returns KeyMap ;
```

```
/* This procedural operator is
  part of sort KeyMapArray.
    keyMap:= keyLookup
      (addr, keyMapArray, keyMapArrayLength)
  If an entry is found with mappedAddr=addr,
    keyMap is set to the value of this entry.
  If no entry is found with mappedAddr=addr,
    keyMap is set to (. nullAddr, false, nullKey .)  */
```

```
dcl lk  Integer := 1 ;
dcl result  KeyMap ;
```

Return first KeyMap element with correct mappedAddr value.

If the end of the key map array is reached without finding addr, indicate the lack of a mapping by returning nullAddr. This avoids ambiguity between an entry which maps to nullKey and nullKey being returned due to lack of a mapping.

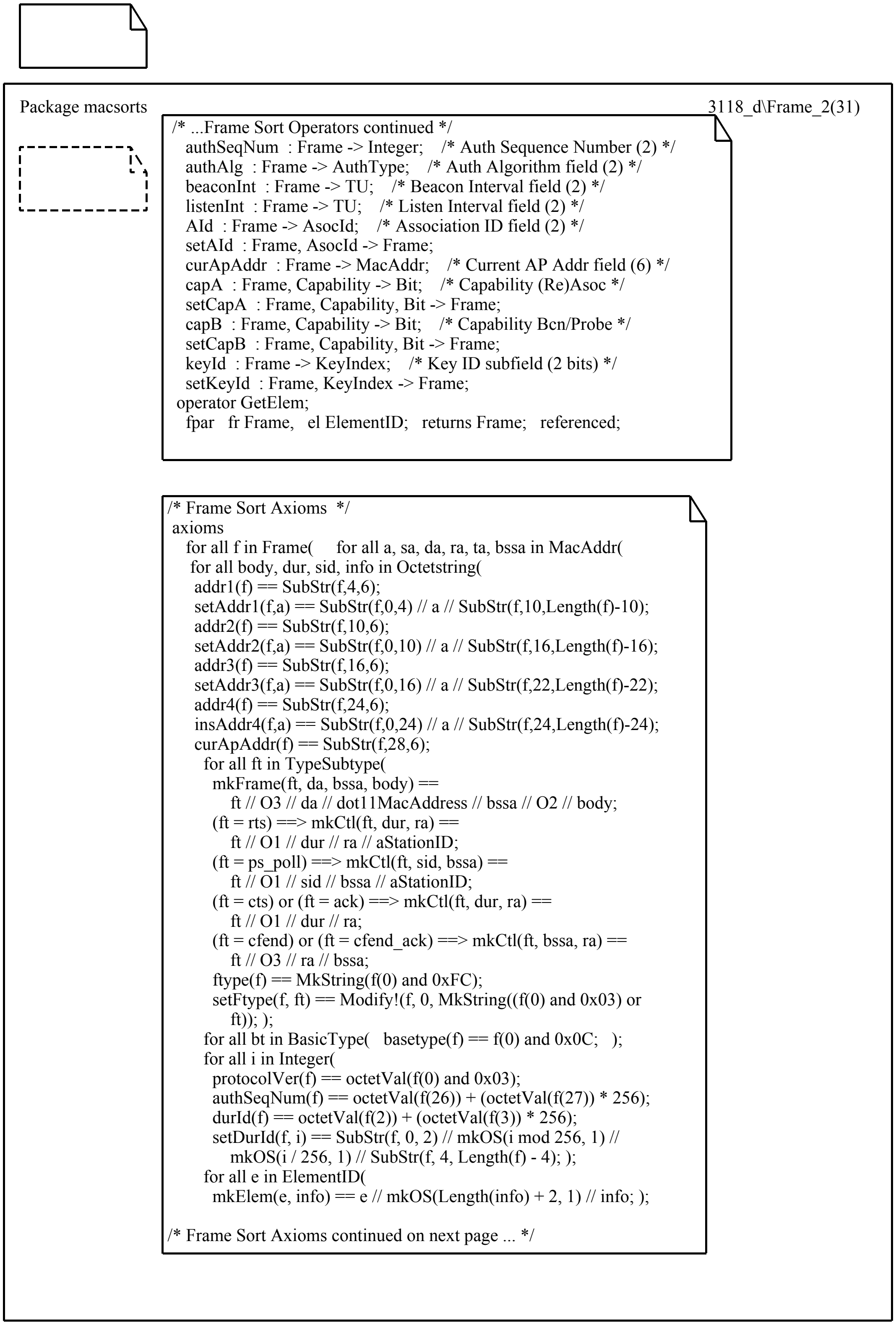


Package macsorts

3117_d\Frame_1(31)

```
/*****************************************************************
 *      FRAME sort (the basic definition of fields in MAC frames)
 *****************************************************************/
/* Frame is a subtype of Octetstring with operators for creating
/* MAC headers, extracting each of the header fields and some
/* management frame fields, and modifying most of these fields.
/* There are operators to create and extract management frame
/* elements, but no operators for the frame body, IV, ICV, and FCS
/* fields, which are handled directly as Octetstrings. */
newtype Frame   inherits Octetstring   operators all;
  adding  operators
    mkFrame  : TypeSubtype, MacAddr, MacAddr, Octetstring -> Frame;
    mkCtl  : TypeSubtype, Octetstring, MacAddr -> Frame;
    protocolVer  : Frame -> Integer;   /* Protocol version (2 bits) */
    basetype  : Frame -> BasicType;    /* Type field (2 bits) */
    ftype  : Frame -> TypeSubtype;    /* Type & Subtype (6 bits) */
    setFtype  : Frame, TypeSubtype -> Frame;
    toDs  : Frame -> Bit;    /* To DS bit (1 bit) */
    setToDs  : Frame, Bit -> Frame;
    frDs  : Frame -> Bit;    /* From DS bit (1 bit) */
    setFrDs  : Frame, Bit -> Frame;
    moreFrag  : Frame -> Bit;    /* More Fragments bit (1 bit) */
    setMoreFrag  : Frame, Bit -> Frame;
    retryBit  : Frame -> Bit;    /* Retry bit (1 bit) */
    setRetryBit  : Frame, Bit -> Frame;
    pwrMgt   : Frame -> Bit;    /* Power Management bit (1 bit) */
    setPwrMgt  : Frame, Bit -> Frame;
    moreData  : Frame -> Bit;    /* More Data bit (1 bit) */
    setMoreData  : Frame, Bit -> Frame;
    wepBit  : Frame -> Bit;    /* WEP bit (1 bit) */
    setWepBit  : Frame, Bit -> Frame;
    orderBit  : Frame -> Bit;    /* {strictly}Order{ed} (1 bit) */
    setOrderBit  : Frame, Bit -> Frame;
    durId  : Frame -> Integer;    /* Duration/ID field (2) */
    setDurId  : Frame, Integer -> Frame;
    addr1  : Frame -> MacAddr;    /* Address 1 [DA/RA] field (6) */
    setAddr1  : Frame, MacAddr -> Frame;
    addr2  : Frame -> MacAddr;    /* Address 2 [SA/TA] field (6) */
    setAddr2  : Frame, MacAddr -> Frame;
    addr3  : Frame -> MacAddr;    /* Address 3 [Bss/DA/SA] field */
    setAddr3  : Frame, MacAddr -> Frame;
    addr4  : Frame -> MacAddr;    /* Address 4 [WDS-SA] field (6) */
    insAddr4  : Frame, MacAddr -> Frame;
    seq  : Frame -> SeqNum;    /* Sequence Number (12 bits) */
    setSeq  : Frame, SeqNum -> Frame;
    frag  : Frame -> FragNum;    /* Fragment Number (4 bits) */
    setFrag  : Frame, FragNum -> Frame;
    ts  : Frame -> Time;    /* Timestamp field (8) */
    setTs  : Frame, Time -> Frame;
    mkElem  : ElementID, Octetstring -> Frame;    /* make element */
    GetElem  : Frame, ElementID -> Frame;    /* get element if aval */
    status  : Frame -> StatusCode;    /* Status Code field (2) */
    setStatus  : Frame, StatusCode -> Frame;
    authStat  : Frame -> StatusCode;    /* Status Code in Auth frame */
    reason  : Frame -> ReasonCode;    /* Reason Code field (2) */

/* Frame operators continued on next page ...*/
```

Gets element from body of Management frame. If the target element is not present an Octetstring of length zero is returned.



```
/* ...Frame Sort Operators continued */
   authSeqNum  : Frame -> Integer;    /* Auth Sequence Number (2) */
   authAlg  : Frame -> AuthType;    /* Auth Algorithm field (2) */
   beaconInt  : Frame -> TU;    /* Beacon Interval field (2) */
   listenInt  : Frame -> TU;    /* Listen Interval field (2) */
   AId  : Frame -> AsocId;    /* Association ID field (2) */
   setAId  : Frame, AsocId -> Frame;
   curApAddr  : Frame -> MacAddr;    /* Current AP Addr field (6) */
   capA  : Frame, Capability -> Bit;    /* Capability (Re)Asoc */
   setCapA  : Frame, Capability, Bit -> Frame;
   capB  : Frame, Capability -> Bit;    /* Capability Bcn/Probe */
   setCapB  : Frame, Capability, Bit -> Frame;
   keyId  : Frame -> KeyIndex;    /* Key ID subfield (2 bits) */
   setKeyId  : Frame, KeyIndex -> Frame;
 operator GetElem;
   fpar   fr Frame,   el ElementID;   returns Frame;   referenced;
```

```
/* Frame Sort Axioms  */
 axioms
   for all f in Frame(     for all a, sa, da, ra, ta, bssa in MacAddr(
    for all body, dur, sid, info in Octetstring(
     addr1(f) == SubStr(f,4,6);
     setAddr1(f,a) == SubStr(f,0,4) // a // SubStr(f,10,Length(f)-10);
     addr2(f) == SubStr(f,10,6);
     setAddr2(f,a) == SubStr(f,0,10) // a // SubStr(f,16,Length(f)-16);
     addr3(f) == SubStr(f,16,6);
     setAddr3(f,a) == SubStr(f,0,16) // a // SubStr(f,22,Length(f)-22);
     addr4(f) == SubStr(f,24,6);
     insAddr4(f,a) == SubStr(f,0,24) // a // SubStr(f,24,Length(f)-24);
     curApAddr(f) == SubStr(f,28,6);
       for all ft in TypeSubtype(
         mkFrame(ft, da, bssa, body) ==
             ft // O3 // da // dot11MacAddress // bssa // O2 // body;
         (ft = rts) ==> mkCtl(ft, dur, ra) ==
             ft // O1 // dur // ra // aStationID;
         (ft = ps_poll) ==> mkCtl(ft, sid, bssa) ==
             ft // O1 // sid // bssa // aStationID;
         (ft = cts) or (ft = ack) ==> mkCtl(ft, dur, ra) ==
             ft // O1 // dur // ra;
         (ft = cfend) or (ft = cfend_ack) ==> mkCtl(ft, bssa, ra) ==
             ft // O3 // ra // bssa;
         ftype(f) == MkString(f(0) and 0xFC);
         setFtype(f, ft) == Modify!(f, 0, MkString((f(0) and 0x03) or
             ft)); );
       for all bt in BasicType(   basetype(f) == f(0) and 0x0C;   );
       for all i in Integer(
         protocolVer(f) == octetVal(f(0) and 0x03);
         authSeqNum(f) == octetVal(f(26)) + (octetVal(f(27)) * 256);
         durId(f) == octetVal(f(2)) + (octetVal(f(3)) * 256);
         setDurId(f, i) == SubStr(f, 0, 2) // mkOS(i mod 256, 1) //
             mkOS(i / 256, 1) // SubStr(f, 4, Length(f) - 4); );
       for all e in ElementID(
         mkElem(e, info) == e // mkOS(Length(info) + 2, 1) // info; );

/* Frame Sort Axioms continued on next page ... */
```

Package macsorts 3119_d\Frame_3(31)

```
/*  ... Frame Sort  Axioms continued */
      for all b in Bit(
        toDs(f) == if (f(1) and 0x01) then 1 else 0 fi;
        setToDs(f, b) ==
            Modify!(f, 1, (f(1) and 0xFE) or S8(0,0,0,0,0,0,0,b));
        frDs(f) == if (f(1) and 0x02) then 1 else 0 fi;
        setFrDs(f, b) ==
            Modify!(f, 1, (f(1) and 0xFD) or S8(0,0,0,0,0,0,b,0));
        moreFrag(f) == if (f(1) and 0x04) then 1 else 0 fi;
        setMoreFrag(f, b) ==
            Modify!(f, 1, (f(1) and 0xFB) or S8(0,0,0,0,0,b,0,0));
        retryBit(f) == if (f(1) and 0x08) then 1 else 0 fi;
       setRetryBit(f, b) ==
            Modify!(f, 1, (f(1) and 0xF7) or S8(0,0,0,0,b,0,0,0));
        pwrMgt(f) == if (f(1) and 0x10) then 1 else 0 fi;
        setPwrMgt(f, b) ==
            Modify!(f, 1, (f(1) and 0xFB) or S8(0,0,0,b,0,0,0,0));
        moreData(f) == if (f(1) and 0x20) then 1 else 0 fi;
        setMoreData(f, b) ==
            Modify!(f, 1, (f(1) and 0xFB) or S8(0,0,b,0,0,0,0,0));
        wepBit(f) == if (f(1) and 0x40) then 1 else 0 fi;
        setWepBit(f, b) ==
            Modify!(f, 1, (f(1) and 0xFB) or S8(0,b,0,0,0,0,0,0));
        orderBit(f) == if (f(1) and 0x80) then 1 else 0 fi;
        setOrderBit(f, b) ==
            Modify!(f, 1, (f(1) and 0xFB) or S8(b,0,0,0,0,0,0,0));
        for all c in Capability(
         capA(f,c) == if (B_S(SubStr(f,24,2)) and c) then 1 else 0 fi;
         setCapA(f,c,b) == SubStr(f,0,24) // (B_S(SubStr(f,24,2) and
             (not c)) or (if b then c else O2 fi)) //
            SubStr(f,26,Length(f) - 26);
        capB(f,c) == if (B_S(SubStr(f,34,2)) and c) then 1 else 0 fi;
         setCapB(f,c,b) == SubStr(f,0,34) // (B_S(SubStr(f,34,2) and
             (not c)) or (if b then c else O2 fi)) //
            SubStr(f,36,Length(f) - 36); ));
      for all sq in SeqNum(
        seq(f) == (octetVal(f(22) and 0xF0)/16)+(octetVal(f(23)*16));
        setSeq(f, sq) == SubStr(f, 0, 22) // MkString((f(22) and 0x0F)
            or mkOctet((sq mod 16) * 16)) // mkOS(sq / 16, 1) //
            SubStr(f, 24, Length(f) - 24); );
      for all fr in FragNum(
        frag(f) == octetVal(f(22) and 0x0F);
        setFrag(f, fr) ==
            SubStr(f, 0, 22) // MkString((f(22) and 0xF0) or
            mkOctet(fr)) // SubStr(f, 23, Length(f) - 23); );
      for all tm in Time(
        ts(f) == tUsec( Usec!(octetVal(f(24)) +
            (256 * (octetVal(f(25)) +
              (256 * (octetVal(f(26)) +
                (256 * (octetVal(f(27)) +
                  (256 * (octetVal(f(28)) +
                    (256 * (octetVal(f(29)) +
                      (256 * (octetVal(f(30)) +
                        (256 * octetVal(f(31)))))))))))))) ) );

/*  Frame Sort Axioms continued on next page ... */
```

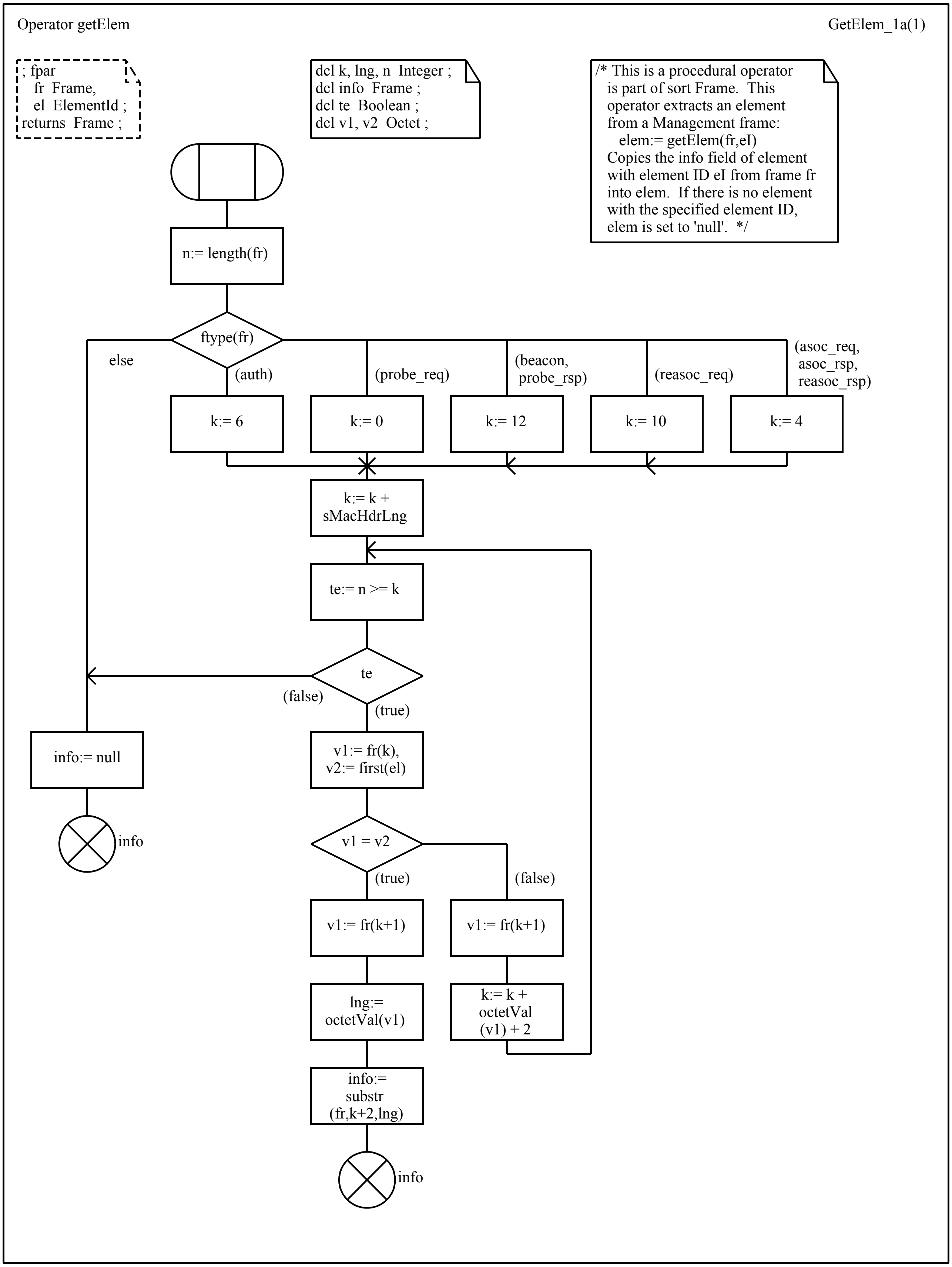
Package macsorts

3120_d\Frame_4(31)

```
/* ... Frame Sort Axioms continued */

        setTs(f, tm) == SubStr(f, 0, 24) // mkOS(fix(tm), 1) //
          mkOS((fix(tm) / 256), 1) // mkOS((fix(tm) / 65536), 1) //
          mkOS((fix(tm) / 16777216), 1) //
          mkOS((fix(tm) / 4294967296), 1) //
          mkOS(((fix(tm) / 4294967296) / 256), 1) //
          mkOS(((fix(tm) / 4294967296) / 65536), 1) //
          mkOS(((fix(tm) / 4294967296) / 16777216), 1) //
          SubStr(f, 32, Length(f) - 32); );

      for all stat in StatusCode(
        status(f) == SubStr(f, 26, 2);
        setStatus(f, stat) ==
          SubStr(f, 0, 26) // stat // SubStr(f, 28, Length(f) - 28);
        authStat(f) == SubStr(f, 28, 2); );

      for all rea in ReasonCode( reason(f) == SubStr(f, 24, 2); );
```

```
/*************************************************************************

*        ReasonCode sort

 *************************************************************************/

newtype ReasonCode  inherits Octetstring  operators all;
```

```
/*************************************************************************

*        StatusCode sort

 *************************************************************************/

newtype StatusCode  inherits Octetstring  operators all;

   adding literals successful,  unspec_fail,  unsup_cap,
     reasoc_no_asoc,  fail_other,  unsupt_alg,  auth_seq_fail,
```


Operator getElem
GetElem_1a(1)
; fpar
fr Frame,
el ElementId ;
returns Frame ;
dcl k, lng, n Integer ;
dcl info Frame ;
dcl te Boolean ;
dcl v1, v2 Octet ;
/* This is a procedural operator
is part of sort Frame. This
operator extracts an element
from a Management frame:
elem:= getElem(fr,eI)
Copies the info field of element
with element ID eI from frame fr
into elem. If there is no element
with the specified element ID,
elem is set to 'null'. */
n:= length(fr)
ftype(fr)
else
(auth)
(probe_req)
(beacon,
probe_rsp)
(reasoc_req)
(asoc_req,
asoc_rsp,
reasoc_rsp)
k:= 6
k:= 0
k:= 12
k:= 10
k:= 4
k:= k +
sMacHdrLng
te:= n >= k
te
(false)
(true)
info:= null
info
v1:= fr(k),
v2:= first(el)
v1 = v2
(true)
(false)
v1:= fr(k+1)
v1:= fr(k+1)
lng:=
octetVal(v1)
k:= k +
octetVal
(v1) + 2
info:=
substr
(fr,k+2,lng)
info

Package macsorts

3121_d\FrameType(31)

```
/*****************************************************************
 *       Frame Type sorts
 *****************************************************************/
/* TypeSubtype defines the full, 6-bit frame type identifiers. */
/* These values are useful with ftype operator of Frame sort. */
newtype TypeSubtype   inherits Octetstring   operators all;
  adding  literals  asoc_req,  asoc_rsp,  reasoc_req,  reasoc_rsp,
    probe_req,  probe_rsp,  beacon,  atim,  disasoc,  auth,  deauth,
    ps_poll,  rts,  cts,  ack,  cfend,  cfend_ack,  data,  data_ack,
    data_poll,  data_poll_ack,  null_frame,  cfack,  cfpoll,  cfpoll_ack;
  axioms
    asoc_req == MkString(S8(0,0,0,0,0,0,0,0));
    asoc_rsp == MkString(S8(0,0,0,0,1,0,0,0));
    reasoc_req == MkString(S8(0,0,0,0,0,1,0,0));
    reasoc_rsp == MkString(S8(0,0,0,0,1,1,0,0));
    probe_req == MkString(S8(0,0,0,0,0,0,1,0));
    probe_rsp == MkString(S8(0,0,0,0,1,0,1,0));
    beacon == MkString(S8(0,0,0,0,0,0,0,1));
    atim == MkString(S8(0,0,0,0,1,0,0,1));
    disasoc == MkString(S8(0,0,0,0,0,1,0,1));
    auth == MkString(S8(0,0,0,0,1,1,0,1));
    deauth == MkString(S8(0,0,0,0,0,0,1,1));
    ps_poll == MkString(S8(0,0,1,0,0,1,0,1));
    rts == MkString(S8(0,0,1,0,1,1,0,1));
    cts == MkString(S8(0,0,1,0,0,0,1,1));
    ack == MkString(S8(0,0,1,0,1,0,1,1));
    cfend == MkString(S8(0,0,1,0,0,1,1,1));
    cfend_ack == MkString(S8(0,0,1,0,1,1,1,1));
    data == MkString(S8(0,0,0,1,0,0,0,0));
    data_ack == MkString(S8(0,0,0,1,1,0,0,0));
    data_poll == MkString(S8(0,0,0,1,0,1,0,0));
    data_poll_ack == MkString(S8(0,0,0,1,1,1,0,0));
    null_frame == MkString(S8(0,0,0,1,0,0,1,0));
    cfack == MkString(S8(0,0,0,1,1,0,1,0));
    cfpoll == MkString(S8(0,0,0,1,0,1,1,0));
    cfpoll_ack == MkString(S8(0,0,0,1,1,1,1,0));
endnewtype TypeSubtype;
/* BasicTypes defines the 2-bit frame type groups */
newtype BasicType   inherits Bitstring   operators all;
  adding  literals   control,   data,   management,   reserved;
  axioms
    control == S8(0,0,1,0,0,0,0,0);    data == S8(0,0,0,1,0,0,0,0);
    management == S8(0,0,0,0,0,0,0,0);    reserved == S8(0,0,1,1,0,0,0,0);
endnewtype BasicType;
```

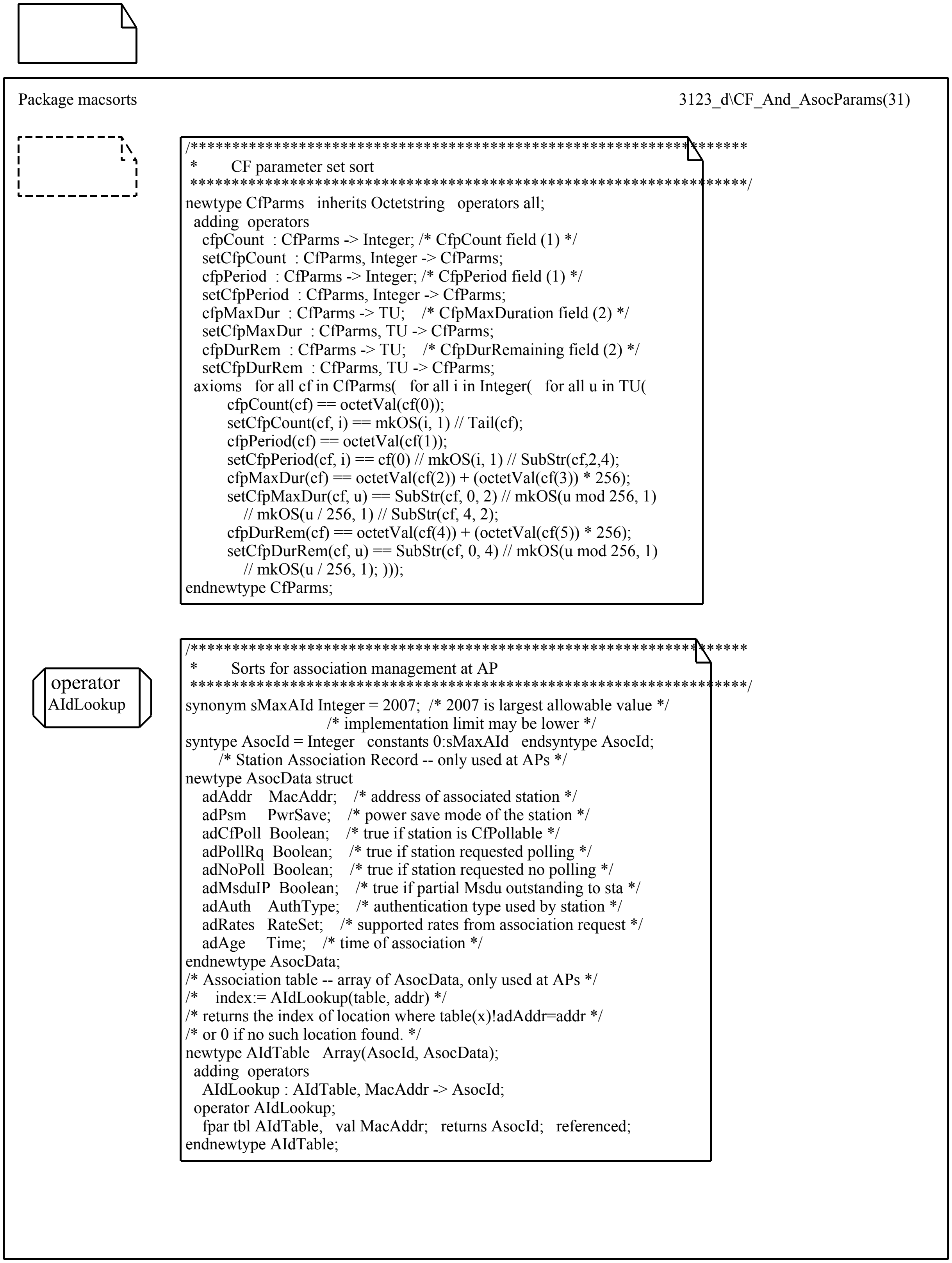
Package macsorts

3122_d\MgmtFields(31)

```
/****************************************************************************

*        ElementID sort

 ****************************************************************************/

newtype ElementID  inherits Octetstring  operators all;
    adding literals eSsId,  eSupRates,  eFhParms,  eDsParms,
       eCfParms,  eTim,  eIbParms,  eCtext,  eERP,  eExtSupRates;

    axioms
```

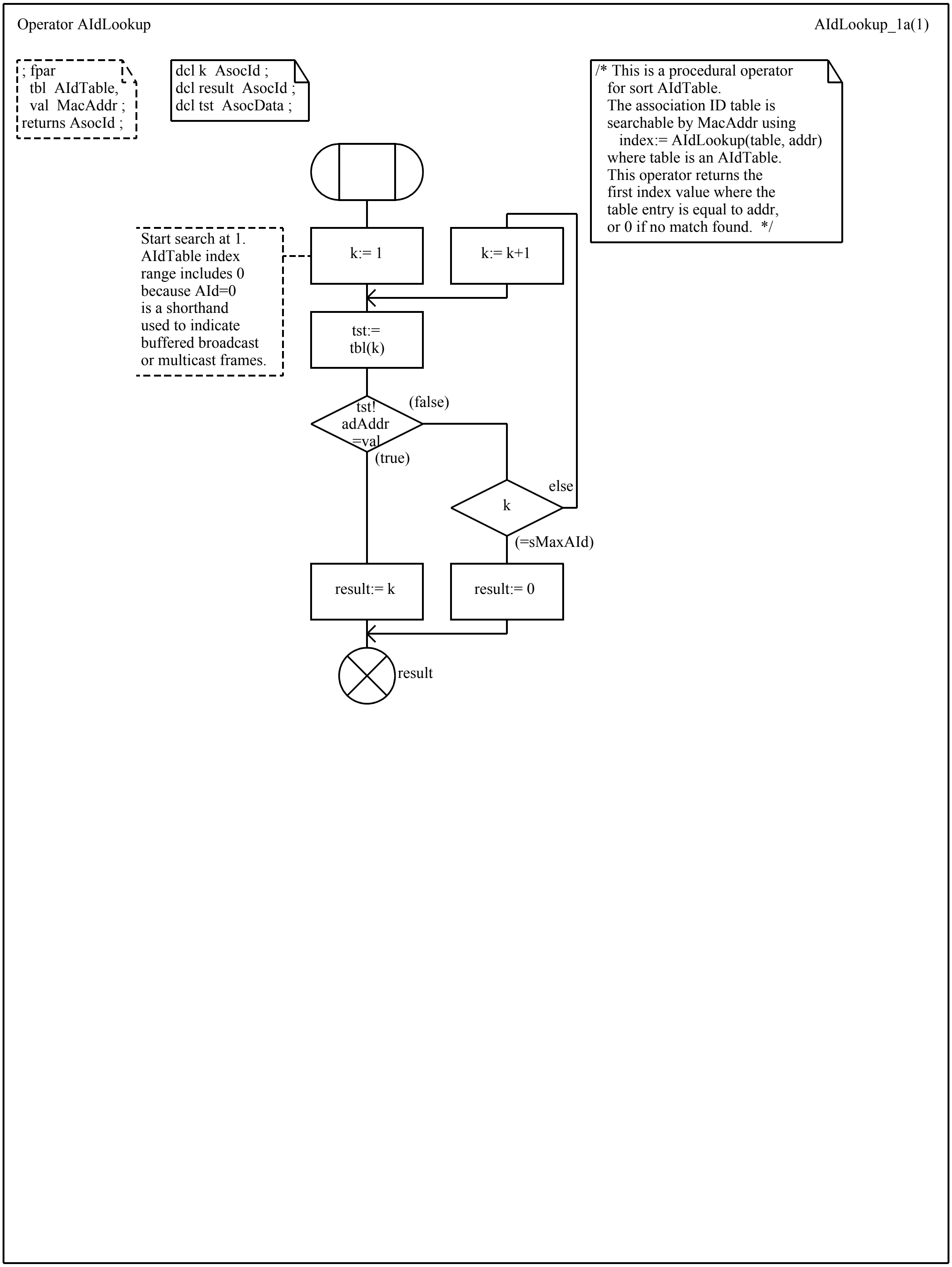
```
/****************************************************************************

*        Capability field bit assignments sort

 ****************************************************************************/

newtype Capability inherits Bitstring operators all;
    adding literals cEss,  cIbss, cPollable,  cPollReq, cPrivacy,  cShortPreamble,
       cPBCC,  cChannelAgility,  cShortSlot,  cDsssOfdm;
```

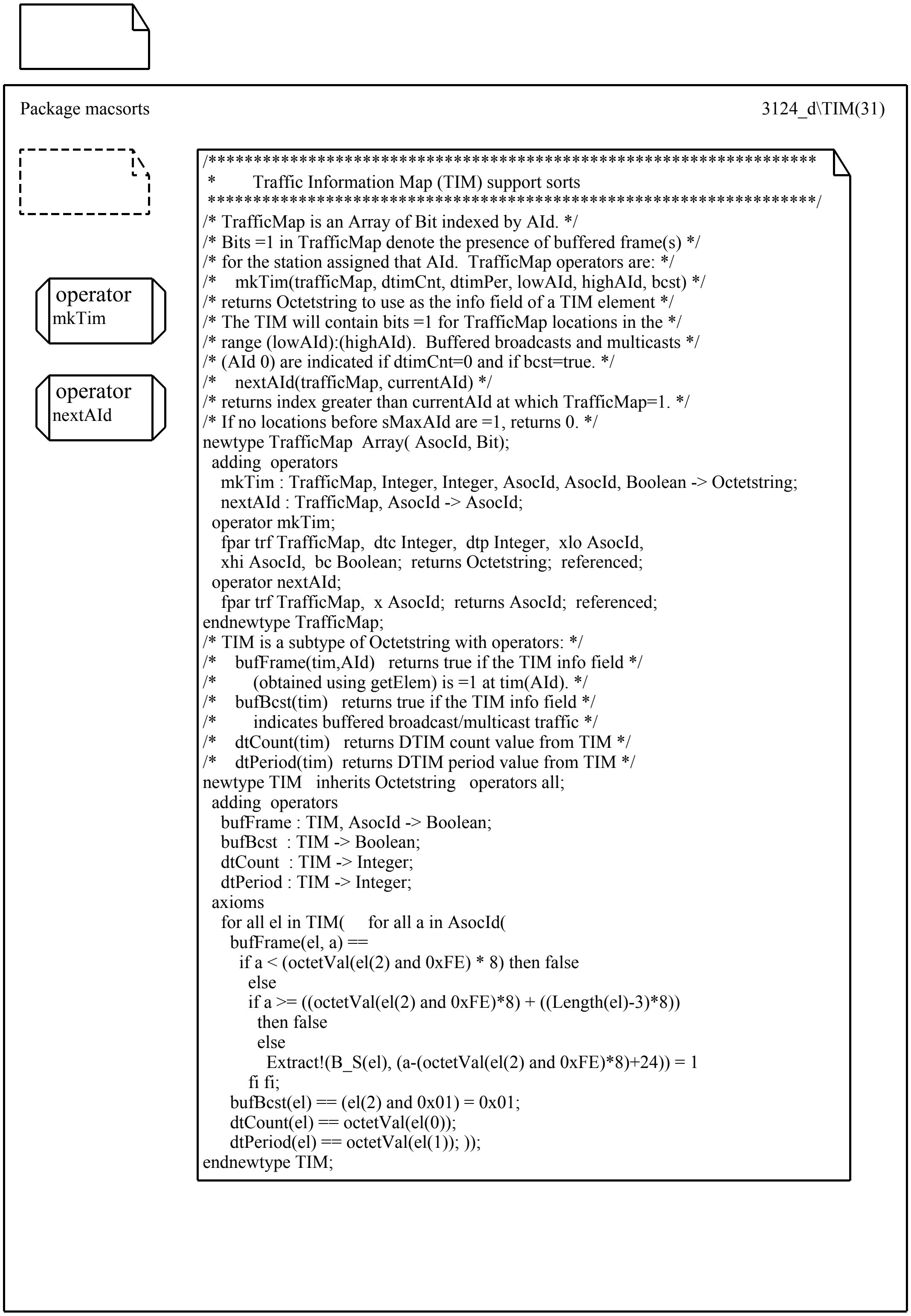
```
/****************************************************************************

*        IBSS parameter set sort

 ****************************************************************************/
```



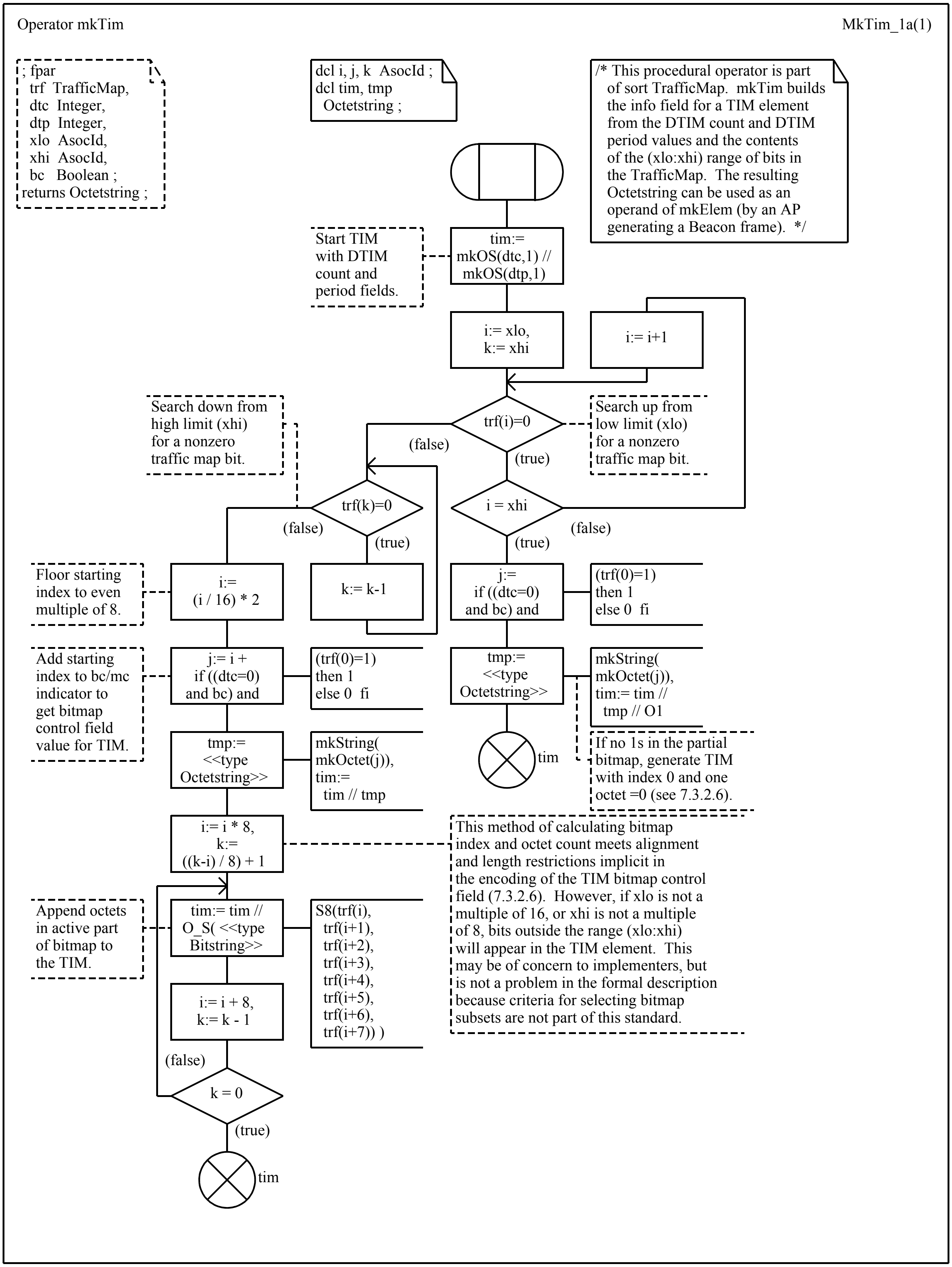
```
/*****************************************************************
 *      CF parameter set sort
 *****************************************************************/
newtype CfParms   inherits Octetstring   operators all;
  adding  operators
    cfpCount  : CfParms -> Integer; /* CfpCount field (1) */
    setCfpCount  : CfParms, Integer -> CfParms;
    cfpPeriod  : CfParms -> Integer; /* CfpPeriod field (1) */
    setCfpPeriod  : CfParms, Integer -> CfParms;
    cfpMaxDur  : CfParms -> TU;    /* CfpMaxDuration field (2) */
    setCfpMaxDur  : CfParms, TU -> CfParms;
    cfpDurRem  : CfParms -> TU;    /* CfpDurRemaining field (2) */
    setCfpDurRem  : CfParms, TU -> CfParms;
  axioms   for all cf in CfParms(   for all i in Integer(   for all u in TU(
        cfpCount(cf) == octetVal(cf(0));
        setCfpCount(cf, i) == mkOS(i, 1) // Tail(cf);
        cfpPeriod(cf) == octetVal(cf(1));
        setCfpPeriod(cf, i) == cf(0) // mkOS(i, 1) // SubStr(cf,2,4);
        cfpMaxDur(cf) == octetVal(cf(2)) + (octetVal(cf(3)) * 256);
        setCfpMaxDur(cf, u) == SubStr(cf, 0, 2) // mkOS(u mod 256, 1)
           // mkOS(u / 256, 1) // SubStr(cf, 4, 2);
        cfpDurRem(cf) == octetVal(cf(4)) + (octetVal(cf(5)) * 256);
        setCfpDurRem(cf, u) == SubStr(cf, 0, 4) // mkOS(u mod 256, 1)
           // mkOS(u / 256, 1); )));
endnewtype CfParms;
```

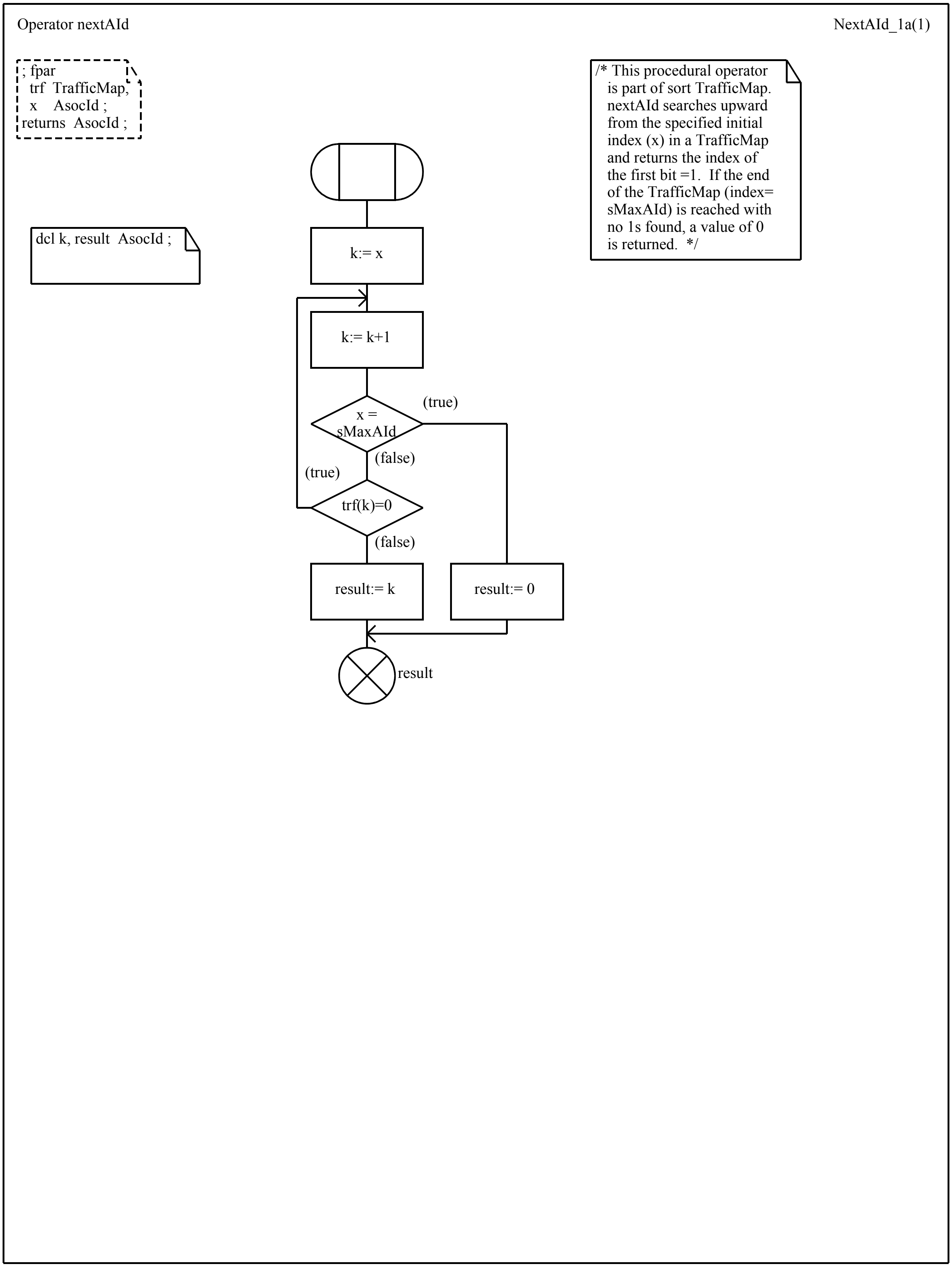
```
/*****************************************************************
 *      Sorts for association management at AP
 *****************************************************************/
synonym sMaxAId Integer = 2007;  /* 2007 is largest allowable value */
                              /* implementation limit may be lower */
syntype AsocId = Integer   constants 0:sMaxAId   endsyntype AsocId;
      /* Station Association Record -- only used at APs */
newtype AsocData struct
    adAddr    MacAddr;    /* address of associated station */
    adPsm     PwrSave;    /* power save mode of the station */
    adCfPoll  Boolean;    /* true if station is CfPollable */
    adPollRq  Boolean;    /* true if station requested polling */
    adNoPoll  Boolean;    /* true if station requested no polling */
    adMsduIP  Boolean;    /* true if partial Msdu outstanding to sta */
    adAuth    AuthType;    /* authentication type used by station */
    adRates   RateSet;    /* supported rates from association request */
    adAge     Time;    /* time of association */
endnewtype AsocData;
/* Association table -- array of AsocData, only used at APs */
/*    index:= AIdLookup(table, addr) */
/* returns the index of location where table(x)!adAddr=addr */
/* or 0 if no such location found. */
newtype AIdTable   Array(AsocId, AsocData);
  adding  operators
    AIdLookup : AIdTable, MacAddr -> AsocId;
  operator AIdLookup;
    fpar tbl AIdTable,   val MacAddr;   returns AsocId;   referenced;
endnewtype AIdTable;
```


Operator AIdLookup
AIdLookup_1a(1)
; fpar
tbl AIdTable,
val MacAddr ;
returns AsocId ;
dcl k AsocId ;
dcl result AsocId ;
dcl tst AsocData ;
/* This is a procedural operator
for sort AIdTable.
The association ID table is
searchable by MacAddr using
index:= AIdLookup(table, addr)
where table is an AIdTable.
This operator returns the
first index value where the
table entry is equal to addr,
or 0 if no match found. */
Start search at 1.
AIdTable index
range includes 0
because AId=0
is a shorthand
used to indicate
buffered broadcast
or multicast frames.
k:= 1
k:= k+1
tst:=
tbl(k)
tst!
adAddr
=val
(false)
(true)
else
k
(=sMaxAId)
result:= k
result:= 0
result



```
/*******************************************************************
 *       Traffic Information Map (TIM) support sorts
 *******************************************************************/
/* TrafficMap is an Array of Bit indexed by AId. */
/* Bits =1 in TrafficMap denote the presence of buffered frame(s) */
/* for the station assigned that AId.  TrafficMap operators are: */
/*    mkTim(trafficMap, dtimCnt, dtimPer, lowAId, highAId, bcst) */
/* returns Octetstring to use as the info field of a TIM element */
/* The TIM will contain bits =1 for TrafficMap locations in the */
/* range (lowAId):(highAId).  Buffered broadcasts and multicasts */
/* (AId 0) are indicated if dtimCnt=0 and if bcst=true. */
/*    nextAId(trafficMap, currentAId) */
/* returns index greater than currentAId at which TrafficMap=1. */
/* If no locations before sMaxAId are =1, returns 0. */
newtype TrafficMap  Array( AsocId, Bit);
  adding  operators
    mkTim : TrafficMap, Integer, Integer, AsocId, AsocId, Boolean -> Octetstring;
    nextAId : TrafficMap, AsocId -> AsocId;
  operator mkTim;
    fpar trf TrafficMap,  dtc Integer,  dtp Integer,  xlo AsocId,
    xhi AsocId,  bc Boolean;  returns Octetstring;  referenced;
  operator nextAId;
    fpar trf TrafficMap,  x AsocId;  returns AsocId;  referenced;
endnewtype TrafficMap;
/* TIM is a subtype of Octetstring with operators: */
/*    bufFrame(tim,AId)   returns true if the TIM info field */
/*        (obtained using getElem) is =1 at tim(AId). */
/*    bufBcst(tim)   returns true if the TIM info field */
/*        indicates buffered broadcast/multicast traffic */
/*    dtCount(tim)   returns DTIM count value from TIM */
/*    dtPeriod(tim)  returns DTIM period value from TIM */
newtype TIM   inherits Octetstring   operators all;
  adding  operators
    bufFrame : TIM, AsocId -> Boolean;
    bufBcst  : TIM -> Boolean;
    dtCount  : TIM -> Integer;
    dtPeriod : TIM -> Integer;
  axioms
    for all el in TIM(     for all a in AsocId(
      bufFrame(el, a) ==
        if a < (octetVal(el(2) and 0xFE) * 8) then false
          else
          if a >= ((octetVal(el(2) and 0xFE)*8) + ((Length(el)-3)*8))
            then false
            else
              Extract!(B_S(el), (a-(octetVal(el(2) and 0xFE)*8)+24)) = 1
          fi fi;
      bufBcst(el) == (el(2) and 0x01) = 0x01;
      dtCount(el) == octetVal(el(0));
      dtPeriod(el) == octetVal(el(1)); ));
endnewtype TIM;
```


Operator mkTim
MkTim_1a(1)
; fpar
trf TrafficMap,
dtc Integer,
dtp Integer,
xlo AsocId,
xhi AsocId,
bc Boolean ;
returns Octetstring ;
dcl i, j, k AsocId ;
dcl tim, tmp
Octetstring ;
/* This procedural operator is part of sort TrafficMap. mkTim builds the info field for a TIM element from the DTIM count and DTIM period values and the contents of the (xlo:xhi) range of bits in the TrafficMap. The resulting Octetstring can be used as an operand of mkElem (by an AP generating a Beacon frame). */
Start TIM with DTIM count and period fields.
tim:= mkOS(dtc,1) // mkOS(dtp,1)
i:= xlo, k:= xhi
i:= i+1
Search down from high limit (xhi) for a nonzero traffic map bit.
trf(i)=0
(false)
(true)
Search up from low limit (xlo) for a nonzero traffic map bit.
trf(k)=0
(false)
(true)
i = xhi
(false)
(true)
Floor starting index to even multiple of 8.
i:= (i / 16) * 2
k:= k-1
j:= if ((dtc=0) and bc) and
(trf(0)=1) then 1 else 0 fi
Add starting index to bc/mc indicator to get bitmap control field value for TIM.
j:= i + if ((dtc=0) and bc) and
(trf(0)=1) then 1 else 0 fi
tmp:= <<type Octetstring>>
mkString(mkOctet(j)), tim:= tim // tmp // O1
tim
If no 1s in the partial bitmap, generate TIM with index 0 and one octet =0 (see 7.3.2.6).
tmp:= <<type Octetstring>>
mkString(mkOctet(j)), tim:= tim // tmp
i:= i * 8, k:= ((k-i) / 8) + 1
This method of calculating bitmap index and octet count meets alignment and length restrictions implicit in the encoding of the TIM bitmap control field (7.3.2.6). However, if xlo is not a multiple of 16, or xhi is not a multiple of 8, bits outside the range (xlo:xhi) will appear in the TIM element. This may be of concern to implementers, but is not a problem in the formal description because criteria for selecting bitmap subsets are not part of this standard.
Append octets in active part of bitmap to the TIM.
tim:= tim // O_S(<<type Bitstring>>
S8(trf(i), trf(i+1), trf(i+2), trf(i+3), trf(i+4), trf(i+5), trf(i+6), trf(i+7)))
i:= i + 8, k:= k - 1
(false)
k = 0
(true)
tim



Operator nextAId

NextAId_1a(1)

```
; fpar
 trf  TrafficMap,
 x    AsocId ;
returns  AsocId ;
```

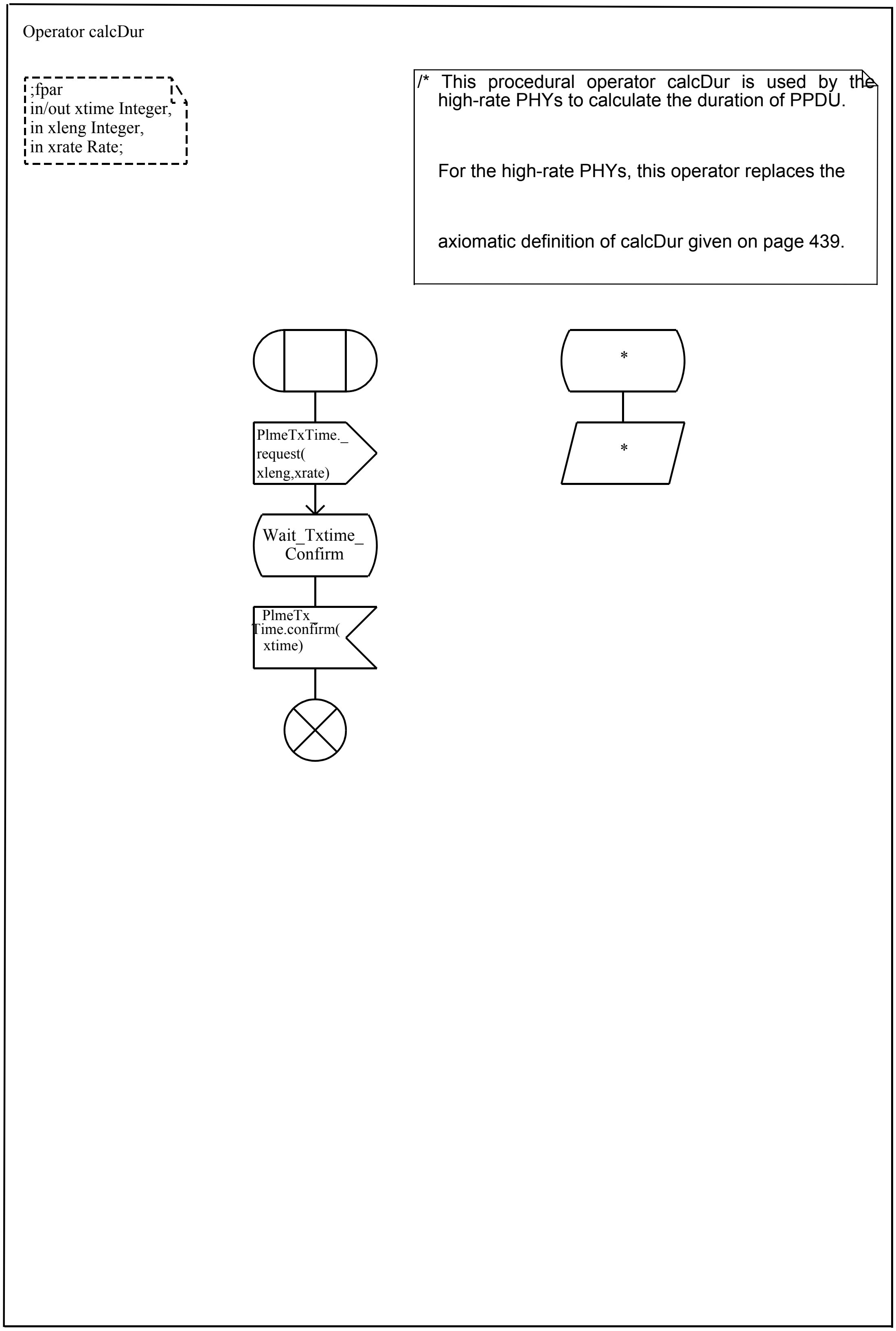
```
dcl k, result  AsocId ;
```

/* This procedural operator is part of sort TrafficMap. nextAId searches upward from the specified initial index (x) in a TrafficMap and returns the index of the first bit =1. If the end of the TrafficMap (index= sMaxAId) is reached with no 1s found, a value of 0 is returned. */

Package macsorts 3125_d\RateAndDurationSorts(31)

```
/*******************************************************************
 *       Multi-rate support sorts
 *******************************************************************/
newtype Rate   inherits Octet   operators all;
  adding  operators
    calcDur : Rate, Integer -> Integer; /* converts (rate,bitCount) to integer usec */
    rateVal : Rate -> Rate; /* clears high-order bit */
    basicRate : Rate -> Rate; /* sets high-order bit */
    isBasic : Rate -> Boolean; /* true if high-order bit set */
  axioms
    for all r in Rate(    for all i in Integer(    for all b in Boolean(
          calcDur(r, i) == ((((10000000 + (octetVal(r and 0x7F) - 1)) /
              (500 * octetVal(r and 0x7F))) * i) + 9999) / 10000;
          rateVal(r) == r and 0x7F;          basicRate(r) == r or 0x80;
          isBasic(r) == (r and 0x80) = 0x80; )));
endnewtype Rate;
syntype RateString = Octetstring   endsyntype RateString;
```

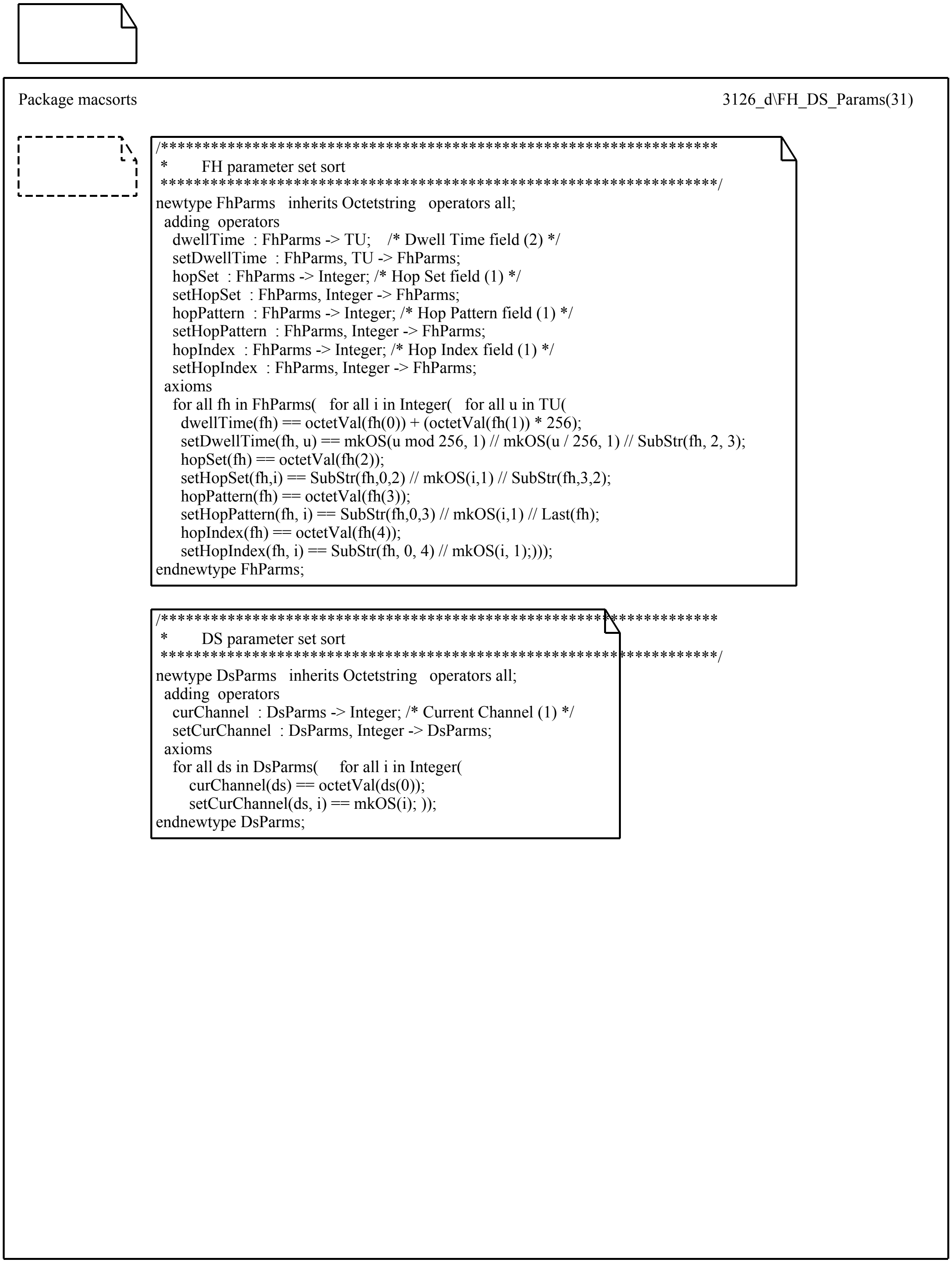
```
/*******************************************************************
 *       MPDU duration factor support sort
 *******************************************************************/
/* These operators support the encoding used to allow */
/* an Integer to represent the value of aMpduDurationFactor. */
/*    calcDF(PlcpBits, MpduBits)  returns an Integer which is */
/* the fractional part of ((PlcpBits/MpduBits)-1)*(1e9). */
/*    stuff(durFactor, MpduBits)  returns the number of PlcpBits */
/* which result from MpduBits at the specified durFactor. */
newtype DurFactor   inherits Integer   operators all;
  adding  operators
    calcDF : Integer, Integer -> DurFactor;
    stuff  : DurFactor, Integer -> Integer;
  axioms
    for all df in DurFactor(     for all mb, pb in Integer(
        calcDF(pb, mb) == ((pb * 1000000000) / mb) - 1000000000;
        stuff(df, mb) == ((mb * df) + (mb - 1)) / 1000000000; ));
endnewtype DurFactor;
```



/* This procedural operator calcDur is used by the high-rate PHYs to calculate the duration of PPDU.

For the high-rate PHYs, this operator replaces the

axiomatic definition of calcDur given on page 439.



Package macsorts

3126_d\FH_DS_Params(31)

```
/*******************************************************************
 *       FH parameter set sort
 *******************************************************************/
newtype FhParms   inherits Octetstring   operators all;
  adding  operators
    dwellTime  : FhParms -> TU;    /* Dwell Time field (2) */
    setDwellTime  : FhParms, TU -> FhParms;
    hopSet  : FhParms -> Integer; /* Hop Set field (1) */
    setHopSet  : FhParms, Integer -> FhParms;
    hopPattern  : FhParms -> Integer; /* Hop Pattern field (1) */
    setHopPattern  : FhParms, Integer -> FhParms;
    hopIndex  : FhParms -> Integer; /* Hop Index field (1) */
    setHopIndex  : FhParms, Integer -> FhParms;
  axioms
    for all fh in FhParms(   for all i in Integer(   for all u in TU(
      dwellTime(fh) == octetVal(fh(0)) + (octetVal(fh(1)) * 256);
      setDwellTime(fh, u) == mkOS(u mod 256, 1) // mkOS(u / 256, 1) // SubStr(fh, 2, 3);
      hopSet(fh) == octetVal(fh(2));
      setHopSet(fh,i) == SubStr(fh,0,2) // mkOS(i,1) // SubStr(fh,3,2);
      hopPattern(fh) == octetVal(fh(3));
      setHopPattern(fh, i) == SubStr(fh,0,3) // mkOS(i,1) // Last(fh);
      hopIndex(fh) == octetVal(fh(4));
      setHopIndex(fh, i) == SubStr(fh, 0, 4) // mkOS(i, 1);)));
endnewtype FhParms;
```

```
/*******************************************************************
 *       DS parameter set sort
 *******************************************************************/
newtype DsParms   inherits Octetstring   operators all;
  adding  operators
    curChannel  : DsParms -> Integer; /* Current Channel (1) */
    setCurChannel  : DsParms, Integer -> DsParms;
  axioms
    for all ds in DsParms(     for all i in Integer(
        curChannel(ds) == octetVal(ds(0));
        setCurChannel(ds, i) == mkOS(i); ));
endnewtype DsParms;
```

Package macsorts

3127_e\PHY_Params(31)

```
/********************************************************************
 *       Generic PHY Parameter Set sort
 ********************************************************************/
/* Generic PHY parameter element for signals related to Beacons */
/* and Probe Responses that are PHY-type independent. */
syntype PhyParms = Octetstring   endsyntype PhyParms;
```

```
NEWTYPE PhyChrstcs struct
  aSlotTime Usec;
  aSifsTime Usec;
  aCCATime Usec;
  aRxTxTurnaroundTime Usec;
  aTxPLCPDelay Usec;
  aRxPLCPDelay Usec;
  aRxTxSwitchTime Usec;
  aTxRampOnTime Usec;
  aTxRampOffTime Usec;
  aTxRFDelay Usec;
  aRxRFDelay Usec;
  aAirPropagationTime Usec;
  aMACProcessingDelay Usec;
  aPreambleLength Usec;
  aPLCPHeaderLength Usec;
  aMPDUMaxLength Integer;
  aCWmin Integer;
  aCWmax Integer;
EndNewType PhyChrstcs;
```

```
use macsorts ;
```

Package macmib

3201_d\StationConfig(5)

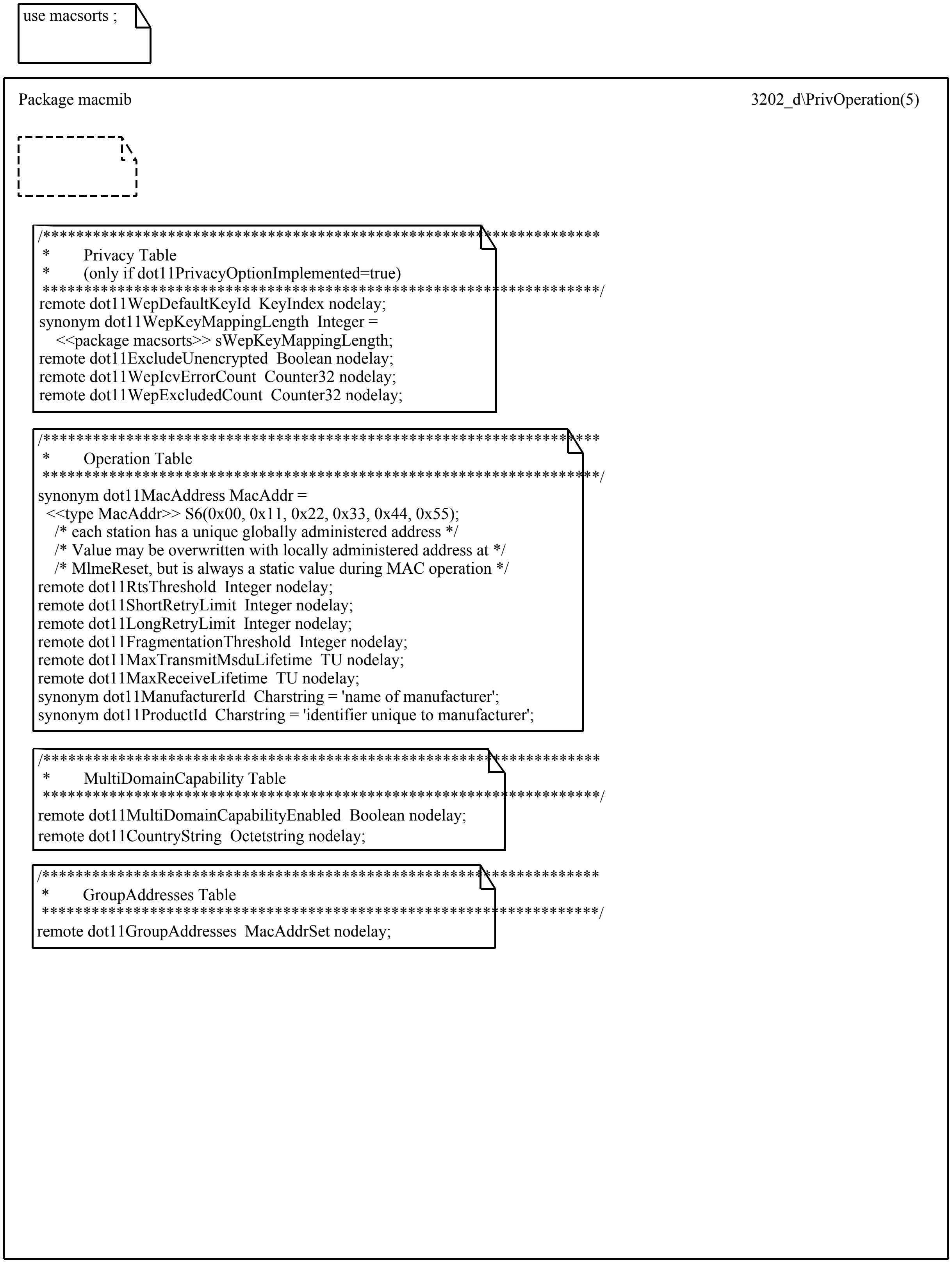
```
/* This Package contains definitions of the MAC MIB attributes
   and the subset of the PHY MIB attributes used by the MAC state
   machines.  These are needed under Z.100 to permit analysis of
   the state machine definitions.  In future revisions these may
   be replaced with the ASN.1 MIB definition which appears as
   as Annex D, for use with a Z.105-compliant SDL tool is available. */
```

```
/*********************************************************************
 *       StationConfig Table
 *********************************************************************/
remote dot11MediumOccupancyLimit  TU nodelay;
synonym dot11CfPollable  Boolean = <<package macsorts>> sCFPollable;
remote dot11CfpPeriod  Integer nodelay;
remote dot11CfpMaxDuration  Integer nodelay;
remote dot11AuthenticationResponseTimeout  TU nodelay;
synonym dot11PrivacyOptionImplemented Boolean = true;
remote dot11PowerMangementMode PsMode = sta_active;
remote dot11DesiredSSID Octetstring nodelay;
remote dot11DesiredBSSType BssType nodelay;
remote dot11OperationalRateSet Octetstring nodelay;
remote dot11BeaconPeriod TU nodelay;
remote dot11DtimPeriod Integer nodelay;
remote dot11AssociationResponseTimeout TU nodelay;
remote dot11DisassociateReason ReasonCode nodelay;
remote dot11DisassociateStation MacAddr nodelay;
remote dot11DeauthenticateReason ReasonCode nodelay;
remote dot11DeauthenticateStation MacAddr nodelay;
remote dot11AuthenticateFailStatus StatusCode nodelay;
remote dot11AuthenticateFailStation MacAddr nodelay;
synonym dot11MultiDomainCapabilityImplemented Boolean = false;
```

```
/*********************************************************************
 *       AuthenticationAlgorithms Table
 *********************************************************************/
synonym dot11AuthenticationAlgorithms  AuthTypeSet =
   incl(open_system, incl(shared_key));
      /* NOTE:  The members of this set are the
          dot11AuthenticationAlgorithm values of all
         dot11 AuthenticationAlgorithmsEntry instances
         for which dot11AuthenticationAlgorithmsEnable=True.
         Do not include shared_key in this set
         unless dot11PrivacyOptionImplemented=true.  */
```

```
/*********************************************************************
 *       WepDefaultKeys Table
 *       (if dot11PrivacyOptionImplemented=true)
 *********************************************************************/
remote dot11WepDefaultKeys  KeyVector nodelay;
```

```
/*********************************************************************
 *       WepKeyMappings Table
 *       (if dot11PrivacyOptionImplemented=true)
 *********************************************************************/
remote dot11WepKeyMappings  KeyMapArray nodelay;
```



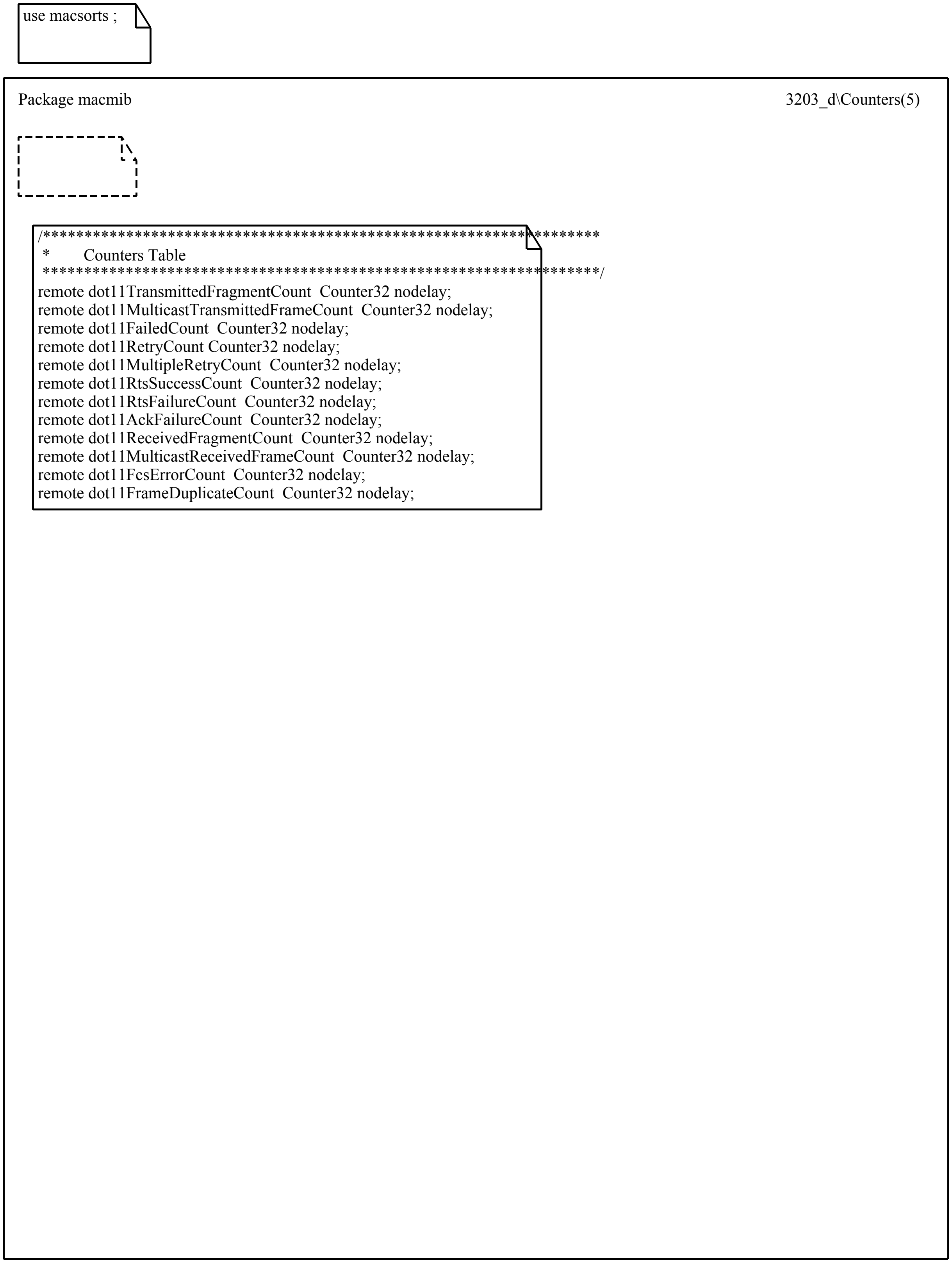
```
use macsorts ;
```

```
/*******************************************************************
 *      Privacy Table
 *      (only if dot11PrivacyOptionImplemented=true)
 *******************************************************************/
remote dot11WepDefaultKeyId  KeyIndex nodelay;
synonym dot11WepKeyMappingLength  Integer =
   <<package macsorts>> sWepKeyMappingLength;
remote dot11ExcludeUnencrypted  Boolean nodelay;
remote dot11WepIcvErrorCount  Counter32 nodelay;
remote dot11WepExcludedCount  Counter32 nodelay;
```

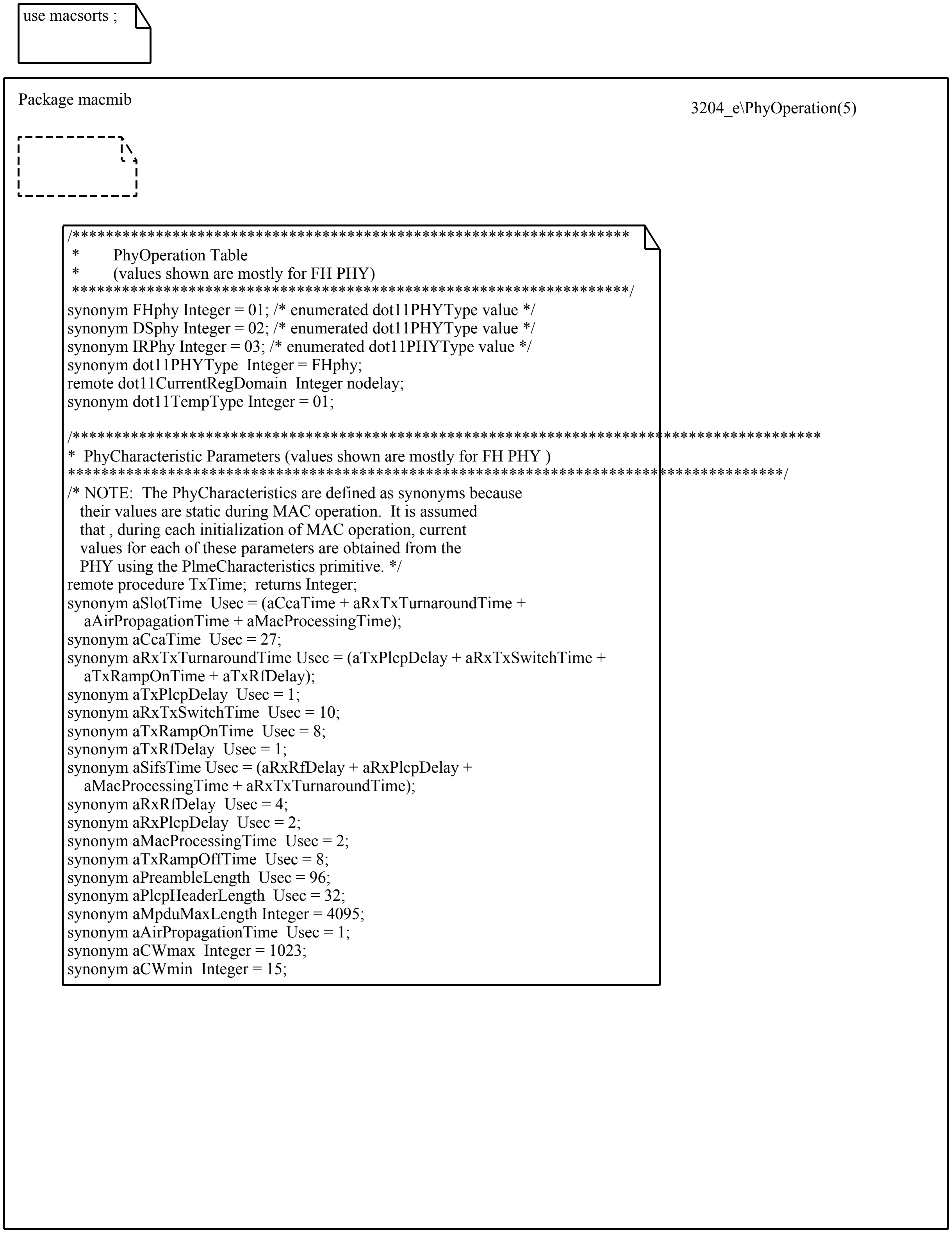
```
/*******************************************************************
 *      Operation Table
 *******************************************************************/
synonym dot11MacAddress MacAddr =
  <<type MacAddr>> S6(0x00, 0x11, 0x22, 0x33, 0x44, 0x55);
   /* each station has a unique globally administered address */
   /* Value may be overwritten with locally administered address at */
   /* MlmeReset, but is always a static value during MAC operation */
remote dot11RtsThreshold  Integer nodelay;
remote dot11ShortRetryLimit  Integer nodelay;
remote dot11LongRetryLimit  Integer nodelay;
remote dot11FragmentationThreshold  Integer nodelay;
remote dot11MaxTransmitMsduLifetime  TU nodelay;
remote dot11MaxReceiveLifetime  TU nodelay;
synonym dot11ManufacturerId  Charstring = 'name of manufacturer';
synonym dot11ProductId  Charstring = 'identifier unique to manufacturer';
```

```
/*******************************************************************
 *      MultiDomainCapability Table
 *******************************************************************/
remote dot11MultiDomainCapabilityEnabled  Boolean nodelay;
remote dot11CountryString  Octetstring nodelay;
```

```
/*******************************************************************
 *      GroupAddresses Table
 *******************************************************************/
remote dot11GroupAddresses  MacAddrSet nodelay;
```



```
/*****************************************************************
 *       Counters Table
 *****************************************************************/
remote dot11TransmittedFragmentCount  Counter32 nodelay;
remote dot11MulticastTransmittedFrameCount  Counter32 nodelay;
remote dot11FailedCount  Counter32 nodelay;
remote dot11RetryCount Counter32 nodelay;
remote dot11MultipleRetryCount  Counter32 nodelay;
remote dot11RtsSuccessCount  Counter32 nodelay;
remote dot11RtsFailureCount  Counter32 nodelay;
remote dot11AckFailureCount  Counter32 nodelay;
remote dot11ReceivedFragmentCount  Counter32 nodelay;
remote dot11MulticastReceivedFrameCount  Counter32 nodelay;
remote dot11FcsErrorCount  Counter32 nodelay;
remote dot11FrameDuplicateCount  Counter32 nodelay;
```



```
use macsorts ;
```

Package macmib

3204_e\PhyOperation(5)

```
/*******************************************************************
 *      PhyOperation Table
 *      (values shown are mostly for FH PHY)
 *******************************************************************/
synonym FHphy Integer = 01; /* enumerated dot11PHYType value */
synonym DSphy Integer = 02; /* enumerated dot11PHYType value */
synonym IRPhy Integer = 03; /* enumerated dot11PHYType value */
synonym dot11PHYType  Integer = FHphy;
remote dot11CurrentRegDomain  Integer nodelay;
synonym dot11TempType Integer = 01;

/*******************************************************************************************
*  PhyCharacteristic Parameters (values shown are mostly for FH PHY )
***************************************************************************************/
/* NOTE:  The PhyCharacteristics are defined as synonyms because
   their values are static during MAC operation.  It is assumed
   that , during each initialization of MAC operation, current
   values for each of these parameters are obtained from the
   PHY using the PlmeCharacteristics primitive. */
remote procedure TxTime;  returns Integer;
synonym aSlotTime  Usec = (aCcaTime + aRxTxTurnaroundTime +
   aAirPropagationTime + aMacProcessingTime);
synonym aCcaTime  Usec = 27;
synonym aRxTxTurnaroundTime Usec = (aTxPlcpDelay + aRxTxSwitchTime +
   aTxRampOnTime + aTxRfDelay);
synonym aTxPlcpDelay  Usec = 1;
synonym aRxTxSwitchTime  Usec = 10;
synonym aTxRampOnTime  Usec = 8;
synonym aTxRfDelay  Usec = 1;
synonym aSifsTime Usec = (aRxRfDelay + aRxPlcpDelay +
   aMacProcessingTime + aRxTxTurnaroundTime);
synonym aRxRfDelay  Usec = 4;
synonym aRxPlcpDelay  Usec = 2;
synonym aMacProcessingTime  Usec = 2;
synonym aTxRampOffTime  Usec = 8;
synonym aPreambleLength  Usec = 96;
synonym aPlcpHeaderLength  Usec = 32;
synonym aMpduMaxLength Integer = 4095;
synonym aAirPropagationTime  Usec = 1;
synonym aCWmax  Integer = 1023;
synonym aCWmin  Integer = 15;
```

```
use macsorts ;
```

Package macmib 3205_d\PhyRateFhss(5)

```
/*************************************************************************************
 *       SupportedDataRatesTx Table (values shown are  for FH PHY)
 ************************************************************************************/
synonym aSupportedRatesTx  Octetstring = S8(0x82, 0x04, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00);

/****************************************************************************************
 *       SupportedDataRatesRx Table (values shown are  for FH PHY)
 ***************************************************************************************/
synonym aSupportedRatesRx  Octetstring = S8(0x82, 0x04, 0x00,0x00, 0x00, 0x00, 0x00, 0x00);

synonym aPrefMaxMpduFragmentLength  Integer = aMpduMaxLength;
```

```
/************************************************************************
 *       PhyFHSS Table
 *       (only used with FH PHY)
 ***********************************************************************/
synonym dot11HopTime  Usec = 224;
remote dot11CurrentChannelNumber  Integer nodelay;
synonym dot11MaxDwellTime  TU = 390;
remote dot11CurrentSet  Integer nodelay;
remote dot11CurrentPattern  Integer nodelay;
remote dot11CurrentIndex  Integer nodelay;
```

```
/*******************************************************************************************/
/* The MAC state machines currently do not reference any attributes in:
     PhyAntenna Table,  PhyTxPower Table,  PhyDsss Table,  PhyIr Table,
     RegDomainsSupported Table,  AntennasList Table.  */
/*endpackage;*/
/*******************************************************************************************/
```

J.5 State machines for MAC STAs

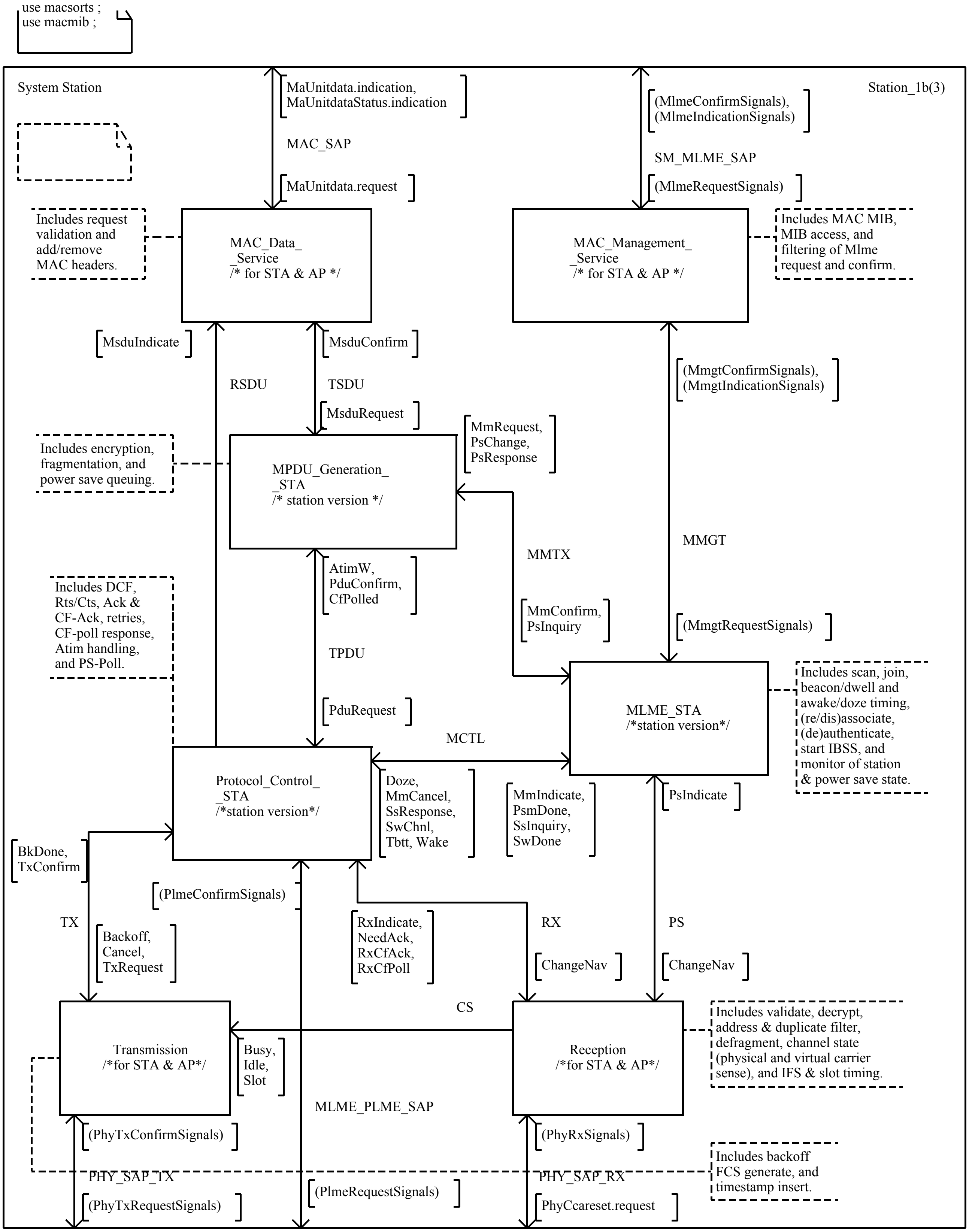
The following SDL-92 system specification defines operation of the MAC protocol at an IEEE 802.11 STA. Many aspects of STA operation also apply to AP operation. These are defined in blocks and processes referenced from both the STA and AP system specifications. Blocks and processes used in both STA and AP are identifiable by the SDL comment /* for STA & AP */ below the block or process name. Blocks and processes specific to STA operation are identifiable by the SDL comment /* station version */ below the block or process name. The definitions of all blocks and processes referenced in the STA system specification appear in J.5.

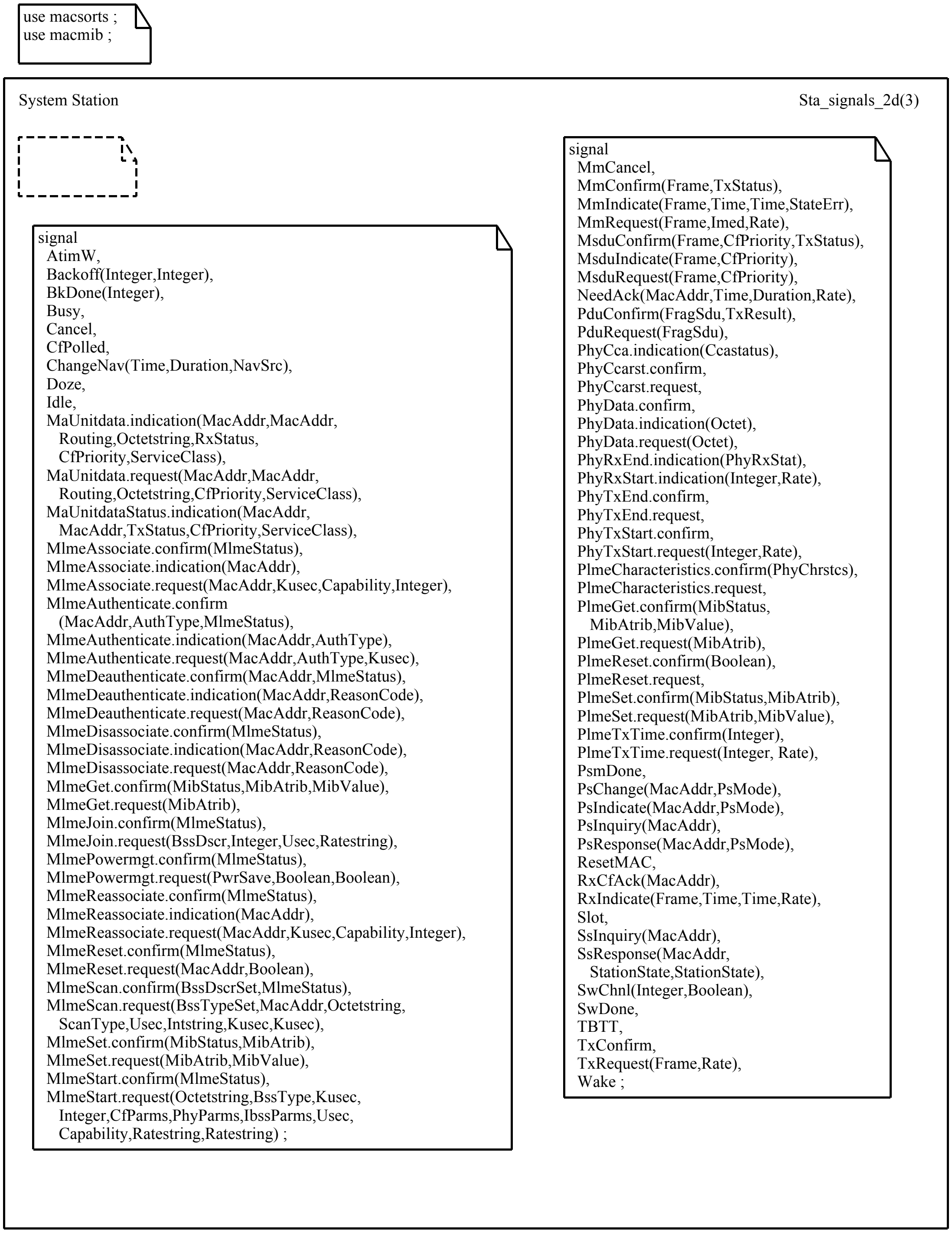
The remainder of J.5 is the formal description, in SDL/GR, of an IEEE 802.11 STA.

This subclause describes the security behavior of only 11.2.2 and 11.2.3.

This subclause does not describe the behavior of a STA with QoS facility.

This clause does not describe the behavior of an HT STA.

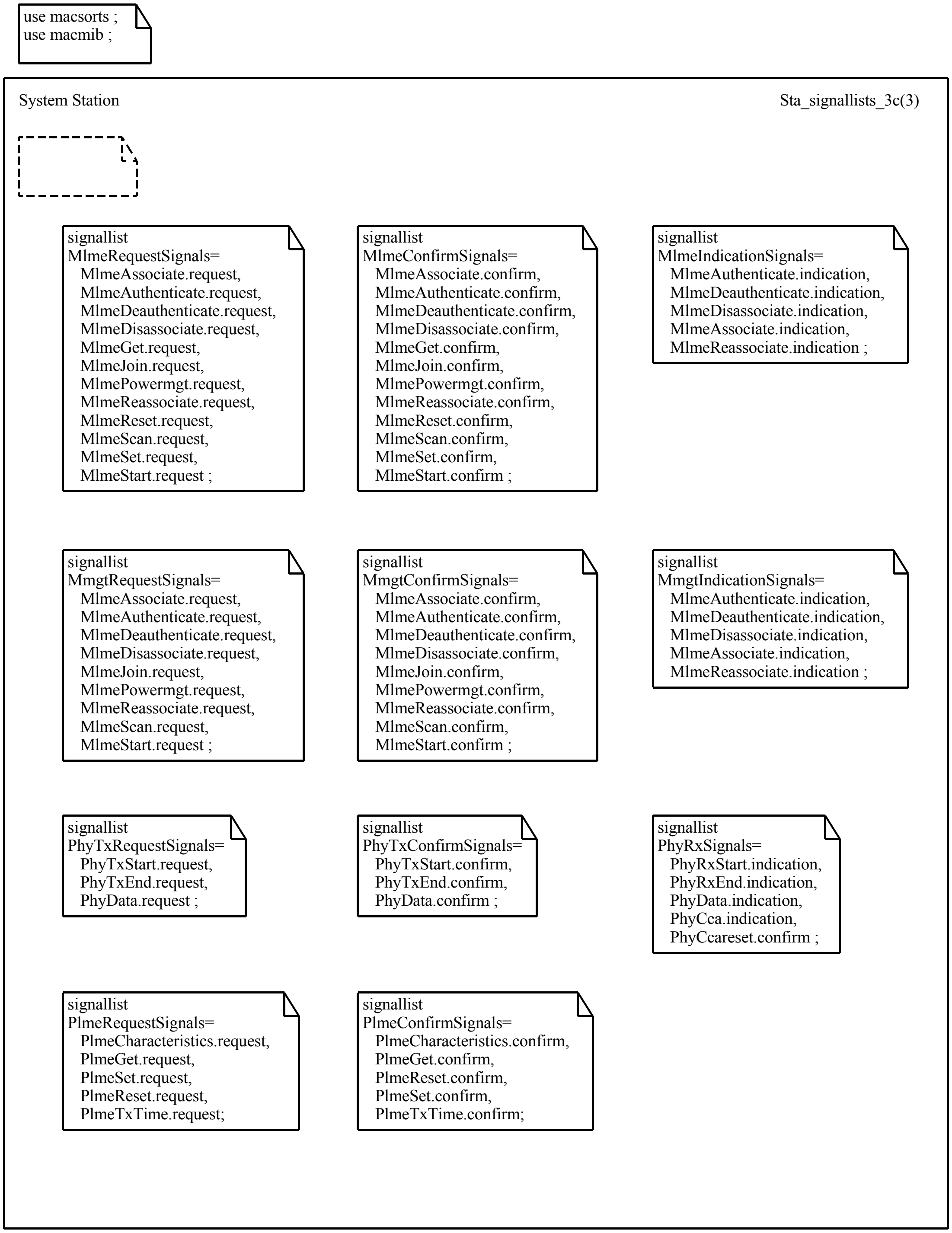

use macsorts ;
use macmib ;
System Station
Station_1b(3)
MaUnitdata.indication,
MaUnitdataStatus.indication
MAC_SAP
MaUnitdata.request
(MlmeConfirmSignals),
(MlmeIndicationSignals)
SM_MLME_SAP
(MlmeRequestSignals)
Includes request validation and add/remove MAC headers.
MAC_Data_
_Service
/* for STA & AP */
MAC_Management_
_Service
/* for STA & AP */
Includes MAC MIB, MIB access, and filtering of Mlme request and confirm.
MsduIndicate
MsduConfirm
RSDU
TSDU
MsduRequest
(MmgtConfirmSignals),
(MmgtIndicationSignals)
MmRequest,
PsChange,
PsResponse
Includes encryption, fragmentation, and power save queuing.
MPDU_Generation_
_STA
/* station version */
MMTX
MMGT
AtimW,
PduConfirm,
CfPolled
Includes DCF, Rts/Cts, Ack & CF-Ack, retries, CF-poll response, Atim handling, and PS-Poll.
MmConfirm,
PsInquiry
(MmgtRequestSignals)
TPDU
PduRequest
MLME_STA
/*station version*/
Includes scan, join, beacon/dwell and awake/doze timing, (re/dis)associate, (de)authenticate, start IBSS, and monitor of station & power save state.
MCTL
Protocol_Control_
_STA
/*station version*/
Doze,
MmCancel,
SsResponse,
SwChnl,
Tbtt, Wake
MmIndicate,
PsmDone,
SsInquiry,
SwDone
PsIndicate
BkDone,
TxConfirm
(PlmeConfirmSignals)
TX
Backoff,
Cancel,
TxRequest
RxIndicate,
NeedAck,
RxCfAck,
RxCfPoll
RX
PS
ChangeNav
ChangeNav
CS
Transmission
/*for STA & AP*/
Busy,
Idle,
Slot
Reception
/*for STA & AP*/
Includes validate, decrypt, address & duplicate filter, defragment, channel state (physical and virtual carrier sense), and IFS & slot timing.
MLME_PLME_SAP
(PhyTxConfirmSignals)
(PhyRxSignals)
PHY_SAP_TX
PHY_SAP_RX
Includes backoff FCS generate, and timestamp insert.
(PhyTxRequestSignals)
(PlmeRequestSignals)
PhyCcareset.request



```
use macsorts ;
use macmib ;
```

```
signal
  AtimW,
  Backoff(Integer,Integer),
  BkDone(Integer),
  Busy,
  Cancel,
  CfPolled,
  ChangeNav(Time,Duration,NavSrc),
  Doze,
  Idle,
  MaUnitdata.indication(MacAddr,MacAddr,
    Routing,Octetstring,RxStatus,
    CfPriority,ServiceClass),
  MaUnitdata.request(MacAddr,MacAddr,
    Routing,Octetstring,CfPriority,ServiceClass),
  MaUnitdataStatus.indication(MacAddr,
    MacAddr,TxStatus,CfPriority,ServiceClass),
  MlmeAssociate.confirm(MlmeStatus),
  MlmeAssociate.indication(MacAddr),
  MlmeAssociate.request(MacAddr,Kusec,Capability,Integer),
  MlmeAuthenticate.confirm
    (MacAddr,AuthType,MlmeStatus),
  MlmeAuthenticate.indication(MacAddr,AuthType),
  MlmeAuthenticate.request(MacAddr,AuthType,Kusec),
  MlmeDeauthenticate.confirm(MacAddr,MlmeStatus),
  MlmeDeauthenticate.indication(MacAddr,ReasonCode),
  MlmeDeauthenticate.request(MacAddr,ReasonCode),
  MlmeDisassociate.confirm(MlmeStatus),
  MlmeDisassociate.indication(MacAddr,ReasonCode),
  MlmeDisassociate.request(MacAddr,ReasonCode),
  MlmeGet.confirm(MibStatus,MibAtrib,MibValue),
  MlmeGet.request(MibAtrib),
  MlmeJoin.confirm(MlmeStatus),
  MlmeJoin.request(BssDscr,Integer,Usec,Ratestring),
  MlmePowermgt.confirm(MlmeStatus),
  MlmePowermgt.request(PwrSave,Boolean,Boolean),
  MlmeReassociate.confirm(MlmeStatus),
  MlmeReassociate.indication(MacAddr),
  MlmeReassociate.request(MacAddr,Kusec,Capability,Integer),
  MlmeReset.confirm(MlmeStatus),
  MlmeReset.request(MacAddr,Boolean),
  MlmeScan.confirm(BssDscrSet,MlmeStatus),
  MlmeScan.request(BssTypeSet,MacAddr,Octetstring,
    ScanType,Usec,Intstring,Kusec,Kusec),
  MlmeSet.confirm(MibStatus,MibAtrib),
  MlmeSet.request(MibAtrib,MibValue),
  MlmeStart.confirm(MlmeStatus),
  MlmeStart.request(Octetstring,BssType,Kusec,
    Integer,CfParms,PhyParms,IbssParms,Usec,
    Capability,Ratestring,Ratestring) ;
```

```
signal
  MmCancel,
  MmConfirm(Frame,TxStatus),
  MmIndicate(Frame,Time,Time,StateErr),
  MmRequest(Frame,Imed,Rate),
  MsduConfirm(Frame,CfPriority,TxStatus),
  MsduIndicate(Frame,CfPriority),
  MsduRequest(Frame,CfPriority),
  NeedAck(MacAddr,Time,Duration,Rate),
  PduConfirm(FragSdu,TxResult),
  PduRequest(FragSdu),
  PhyCca.indication(Ccastatus),
  PhyCcarst.confirm,
  PhyCcarst.request,
  PhyData.confirm,
  PhyData.indication(Octet),
  PhyData.request(Octet),
  PhyRxEnd.indication(PhyRxStat),
  PhyRxStart.indication(Integer,Rate),
  PhyTxEnd.confirm,
  PhyTxEnd.request,
  PhyTxStart.confirm,
  PhyTxStart.request(Integer,Rate),
  PlmeCharacteristics.confirm(PhyChrstcs),
  PlmeCharacteristics.request,
  PlmeGet.confirm(MibStatus,
    MibAtrib,MibValue),
  PlmeGet.request(MibAtrib),
  PlmeReset.confirm(Boolean),
  PlmeReset.request,
  PlmeSet.confirm(MibStatus,MibAtrib),
  PlmeSet.request(MibAtrib,MibValue),
  PlmeTxTime.confirm(Integer),
  PlmeTxTime.request(Integer, Rate),
  PsmDone,
  PsChange(MacAddr,PsMode),
  PsIndicate(MacAddr,PsMode),
  PsInquiry(MacAddr),
  PsResponse(MacAddr,PsMode),
  ResetMAC,
  RxCfAck(MacAddr),
  RxIndicate(Frame,Time,Time,Rate),
  Slot,
  SsInquiry(MacAddr),
  SsResponse(MacAddr,
    StationState,StationState),
  SwChnl(Integer,Boolean),
  SwDone,
  TBTT,
  TxConfirm,
  TxRequest(Frame,Rate),
  Wake ;
```



```
use macsorts ;
use macmib ;
```

System Station

Sta_signallists_3c(3)

```
signallist
MlmeRequestSignals=
   MlmeAssociate.request,
   MlmeAuthenticate.request,
   MlmeDeauthenticate.request,
   MlmeDisassociate.request,
   MlmeGet.request,
   MlmeJoin.request,
   MlmePowermgt.request,
   MlmeReassociate.request,
   MlmeReset.request,
   MlmeScan.request,
   MlmeSet.request,
   MlmeStart.request ;
```

```
signallist
MlmeConfirmSignals=
   MlmeAssociate.confirm,
   MlmeAuthenticate.confirm,
   MlmeDeauthenticate.confirm,
   MlmeDisassociate.confirm,
   MlmeGet.confirm,
   MlmeJoin.confirm,
   MlmePowermgt.confirm,
   MlmeReassociate.confirm,
   MlmeReset.confirm,
   MlmeScan.confirm,
   MlmeSet.confirm,
   MlmeStart.confirm ;
```

```
signallist
MlmeIndicationSignals=
   MlmeAuthenticate.indication,
   MlmeDeauthenticate.indication,
   MlmeDisassociate.indication,
   MlmeAssociate.indication,
   MlmeReassociate.indication ;
```

```
signallist
MmgtRequestSignals=
   MlmeAssociate.request,
   MlmeAuthenticate.request,
   MlmeDeauthenticate.request,
   MlmeDisassociate.request,
   MlmeJoin.request,
   MlmePowermgt.request,
   MlmeReassociate.request,
   MlmeScan.request,
   MlmeStart.request ;
```

```
signallist
MmgtConfirmSignals=
   MlmeAssociate.confirm,
   MlmeAuthenticate.confirm,
   MlmeDeauthenticate.confirm,
   MlmeDisassociate.confirm,
   MlmeJoin.confirm,
   MlmePowermgt.confirm,
   MlmeReassociate.confirm,
   MlmeScan.confirm,
   MlmeStart.confirm ;
```

```
signallist
MmgtIndicationSignals=
   MlmeAuthenticate.indication,
   MlmeDeauthenticate.indication,
   MlmeDisassociate.indication,
   MlmeAssociate.indication,
   MlmeReassociate.indication ;
```

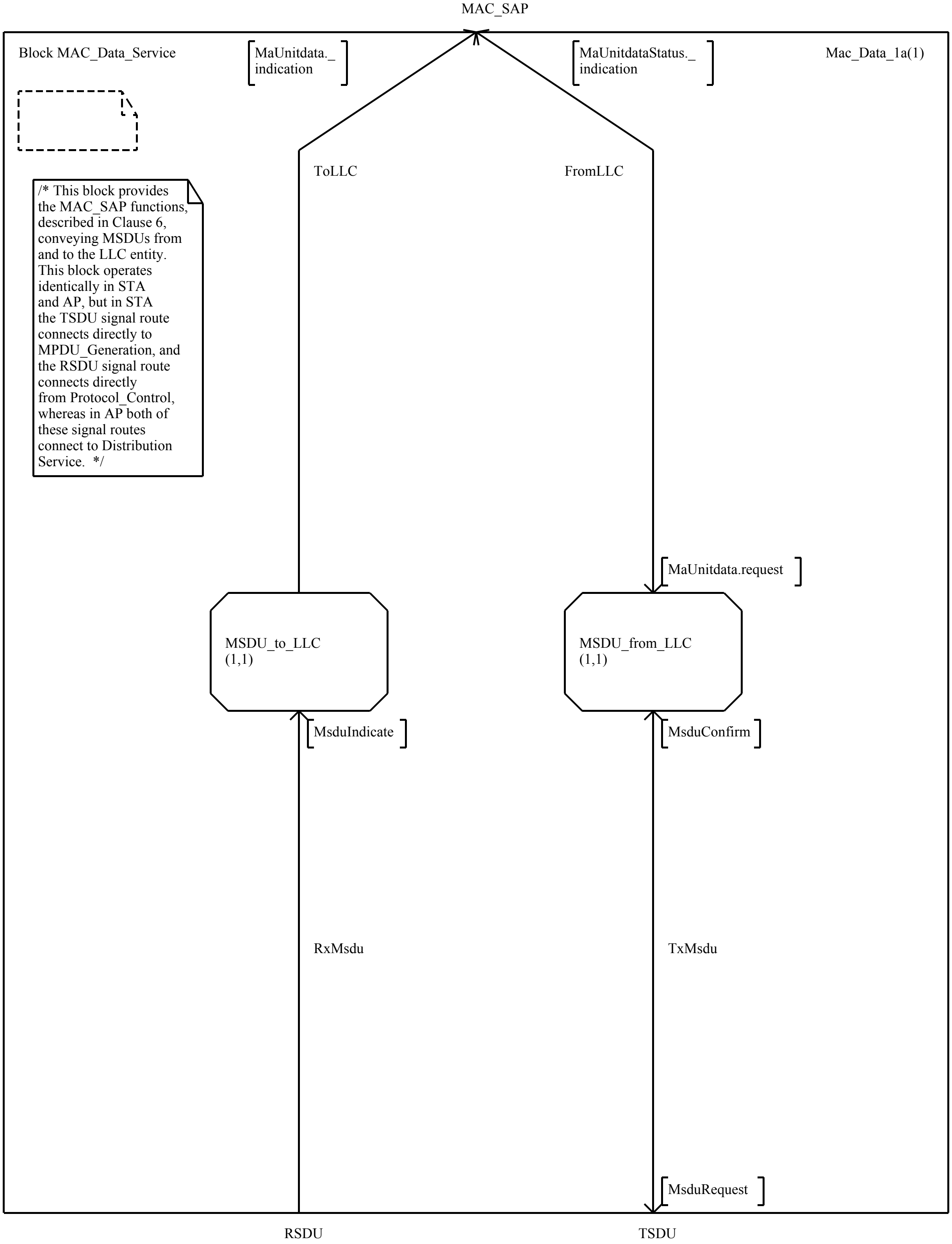
```
signallist
PhyTxRequestSignals=
   PhyTxStart.request,
   PhyTxEnd.request,
   PhyData.request ;
```

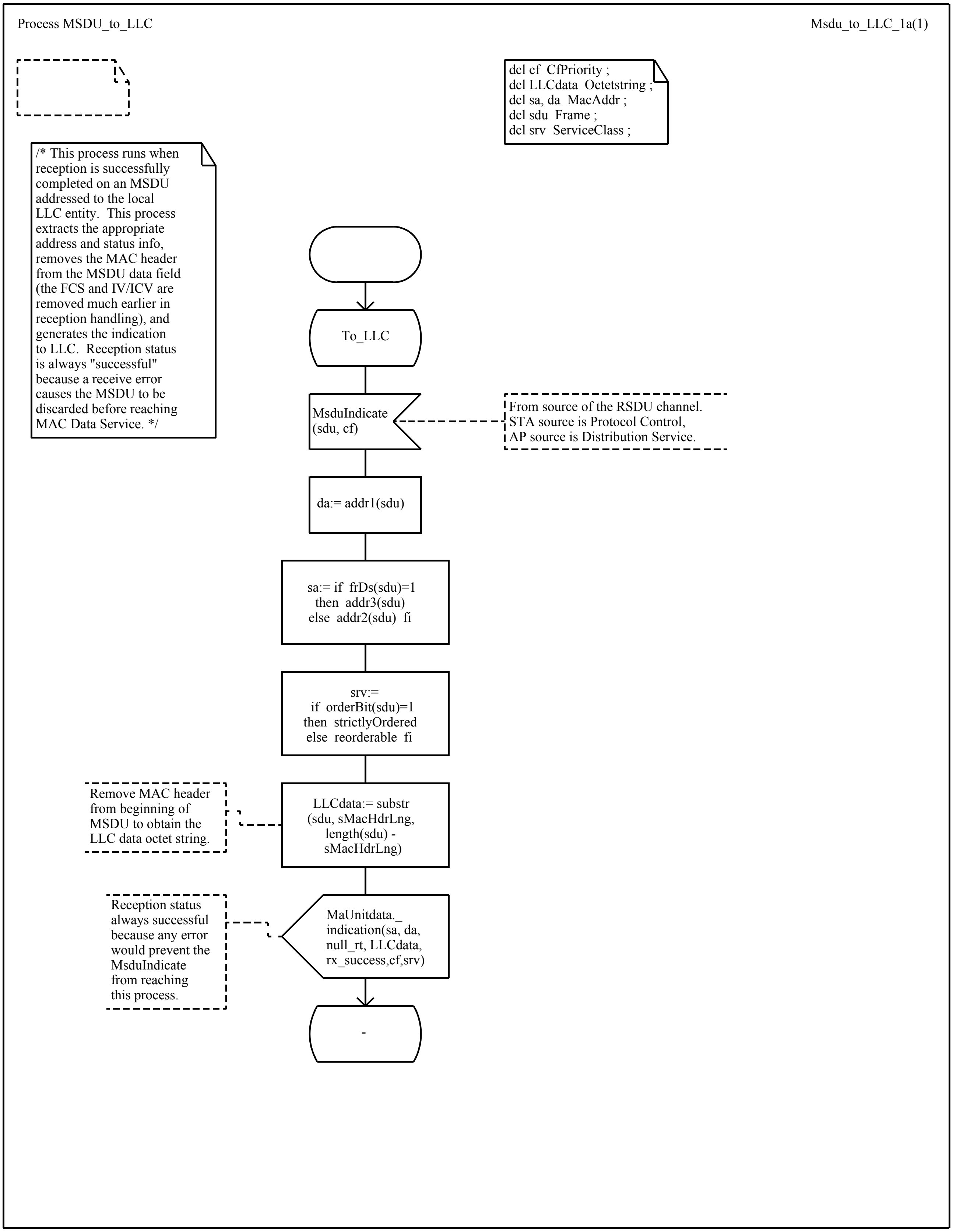
```
signallist
PhyTxConfirmSignals=
   PhyTxStart.confirm,
   PhyTxEnd.confirm,
   PhyData.confirm ;
```

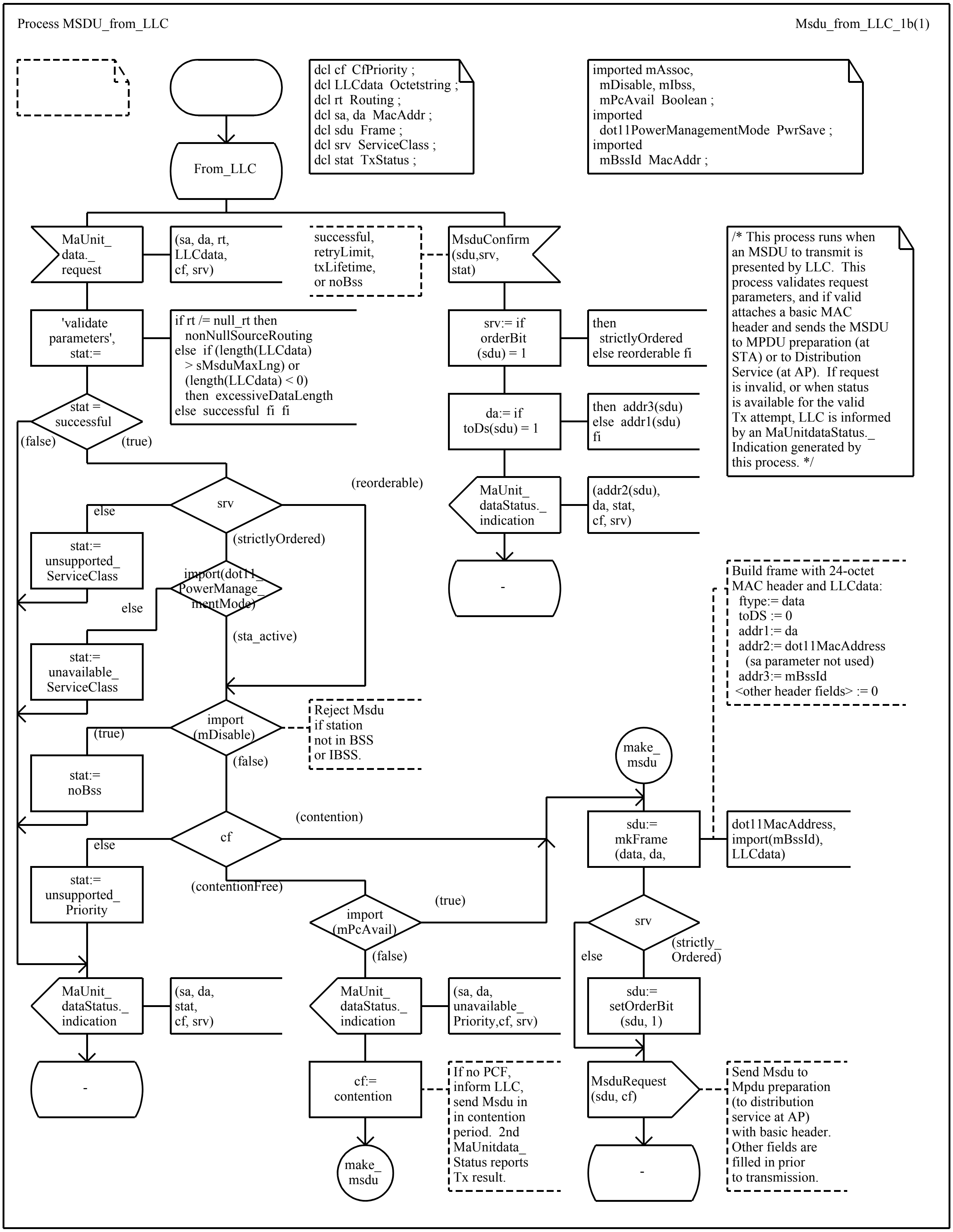
```
signallist
PhyRxSignals=
   PhyRxStart.indication,
   PhyRxEnd.indication,
   PhyData.indication,
   PhyCca.indication,
   PhyCcareset.confirm ;
```

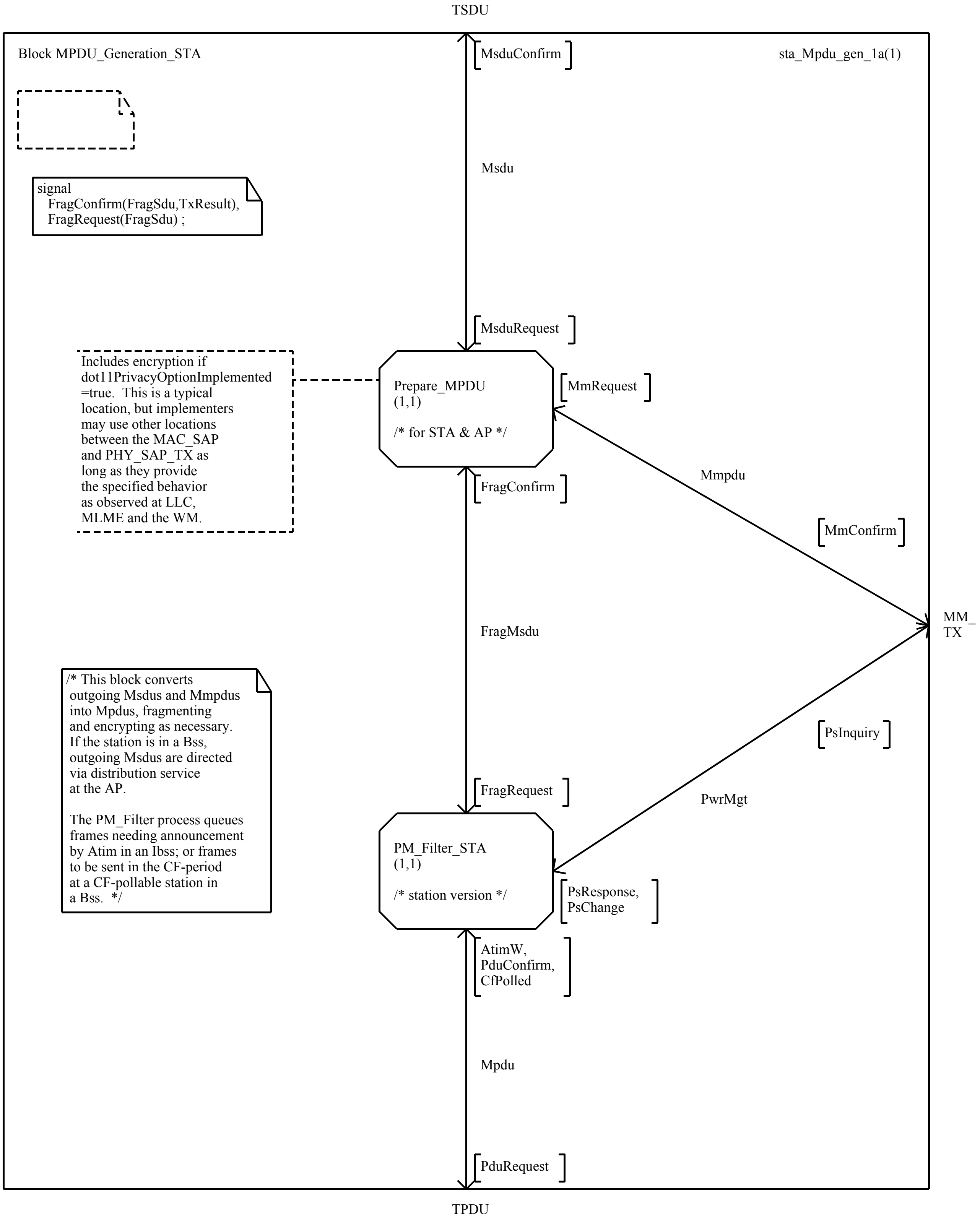
```
signallist
PlmeRequestSignals=
   PlmeCharacteristics.request,
   PlmeGet.request,
   PlmeSet.request,
   PlmeReset.request,
   PlmeTxTime.request;
```

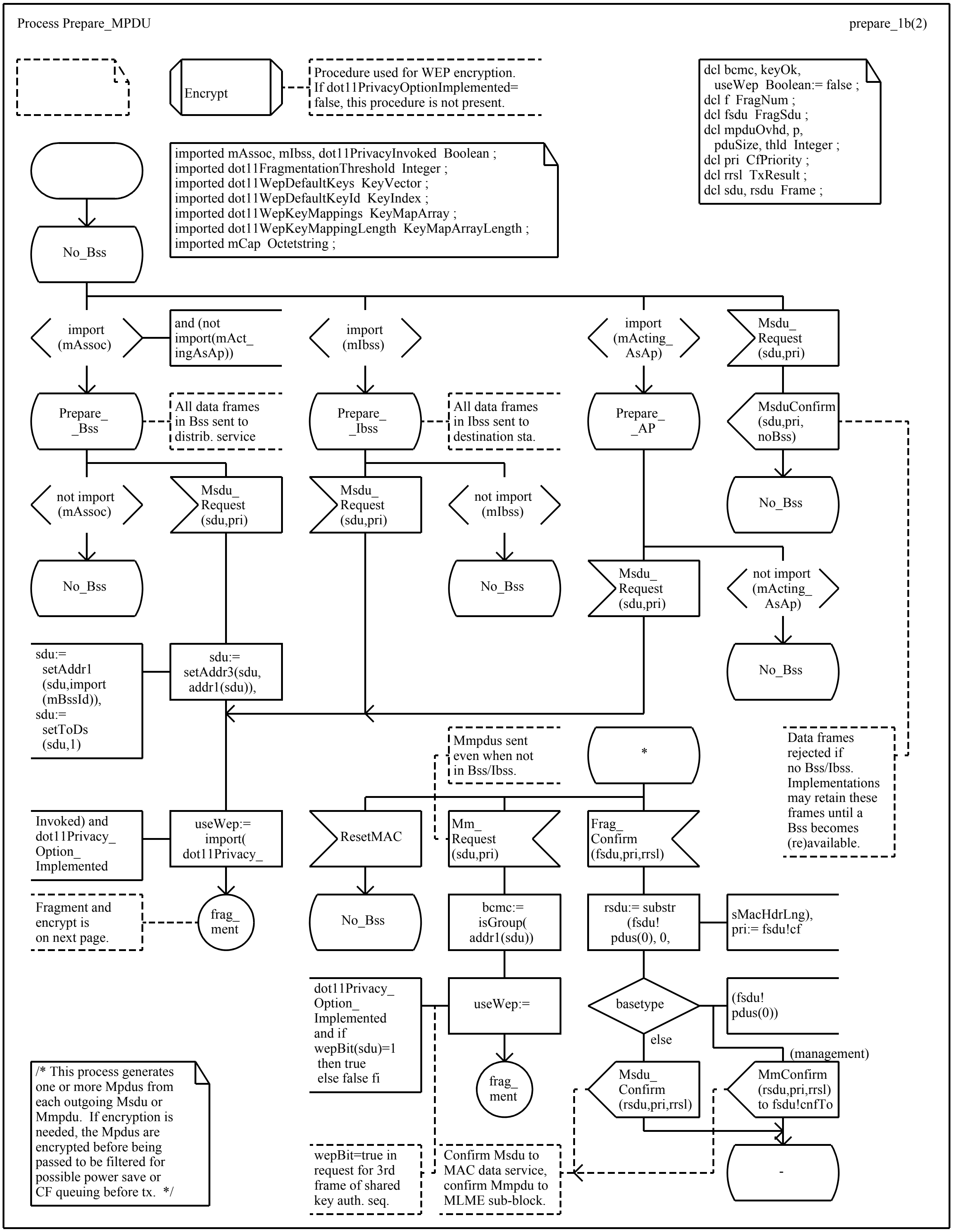
```
signallist
PlmeConfirmSignals=
   PlmeCharacteristics.confirm,
   PlmeGet.confirm,
   PlmeReset.confirm,
   PlmeSet.confirm,
   PlmeTxTime.confirm;
```

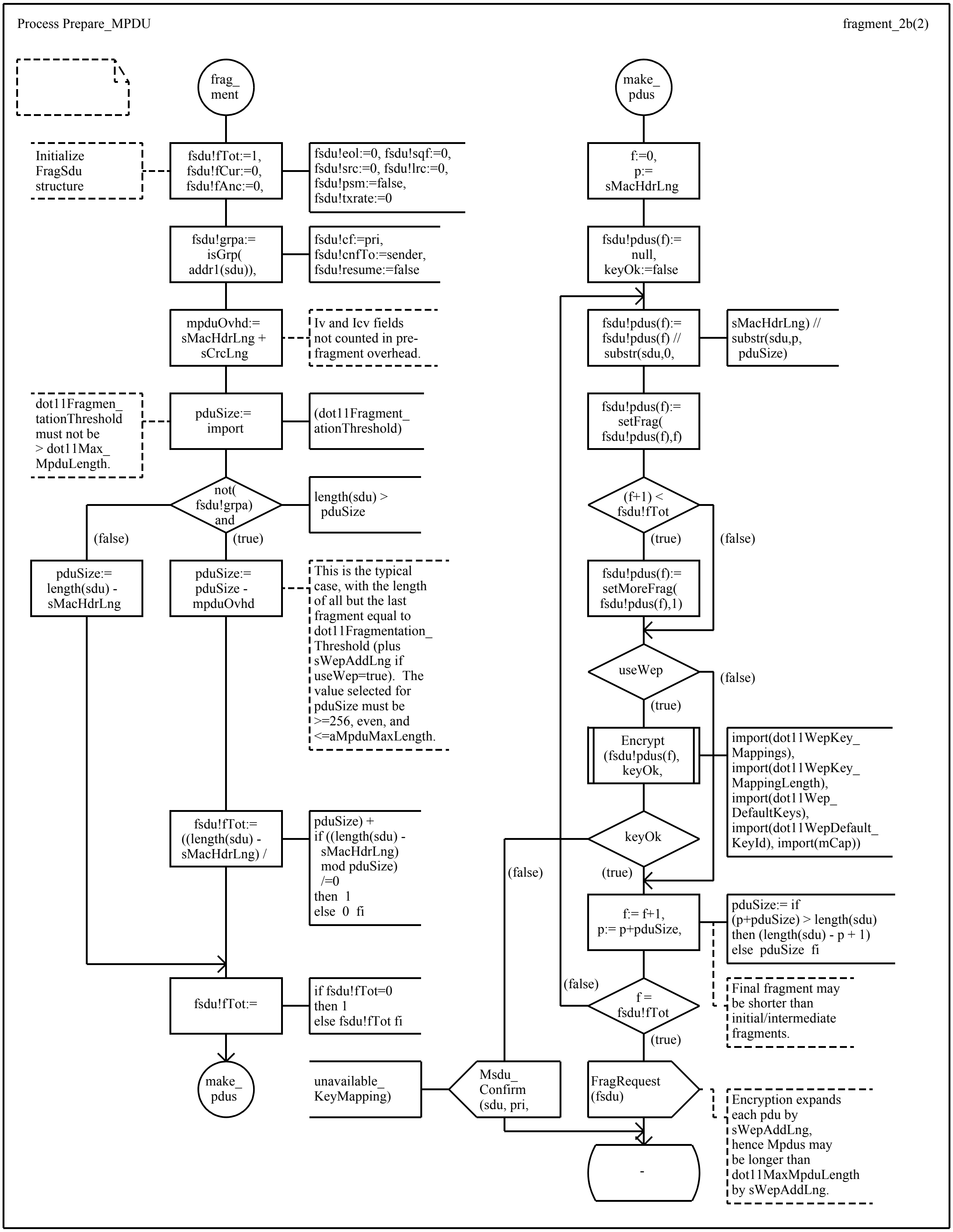

MAC_SAP
Block MAC_Data_Service
MaUnitdata._
indication
MaUnitdataStatus._
indication
Mac_Data_1a(1)
/* This block provides the MAC_SAP functions, described in Clause 6, conveying MSDUs from and to the LLC entity. This block operates identically in STA and AP, but in STA the TSDU signal route connects directly to MPDU_Generation, and the RSDU signal route connects directly from Protocol_Control, whereas in AP both of these signal routes connect to Distribution Service. */
ToLLC
FromLLC
MaUnitdata.request
MSDU_to_LLC
(1,1)
MSDU_from_LLC
(1,1)
MsduIndicate
MsduConfirm
RxMsdu
TxMsdu
MsduRequest
RSDU
TSDU

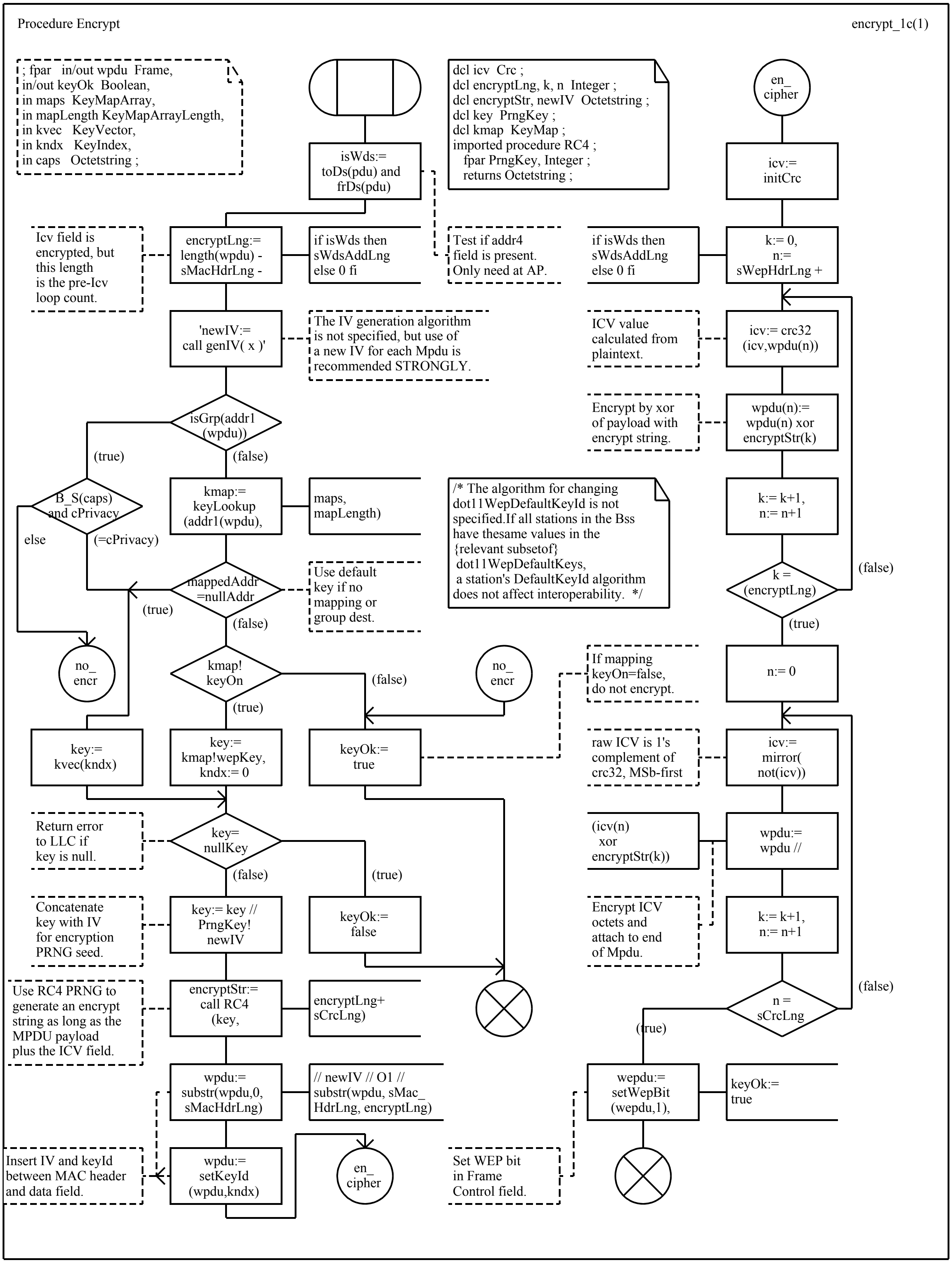

Process MSDU_to_LLC
Msdu_to_LLC_1a(1)
dcl cf CfPriority ;
dcl LLCdata Octetstring ;
dcl sa, da MacAddr ;
dcl sdu Frame ;
dcl srv ServiceClass ;
/* This process runs when reception is successfully completed on an MSDU addressed to the local LLC entity. This process extracts the appropriate address and status info, removes the MAC header from the MSDU data field (the FCS and IV/ICV are removed much earlier in reception handling), and generates the indication to LLC. Reception status is always "successful" because a receive error causes the MSDU to be discarded before reaching MAC Data Service. */
To_LLC
MsduIndicate (sdu, cf)
From source of the RSDU channel. STA source is Protocol Control, AP source is Distribution Service.
da:= addr1(sdu)
sa:= if frDs(sdu)=1 then addr3(sdu) else addr2(sdu) fi
srv:= if orderBit(sdu)=1 then strictlyOrdered else reorderable fi
Remove MAC header from beginning of MSDU to obtain the LLC data octet string.
LLCdata:= substr (sdu, sMacHdrLng, length(sdu) - sMacHdrLng)
Reception status always successful because any error would prevent the MsduIndicate from reaching this process.
MaUnitdata._ indication(sa, da, null_rt, LLCdata, rx_success,cf,srv)
-

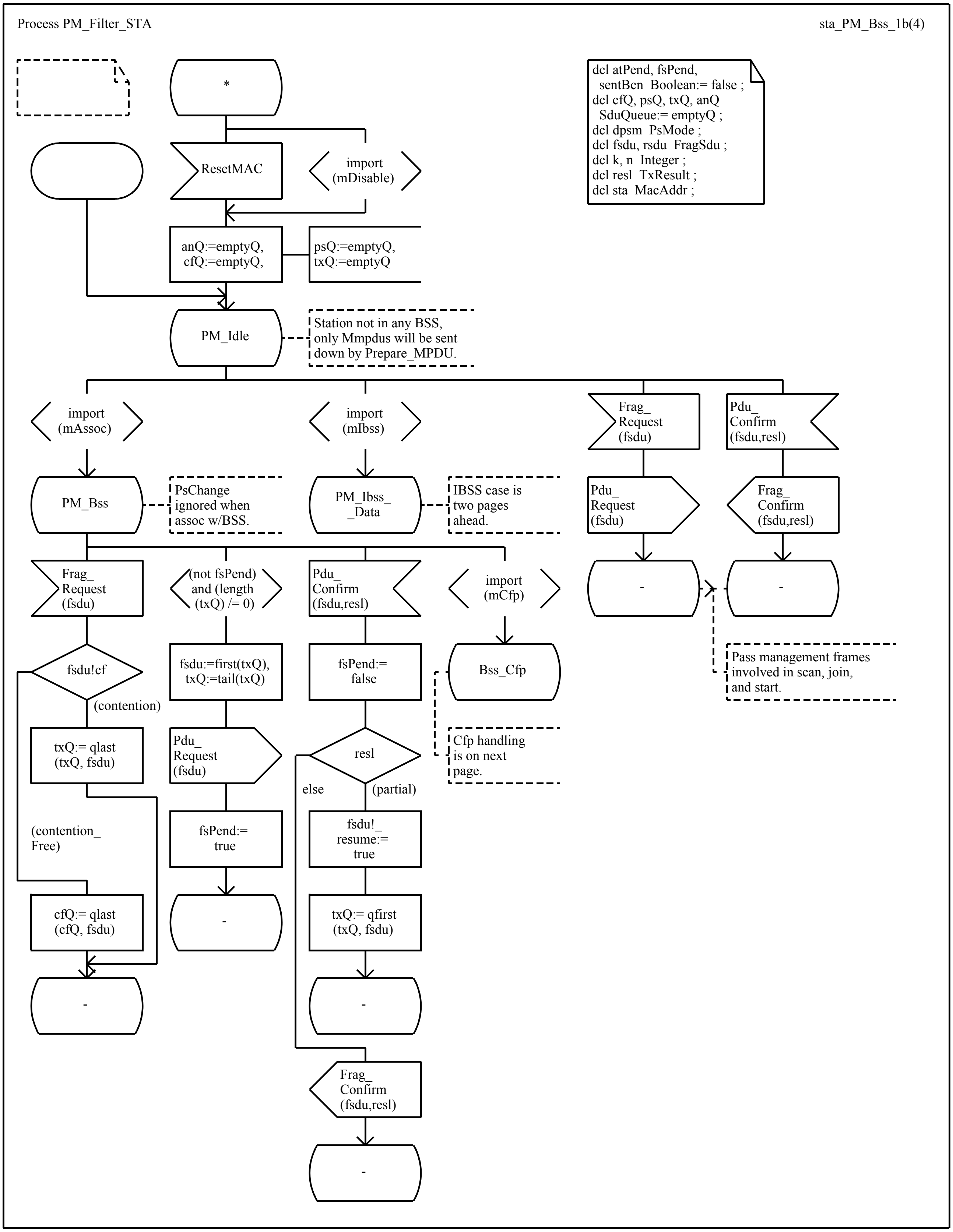

Process MSDU_from_LLC
Msdu_from_LLC_1b(1)
dcl cf CfPriority ;
dcl LLCdata Octetstring ;
dcl rt Routing ;
dcl sa, da MacAddr ;
dcl sdu Frame ;
dcl srv ServiceClass ;
dcl stat TxStatus ;
imported mAssoc,
mDisable, mIbss,
mPcAvail Boolean ;
imported
dot11PowerManagementMode PwrSave ;
imported
mBssId MacAddr ;
From_LLC
MaUnit_
data._
request
(sa, da, rt,
LLCdata,
cf, srv)
successful,
retryLimit,
txLifetime,
or noBss
MsduConfirm
(sdu,srv,
stat)
/* This process runs when an MSDU to transmit is presented by LLC. This process validates request parameters, and if valid attaches a basic MAC header and sends the MSDU to MPDU preparation (at STA) or to Distribution Service (at AP). If request is invalid, or when status is available for the valid Tx attempt, LLC is informed by an MaUnitdataStatus._ Indication generated by this process. */
'validate
parameters',
stat:=
if rt /= null_rt then
nonNullSourceRouting
else if (length(LLCdata)
> sMsduMaxLng) or
(length(LLCdata) < 0)
then excessiveDataLength
else successful fi fi
srv:= if
orderBit
(sdu) = 1
then
strictlyOrdered
else reorderable fi
stat =
successful
(false)
(true)
da:= if
toDs(sdu) = 1
then addr3(sdu)
else addr1(sdu)
fi
(reorderable)
srv
else
MaUnit_
dataStatus._
indication
(addr2(sdu),
da, stat,
cf, srv)
(strictlyOrdered)
stat:=
unsupported_
ServiceClass
import(dot11_
PowerManage_
mentMode)
else
-
Build frame with 24-octet
MAC header and LLCdata:
ftype:= data
toDS := 0
addr1:= da
addr2:= dot11MacAddress
(sa parameter not used)
addr3:= mBssId
<other header fields> := 0
(sta_active)
stat:=
unavailable_
ServiceClass
import
(mDisable)
(true)
Reject Msdu
if station
not in BSS
or IBSS.
make_
msdu
(false)
stat:=
noBss
(contention)
cf
else
sdu:=
mkFrame
(data, da,
dot11MacAddress,
import(mBssId),
LLCdata)
stat:=
unsupported_
Priority
(contentionFree)
import
(mPcAvail)
(true)
srv
else
(strictly_
Ordered)
(false)
MaUnit_
dataStatus._
indication
(sa, da,
stat,
cf, srv)
MaUnit_
dataStatus._
indication
(sa, da,
unavailable_
Priority,cf, srv)
sdu:=
setOrderBit
(sdu, 1)
-
cf:=
contention
If no PCF,
inform LLC,
send Msdu in
in contention
period. 2nd
MaUnitdata_
Status reports
Tx result.
MsduRequest
(sdu, cf)
Send Msdu to
Mpdu preparation
(to distribution
service at AP)
with basic header.
Other fields are
filled in prior
to transmission.
make_
msdu
-

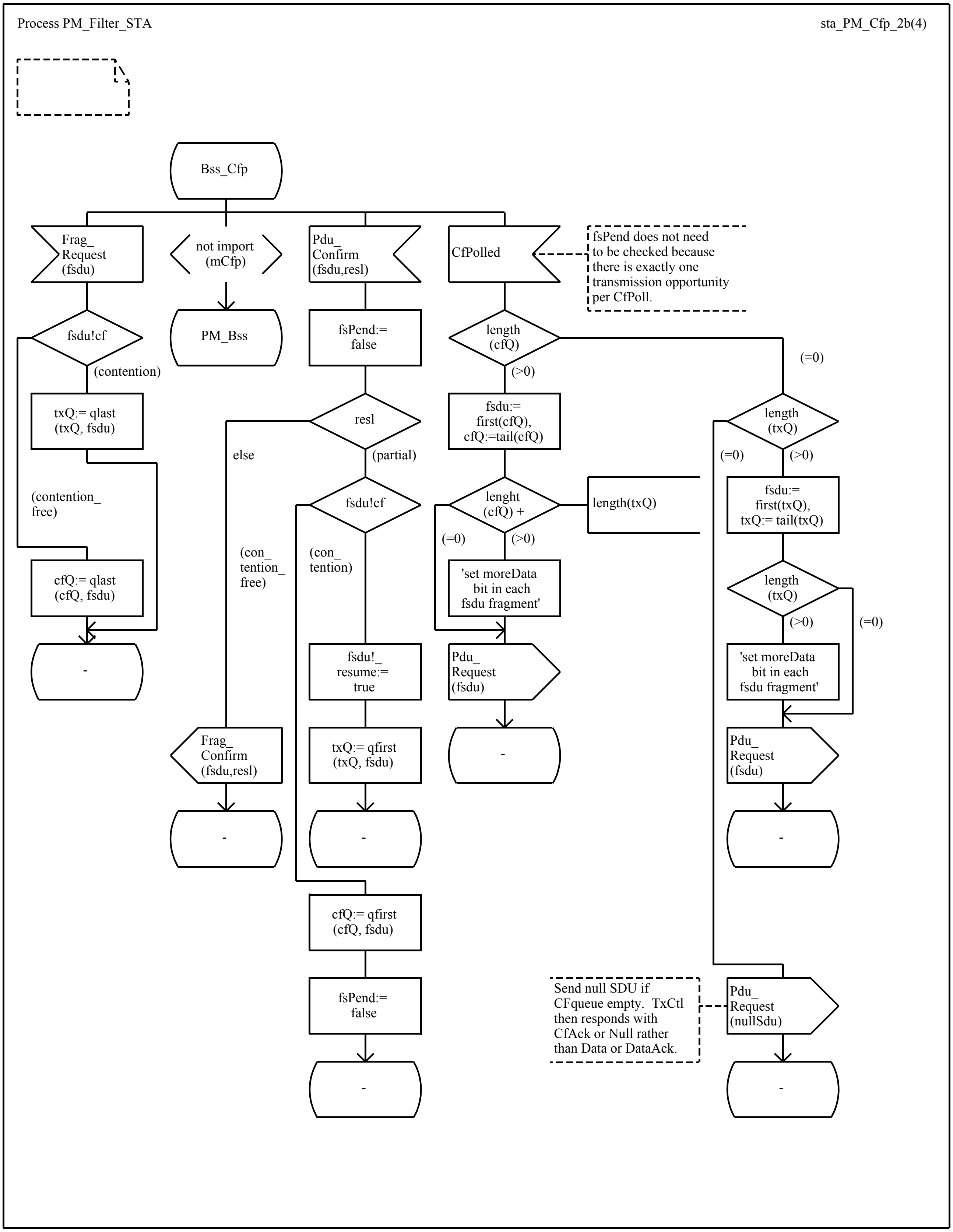

TSDU
Block MPDU_Generation_STA
sta_Mpdu_gen_1a(1)
MsduConfirm
Msdu
signal
FragConfirm(FragSdu,TxResult),
FragRequest(FragSdu) ;
MsduRequest
Includes encryption if dot11PrivacyOptionImplemented =true. This is a typical location, but implementers may use other locations between the MAC_SAP and PHY_SAP_TX as long as they provide the specified behavior as observed at LLC, MLME and the WM.
Prepare_MPDU
(1,1)
/* for STA & AP */
MmRequest
Mmpdu
MmConfirm
FragConfirm
FragMsdu
MM_
TX
/* This block converts outgoing Msdus and Mmpdus into Mpdus, fragmenting and encrypting as necessary. If the station is in a Bss, outgoing Msdus are directed via distribution service at the AP.
The PM_Filter process queues frames needing announcement by Atim in an Ibss; or frames to be sent in the CF-period at a CF-pollable station in a Bss. */
PsInquiry
FragRequest
PwrMgt
PM_Filter_STA
(1,1)
/* station version */
PsResponse,
PsChange
AtimW,
PduConfirm,
CfPolled
Mpdu
PduRequest
TPDU

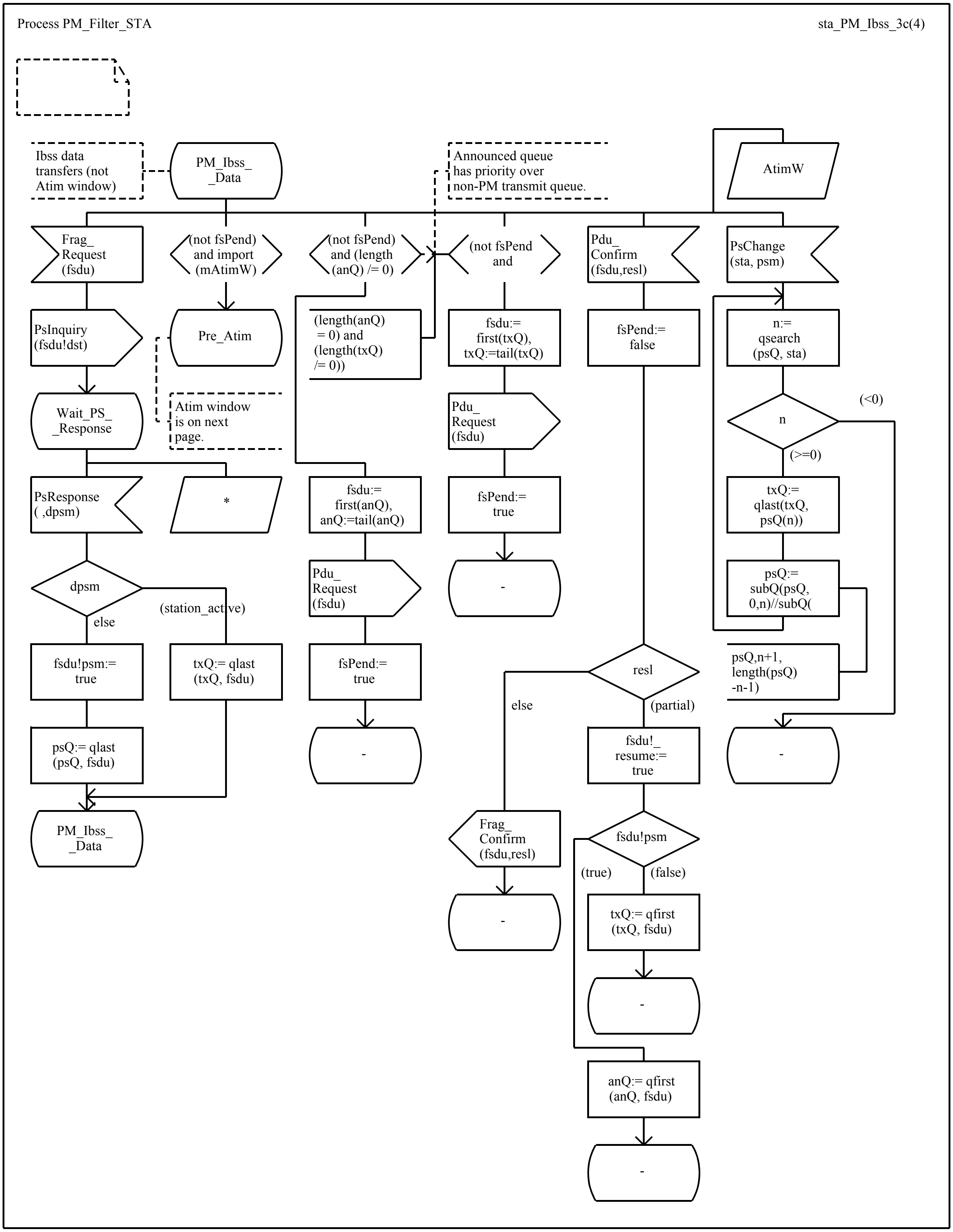

Process Prepare_MPDU
prepare_1b(2)
Encrypt
Procedure used for WEP encryption.
If dot11PrivacyOptionImplemented=
false, this procedure is not present.
dcl bcmc, keyOk,
useWep Boolean:= false ;
dcl f FragNum ;
dcl fsdu FragSdu ;
dcl mpduOvhd, p,
pduSize, thld Integer ;
dcl pri CfPriority ;
dcl rrsl TxResult ;
dcl sdu, rsdu Frame ;
imported mAssoc, mIbss, dot11PrivacyInvoked Boolean ;
imported dot11FragmentationThreshold Integer ;
imported dot11WepDefaultKeys KeyVector ;
imported dot11WepDefaultKeyId KeyIndex ;
imported dot11WepKeyMappings KeyMapArray ;
imported dot11WepKeyMappingLength KeyMapArrayLength ;
imported mCap Octetstring ;
No_Bss
import
(mAssoc)
and (not
import(mAct_
ingAsAp))
import
(mIbss)
import
(mActing_
AsAp)
Msdu_
Request
(sdu,pri)
Prepare_
_Bss
All data frames
in Bss sent to
distrib. service
Prepare_
_Ibss
All data frames
in Ibss sent to
destination sta.
Prepare_
_AP
MsduConfirm
(sdu,pri,
noBss)
not import
(mAssoc)
Msdu_
Request
(sdu,pri)
Msdu_
Request
(sdu,pri)
not import
(mIbss)
No_Bss
No_Bss
No_Bss
Msdu_
Request
(sdu,pri)
not import
(mActing_
AsAp)
No_Bss
sdu:=
setAddr1
(sdu,import
(mBssId)),
sdu:=
setToDs
(sdu,1)
sdu:=
setAddr3(sdu,
addr1(sdu)),
Mmpdus sent
even when not
in Bss/Ibss.
*
Data frames
rejected if
no Bss/Ibss.
Implementations
may retain these
frames until a
Bss becomes
(re)available.
Invoked) and
dot11Privacy_
Option_
Implemented
useWep:=
import(
dot11Privacy_
ResetMAC
Mm_
Request
(sdu,pri)
Frag_
Confirm
(fsdu,pri,rrsl)
Fragment and
encrypt is
on next page.
frag_
ment
No_Bss
bcmc:=
isGroup(
addr1(sdu))
rsdu:= substr
(fsdu!
pdus(0), 0,
sMacHdrLng),
pri:= fsdu!cf
dot11Privacy_
Option_
Implemented
and if
wepBit(sdu)=1
then true
else false fi
useWep:=
basetype
(fsdu!
pdus(0))
else
(management)
frag_
ment
Msdu_
Confirm
(rsdu,pri,rrsl)
MmConfirm
(rsdu,pri,rrsl)
to fsdu!cnfTo
/* This process generates
one or more Mpdus from
each outgoing Msdu or
Mmpdu. If encryption is
needed, the Mpdus are
encrypted before being
passed to be filtered for
possible power save or
CF queuing before tx. */
wepBit=true in
request for 3rd
frame of shared
key auth. seq.
Confirm Msdu to
MAC data service,
confirm Mmpdu to
MLME sub-block.
-

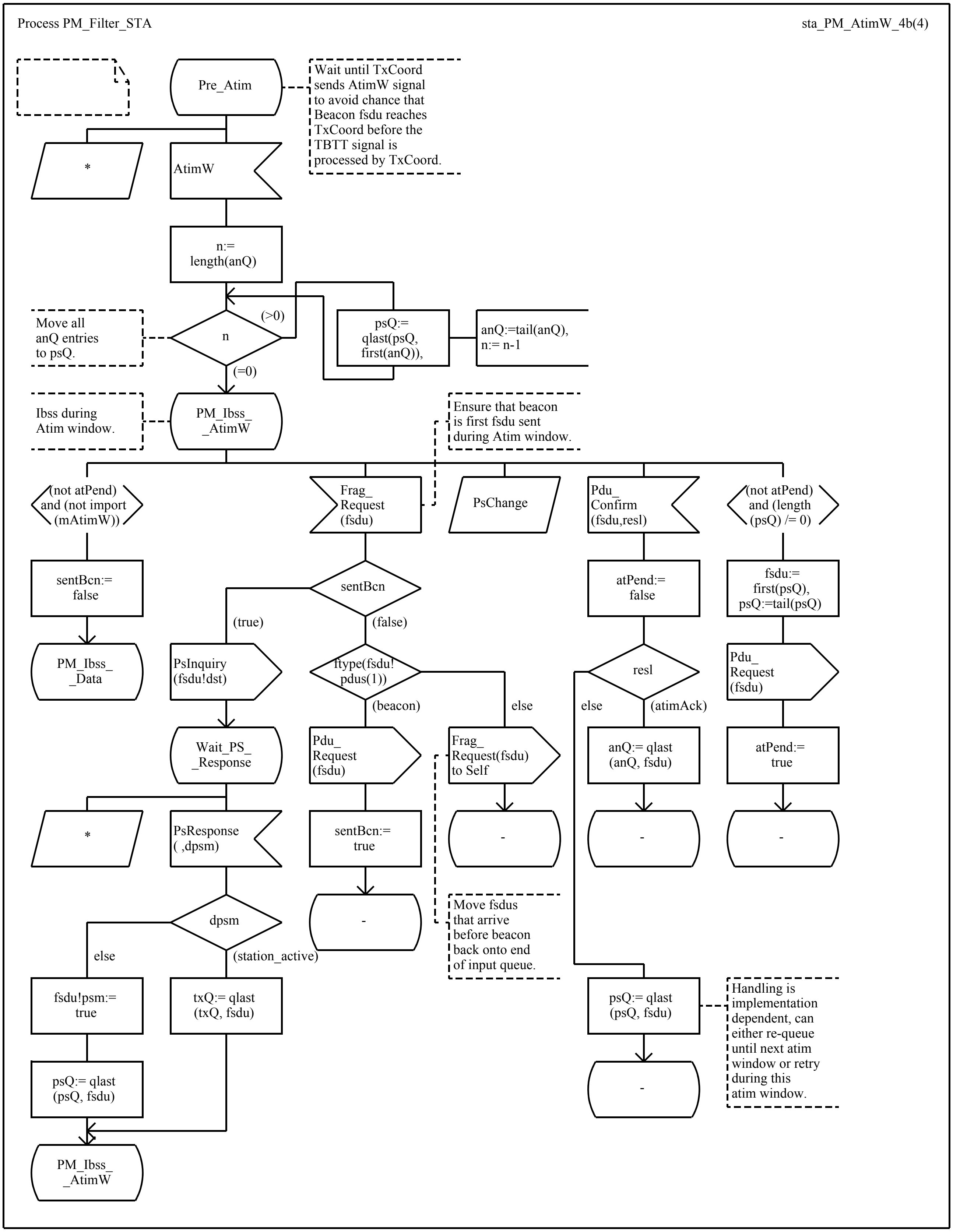

Process Prepare_MPDU
fragment_2b(2)
frag_ ment
Initialize FragSdu structure
fsdu!fTot:=1, fsdu!fCur:=0, fsdu!fAnc:=0,
fsdu!eol:=0, fsdu!sqf:=0, fsdu!src:=0, fsdu!lrc:=0, fsdu!psm:=false, fsdu!txrate:=0
fsdu!grpa:= isGrp(addr1(sdu)),
fsdu!cf:=pri, fsdu!cnfTo:=sender, fsdu!resume:=false
mpduOvhd:= sMacHdrLng + sCrcLng
Iv and Icv fields not counted in pre-fragment overhead.
dot11Fragmen_ tationThreshold must not be > dot11Max_ MpduLength.
pduSize:= import
(dot11Fragment_ ationThreshold)
not(fsdu!grpa) and
length(sdu) > pduSize
(false)
(true)
pduSize:= length(sdu) - sMacHdrLng
pduSize:= pduSize - mpduOvhd
This is the typical case, with the length of all but the last fragment equal to dot11Fragmentation_ Threshold (plus sWepAddLng if useWep=true). The value selected for pduSize must be >=256, even, and <=aMpduMaxLength.
fsdu!fTot:= ((length(sdu) - sMacHdrLng) /
pduSize) + if ((length(sdu) - sMacHdrLng) mod pduSize) /=0 then 1 else 0 fi
fsdu!fTot:=
if fsdu!fTot=0 then 1 else fsdu!fTot fi
make_ pdus
make_ pdus
f:=0, p:= sMacHdrLng
fsdu!pdus(f):= null, keyOk:=false
fsdu!pdus(f):= fsdu!pdus(f) // substr(sdu,0,
sMacHdrLng) // substr(sdu,p, pduSize)
fsdu!pdus(f):= setFrag(fsdu!pdus(f),f)
(f+1) < fsdu!fTot
(true)
(false)
fsdu!pdus(f):= setMoreFrag(fsdu!pdus(f),1)
useWep
(false)
(true)
Encrypt (fsdu!pdus(f), keyOk,
import(dot11WepKey_ Mappings), import(dot11WepKey_ MappingLength), import(dot11Wep_ DefaultKeys), import(dot11WepDefault_ KeyId), import(mCap))
keyOk
(false)
(true)
f:= f+1, p:= p+pduSize,
pduSize:= if (p+pduSize) > length(sdu) then (length(sdu) - p + 1) else pduSize fi
Final fragment may be shorter than initial/intermediate fragments.
(false)
f = fsdu!fTot
(true)
unavailable_ KeyMapping)
Msdu_ Confirm (sdu, pri,
FragRequest (fsdu)
Encryption expands each pdu by sWepAddLng, hence Mpdus may be longer than dot11MaxMpduLength by sWepAddLng.
-

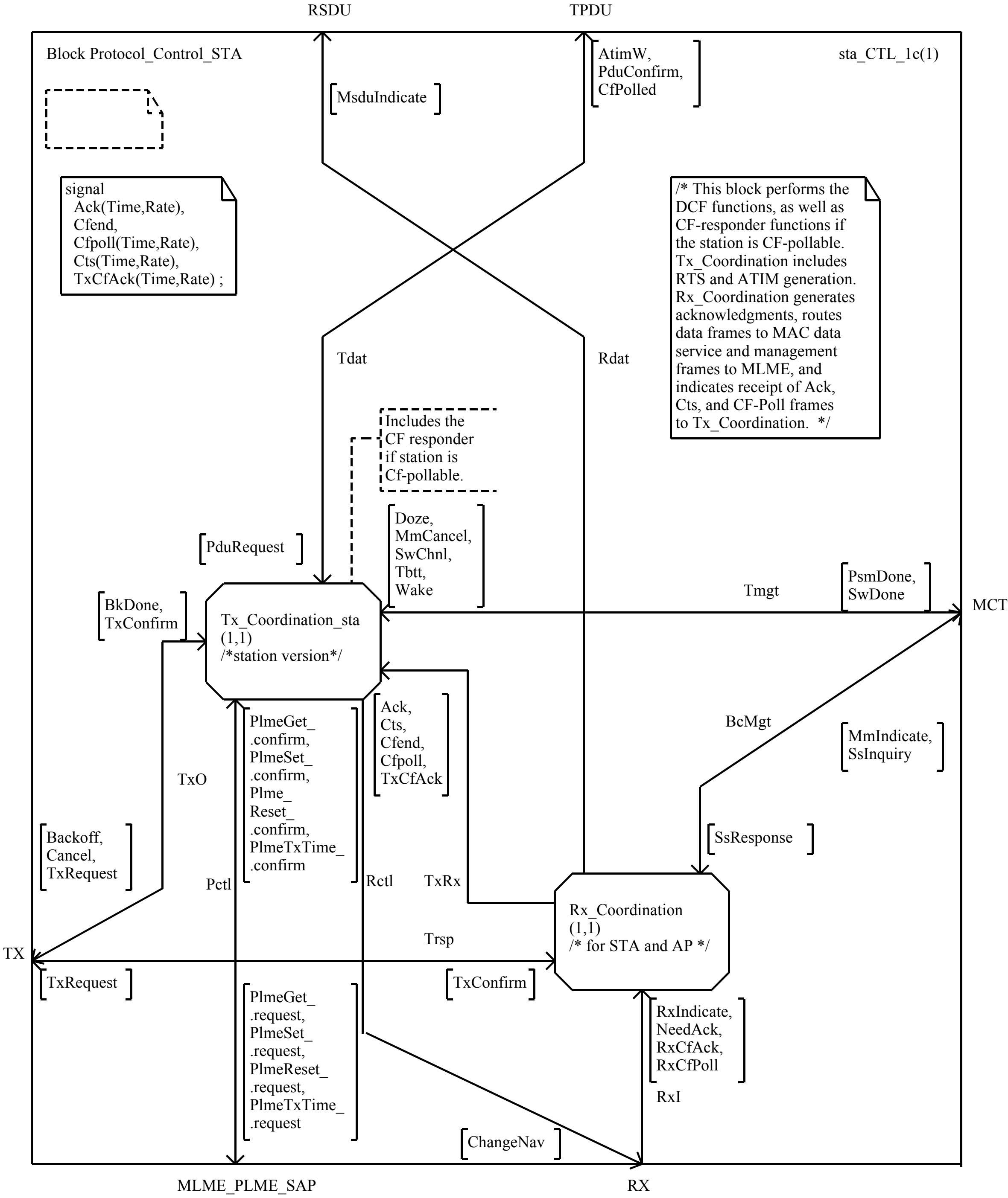

Procedure Encrypt
encrypt_1c(1)
; fpar in/out wpdu Frame,
in/out keyOk Boolean,
in maps KeyMapArray,
in mapLength KeyMapArrayLength,
in kvec KeyVector,
in kndx KeyIndex,
in caps Octetstring ;
dcl icv Crc ;
dcl encryptLng, k, n Integer ;
dcl encryptStr, newIV Octetstring ;
dcl key PrngKey ;
dcl kmap KeyMap ;
imported procedure RC4 ;
fpar PrngKey, Integer ;
returns Octetstring ;
isWds:= toDs(pdu) and frDs(pdu)
Test if addr4 field is present. Only need at AP.
Icv field is encrypted, but this length is the pre-Icv loop count.
encryptLng:= length(wpdu) - sMacHdrLng -
if isWds then sWdsAddLng else 0 fi
'newIV:= call genIV(x)'
The IV generation algorithm is not specified, but use of a new IV for each Mpdu is recommended STRONGLY.
isGrp(addr1 (wpdu))
(true)
(false)
B_S(caps) and cPrivacy
else
(=cPrivacy)
kmap:= keyLookup (addr1(wpdu),
maps, mapLength)
/* The algorithm for changing dot11WepDefaultKeyId is not specified.If all stations in the Bss have thesame values in the {relevant subsetof} dot11WepDefaultKeys, a station's DefaultKeyId algorithm does not affect interoperability. */
mappedAddr =nullAddr
(true)
(false)
Use default key if no mapping or group dest.
no_ encr
kmap! keyOn
(false)
(true)
no_ encr
If mapping keyOn=false, do not encrypt.
key:= kvec(kndx)
key:= kmap!wepKey, kndx:= 0
keyOk:= true
Return error to LLC if key is null.
key= nullKey
(false)
(true)
Concatenate key with IV for encryption PRNG seed.
key:= key // PrngKey! newIV
keyOk:= false
Use RC4 PRNG to generate an encrypt string as long as the MPDU payload plus the ICV field.
encryptStr:= call RC4 (key,
encryptLng+ sCrcLng)
wpdu:= substr(wpdu,0, sMacHdrLng)
// newIV // O1 // substr(wpdu, sMac_ HdrLng, encryptLng)
Insert IV and keyId between MAC header and data field.
wpdu:= setKeyId (wpdu,kndx)
en_ cipher
en_ cipher
icv:= initCrc
if isWds then sWdsAddLng else 0 fi
k:= 0, n:= sWepHdrLng +
ICV value calculated from plaintext.
icv:= crc32 (icv,wpdu(n))
Encrypt by xor of payload with encrypt string.
wpdu(n):= wpdu(n) xor encryptStr(k)
k:= k+1, n:= n+1
k = (encryptLng)
(false)
(true)
n:= 0
raw ICV is 1's complement of crc32, MSb-first
icv:= mirror(not(icv))
(icv(n) xor encryptStr(k))
wpdu:= wpdu //
Encrypt ICV octets and attach to end of Mpdu.
k:= k+1, n:= n+1
n = sCrcLng
(false)
(true)
Set WEP bit in Frame Control field.
wepdu:= setWepBit (wepdu,1),
keyOk:= true

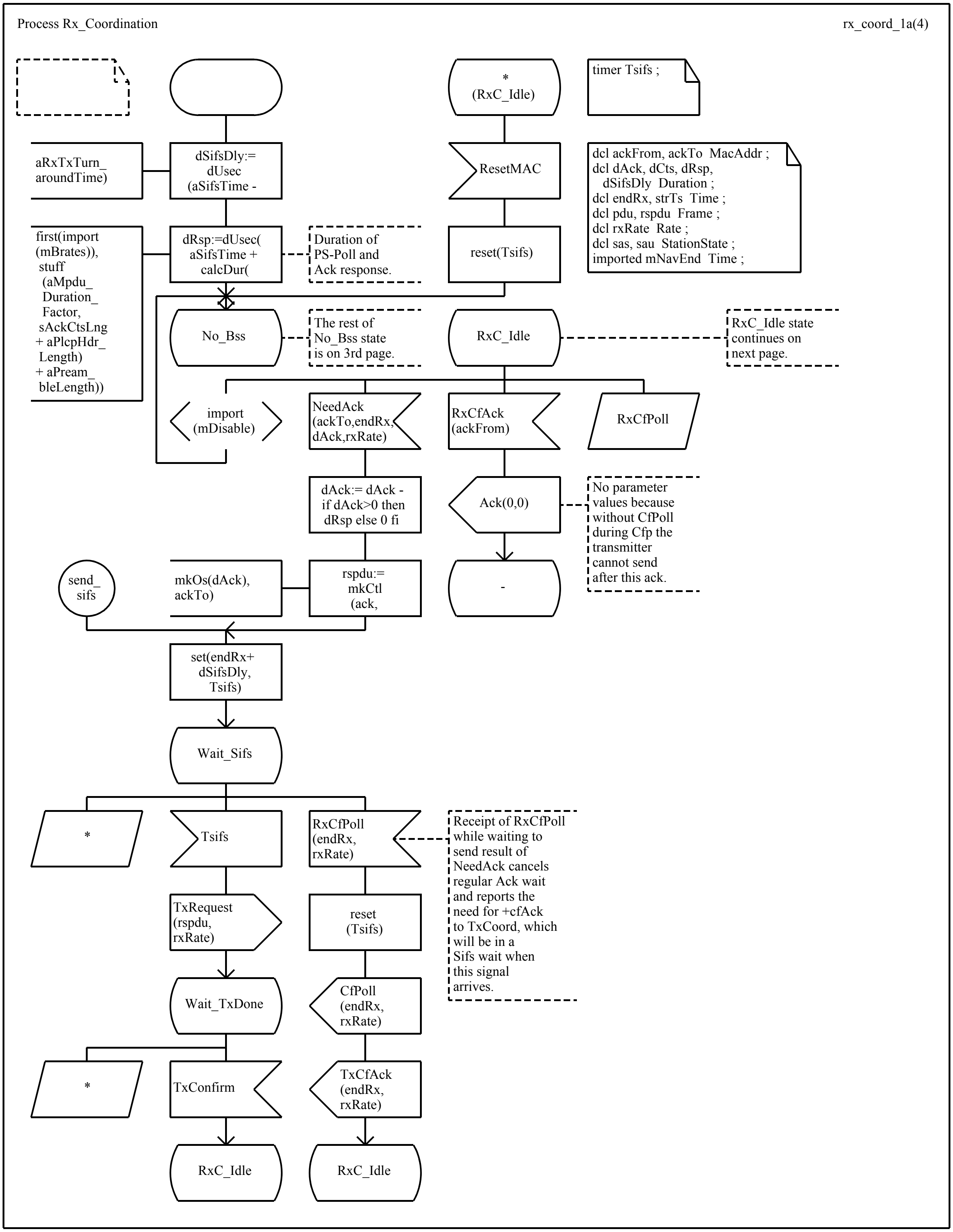

Process PM_Filter_STA
sta_PM_Bss_1b(4)
*
ResetMAC
import (mDisable)
dcl atPend, fsPend, sentBcn Boolean:= false ;
dcl cfQ, psQ, txQ, anQ SduQueue:= emptyQ ;
dcl dpsm PsMode ;
dcl fsdu, rsdu FragSdu ;
dcl k, n Integer ;
dcl resl TxResult ;
dcl sta MacAddr ;
anQ:=emptyQ, cfQ:=emptyQ,
psQ:=emptyQ, txQ:=emptyQ
PM_Idle
Station not in any BSS, only Mmpdus will be sent down by Prepare_MPDU.
import (mAssoc)
import (mIbss)
Frag_ Request (fsdu)
Pdu_ Confirm (fsdu,resl)
PM_Bss
PsChange ignored when assoc w/BSS.
PM_Ibss_ _Data
IBSS case is two pages ahead.
Pdu_ Request (fsdu)
Frag_ Confirm (fsdu,resl)
-
-
Pass management frames involved in scan, join, and start.
Frag_ Request (fsdu)
(not fsPend) and (length (txQ) /= 0)
Pdu_ Confirm (fsdu,resl)
import (mCfp)
fsdu!cf
(contention)
fsdu:=first(txQ), txQ:=tail(txQ)
fsPend:= false
Bss_Cfp
Cfp handling is on next page.
txQ:= qlast (txQ, fsdu)
Pdu_ Request (fsdu)
resl
else
(partial)
(contention_ Free)
fsPend:= true
fsdu!_ resume:= true
cfQ:= qlast (cfQ, fsdu)
-
txQ:= qfirst (txQ, fsdu)
-
-
Frag_ Confirm (fsdu,resl)
-

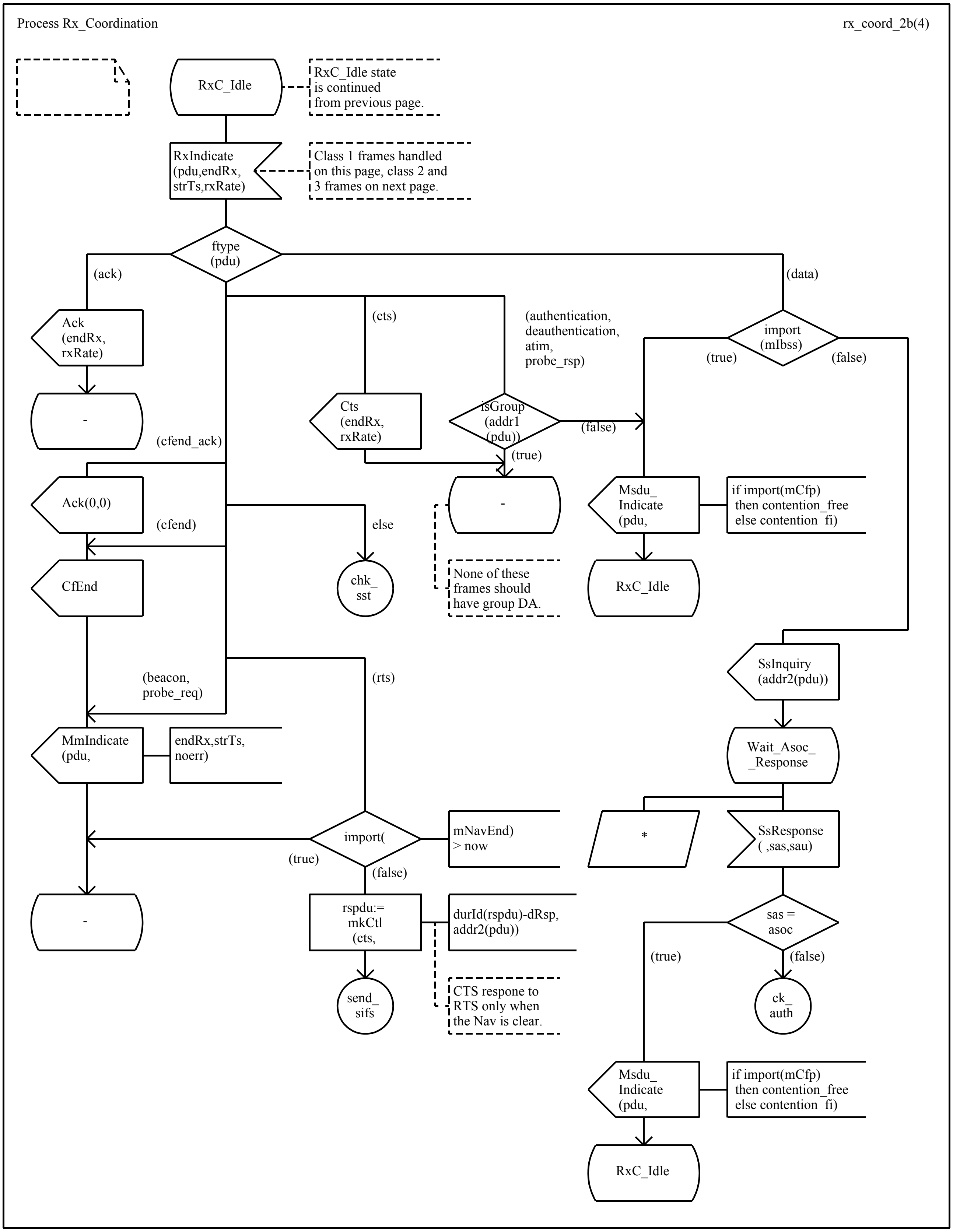

Process PM_Filter_STA
sta_PM_Cfp_2b(4)
Bss_Cfp
Frag_ Request (fsdu)
not import (mCfp)
Pdu_ Confirm (fsdu,resl)
CfPolled
fsPend does not need to be checked because there is exactly one transmission opportunity per CfPoll.
fsdu!cf
(contention)
(contention_ free)
txQ:= qlast (txQ, fsdu)
cfQ:= qlast (cfQ, fsdu)
-
PM_Bss
fsPend:= false
resl
else
(partial)
fsdu!cf
(con_ tention_ free)
(con_ tention)
fsdu!_ resume:= true
txQ:= qfirst (txQ, fsdu)
Frag_ Confirm (fsdu,resl)
-
-
cfQ:= qfirst (cfQ, fsdu)
fsPend:= false
-
length (cfQ)
(>0)
(=0)
fsdu:= first(cfQ), cfQ:=tail(cfQ)
lenght (cfQ) +
length(txQ)
(=0)
(>0)
'set moreData bit in each fsdu fragment'
Pdu_ Request (fsdu)
-
length (txQ)
(=0)
(>0)
fsdu:= first(txQ), txQ:= tail(txQ)
length (txQ)
(>0)
(=0)
'set moreData bit in each fsdu fragment'
Pdu_ Request (fsdu)
-
Send null SDU if CFqueue empty. TxCtl then responds with CfAck or Null rather than Data or DataAck.
Pdu_ Request (nullSdu)
-

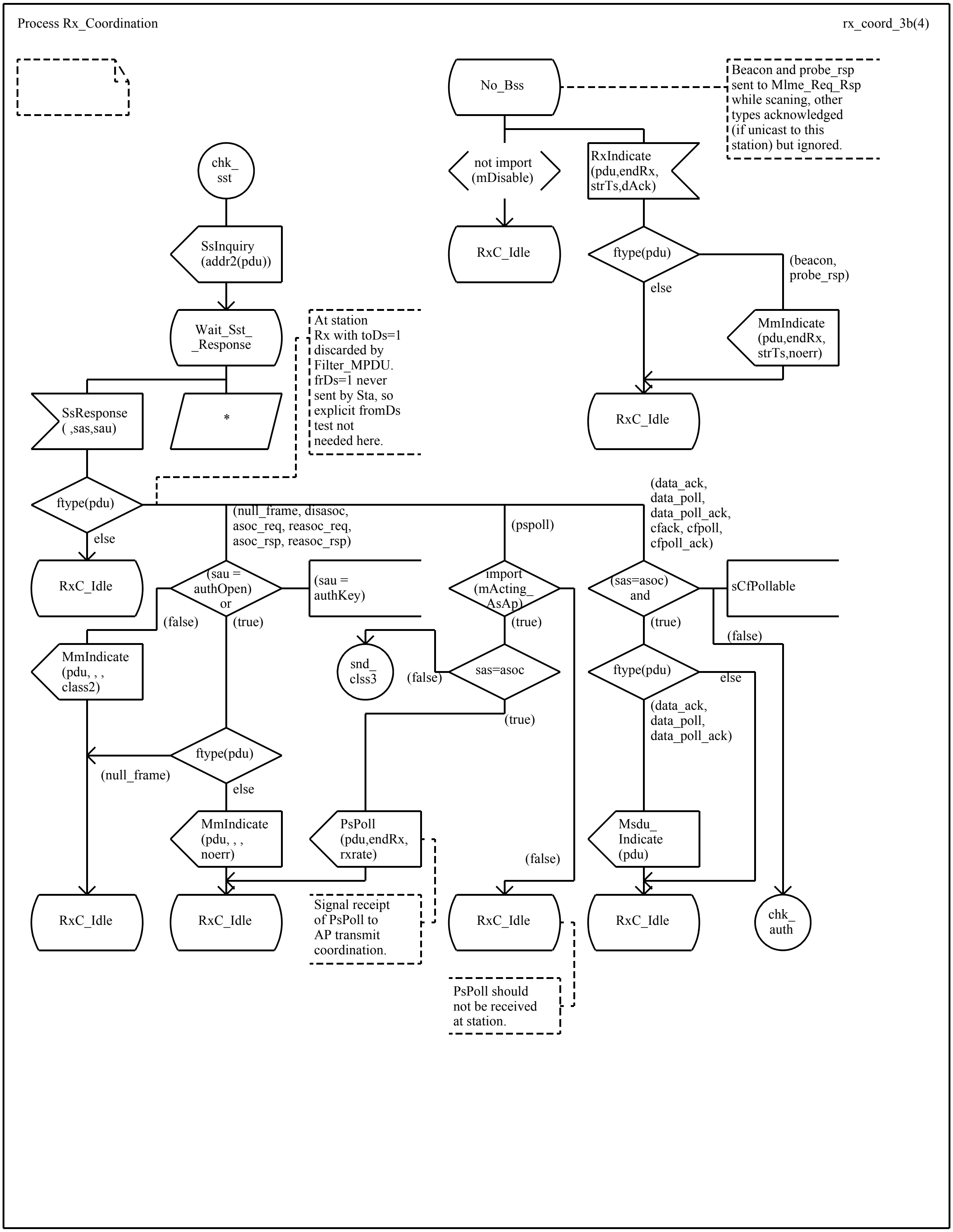

Process PM_Filter_STA
sta_PM_Ibss_3c(4)
Ibss data transfers (not Atim window)
PM_Ibss_ _Data
Announced queue has priority over non-PM transmit queue.
AtimW
Frag_ Request (fsdu)
(not fsPend) and import (mAtimW)
(not fsPend) and (length (anQ) /= 0)
(not fsPend and
Pdu_ Confirm (fsdu,resl)
PsChange (sta, psm)
PsInquiry (fsdu!dst)
Pre_Atim
(length(anQ) = 0) and (length(txQ) /= 0))
fsdu:= first(txQ), txQ:=tail(txQ)
fsPend:= false
n:= qsearch (psQ, sta)
Wait_PS_ _Response
Atim window is on next page.
Pdu_ Request (fsdu)
n
(<0)
(>=0)
PsResponse (,dpsm)
*
fsdu:= first(anQ), anQ:=tail(anQ)
fsPend:= true
txQ:= qlast(txQ, psQ(n))
dpsm
(station_active)
else
Pdu_ Request (fsdu)
-
psQ:= subQ(psQ, 0,n)//subQ(
fsdu!psm:= true
txQ:= qlast (txQ, fsdu)
fsPend:= true
resl
psQ,n+1, length(psQ) -n-1)
else
(partial)
psQ:= qlast (psQ, fsdu)
-
fsdu!_ resume:= true
-
PM_Ibss_ _Data
Frag_ Confirm (fsdu,resl)
fsdu!psm
(true)
(false)
-
txQ:= qfirst (txQ, fsdu)
-
anQ:= qfirst (anQ, fsdu)
-

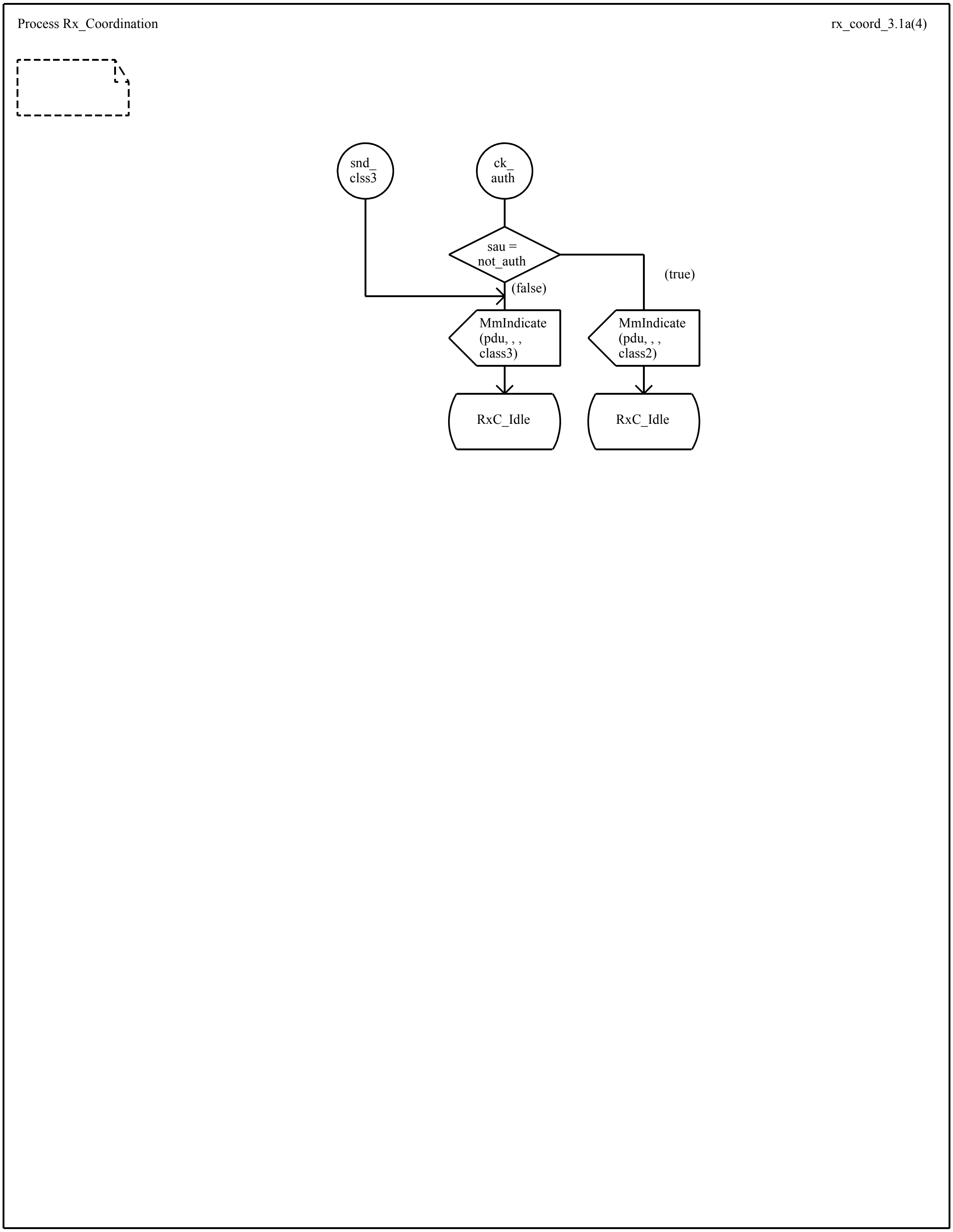

Process PM_Filter_STA
sta_PM_AtimW_4b(4)
Pre_Atim
Wait until TxCoord sends AtimW signal to avoid chance that Beacon fsdu reaches TxCoord before the TBTT signal is processed by TxCoord.
*
AtimW
n:= length(anQ)
Move all anQ entries to psQ.
n
(>0)
(=0)
psQ:= qlast(psQ, first(anQ)),
anQ:=tail(anQ), n:= n-1
Ibss during Atim window.
PM_Ibss_ _AtimW
Ensure that beacon is first fsdu sent during Atim window.
(not atPend) and (not import (mAtimW))
Frag_ Request (fsdu)
PsChange
Pdu_ Confirm (fsdu,resl)
(not atPend) and (length (psQ) /= 0)
sentBcn:= false
PM_Ibss_ _Data
sentBcn
(true)
(false)
PsInquiry (fsdu!dst)
Wait_PS_ _Response
*
PsResponse (,dpsm)
dpsm
else
(station_active)
fsdu!psm:= true
txQ:= qlast (txQ, fsdu)
psQ:= qlast (psQ, fsdu)
PM_Ibss_ _AtimW
ftype(fsdu! pdus(1))
(beacon)
else
Pdu_ Request (fsdu)
sentBcn:= true
-
Frag_ Request(fsdu) to Self
-
Move fsdus that arrive before beacon back onto end of input queue.
atPend:= false
resl
else
(atimAck)
anQ:= qlast (anQ, fsdu)
-
psQ:= qlast (psQ, fsdu)
-
Handling is implementation dependent, can either re-queue until next atim window or retry during this atim window.
fsdu:= first(psQ), psQ:=tail(psQ)
Pdu_ Request (fsdu)
atPend:= true
-

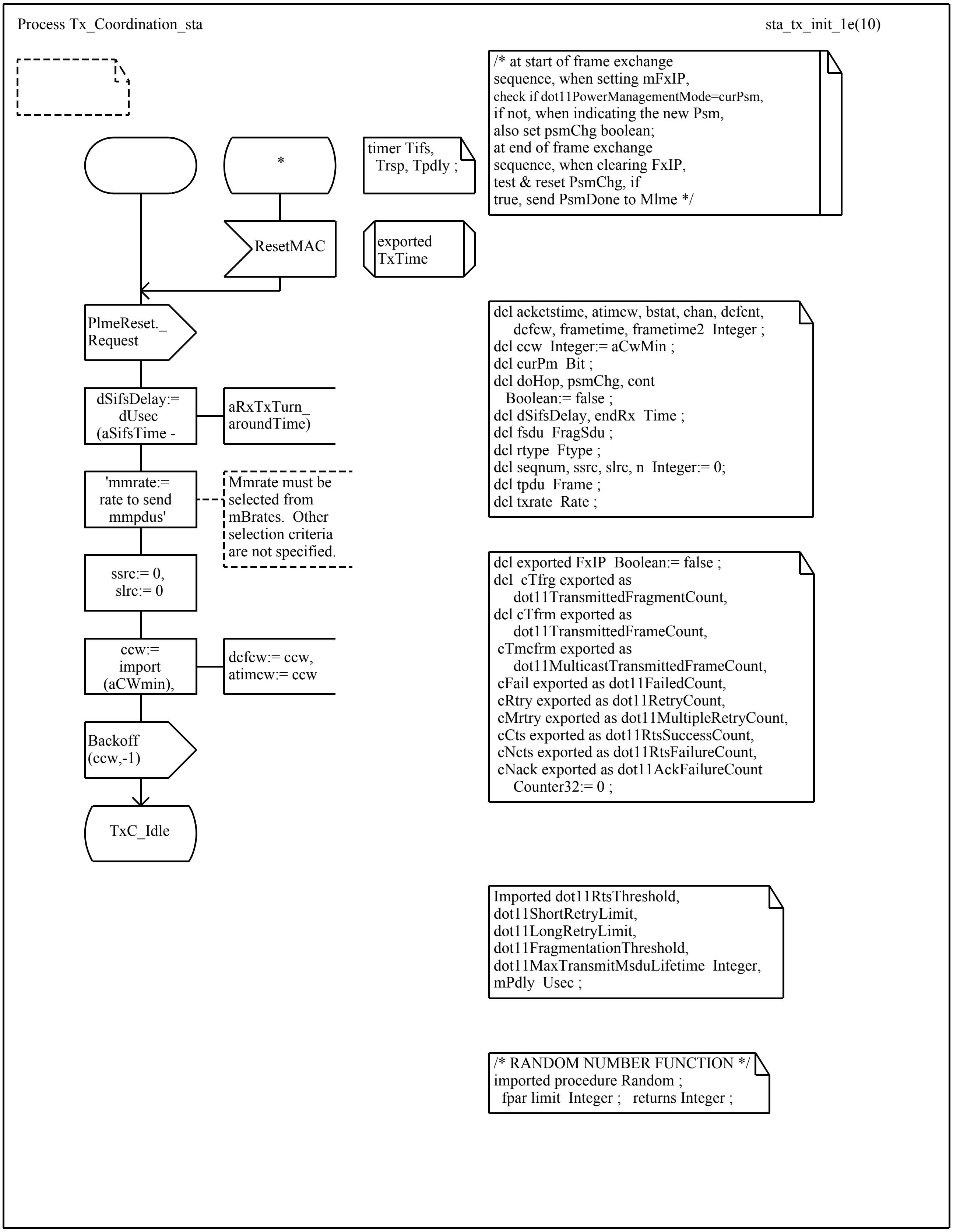

RSDU
TPDU
Block Protocol_Control_STA
sta_CTL_1c(1)
MsduIndicate
AtimW,
PduConfirm,
CfPolled
signal
Ack(Time,Rate),
Cfend,
Cfpoll(Time,Rate),
Cts(Time,Rate),
TxCfAck(Time,Rate) ;
/* This block performs the DCF functions, as well as CF-responder functions if the station is CF-pollable. Tx_Coordination includes RTS and ATIM generation. Rx_Coordination generates acknowledgments, routes data frames to MAC data service and management frames to MLME, and indicates receipt of Ack, Cts, and CF-Poll frames to Tx_Coordination. */
Tdat
Rdat
Includes the CF responder if station is Cf-pollable.
PduRequest
Doze,
MmCancel,
SwChnl,
Tbtt,
Wake
Tmgt
PsmDone,
SwDone
MCTL
BkDone,
TxConfirm
Tx_Coordination_sta
(1,1)
/*station version*/
Ack,
Cts,
Cfend,
Cfpoll,
TxCfAck
BcMgt
MmIndicate,
SsInquiry
PlmeGet_
.confirm,
PlmeSet_
.confirm,
Plme_
Reset_
.confirm,
PlmeTxTime_
.confirm
TxO
Backoff,
Cancel,
TxRequest
SsResponse
Pctl
Rctl
TxRx
Rx_Coordination
(1,1)
/* for STA and AP */
Trsp
TX
TxRequest
TxConfirm
PlmeGet_
.request,
PlmeSet_
.request,
PlmeReset_
.request,
PlmeTxTime_
.request
RxIndicate,
NeedAck,
RxCfAck,
RxCfPoll
RxI
ChangeNav
MLME_PLME_SAP
RX


Process Rx_Coordination
rx_coord_1a(4)
*
(RxC_Idle)
timer Tsifs ;
aRxTxTurn_
aroundTime)
dSifsDly:=
dUsec
(aSifsTime -
ResetMAC
dcl ackFrom, ackTo MacAddr ;
dcl dAck, dCts, dRsp,
dSifsDly Duration ;
dcl endRx, strTs Time ;
dcl pdu, rspdu Frame ;
dcl rxRate Rate ;
dcl sas, sau StationState ;
imported mNavEnd Time ;
first(import
(mBrates)),
stuff
(aMpdu_
Duration_
Factor,
sAckCtsLng
+ aPlcpHdr_
Length)
+ aPream_
bleLength))
dRsp:=dUsec(
aSifsTime +
calcDur(
Duration of
PS-Poll and
Ack response.
reset(Tsifs)
No_Bss
The rest of
No_Bss state
is on 3rd page.
RxC_Idle
RxC_Idle state
continues on
next page.
import
(mDisable)
NeedAck
(ackTo,endRx,
dAck,rxRate)
RxCfAck
(ackFrom)
RxCfPoll
dAck:= dAck -
if dAck>0 then
dRsp else 0 fi
Ack(0,0)
No parameter
values because
without CfPoll
during Cfp the
transmitter
cannot send
after this ack.
send_
sifs
mkOs(dAck),
ackTo)
rspdu:=
mkCtl
(ack,
-
set(endRx+
dSifsDly,
Tsifs)
Wait_Sifs
*
Tsifs
RxCfPoll
(endRx,
rxRate)
Receipt of RxCfPoll
while waiting to
send result of
NeedAck cancels
regular Ack wait
and reports the
need for +cfAck
to TxCoord, which
will be in a
Sifs wait when
this signal
arrives.
TxRequest
(rspdu,
rxRate)
reset
(Tsifs)
Wait_TxDone
CfPoll
(endRx,
rxRate)
*
TxConfirm
TxCfAck
(endRx,
rxRate)
RxC_Idle
RxC_Idle


Process Rx_Coordination
rx_coord_2b(4)
RxC_Idle
RxC_Idle state
is continued
from previous page.
RxIndicate
(pdu,endRx,
strTs,rxRate)
Class 1 frames handled
on this page, class 2 and
3 frames on next page.
ftype
(pdu)
(ack)
(data)
Ack
(endRx,
rxRate)
(cts)
(authentication,
deauthentication,
atim,
probe_rsp)
import
(mIbss)
(true)
(false)
-
(cfend_ack)
Cts
(endRx,
rxRate)
isGroup
(addr1
(pdu))
(false)
(true)
Ack(0,0)
(cfend)
else
-
Msdu_
Indicate
(pdu,
if import(mCfp)
then contention_free
else contention fi)
CfEnd
chk_
sst
None of these
frames should
have group DA.
RxC_Idle
(beacon,
probe_req)
(rts)
SsInquiry
(addr2(pdu))
MmIndicate
(pdu,
endRx,strTs,
noerr)
Wait_Asoc_
_Response
import(
mNavEnd)
> now
(true)
(false)
*
SsResponse
(,sas,sau)
-
rspdu:=
mkCtl
(cts,
durId(rspdu)-dRsp,
addr2(pdu))
sas =
asoc
(true)
(false)
send_
sifs
CTS respone to
RTS only when
the Nav is clear.
ck_
auth
Msdu_
Indicate
(pdu,
if import(mCfp)
then contention_free
else contention fi)
RxC_Idle


Process Rx_Coordination
rx_coord_3b(4)
No_Bss
Beacon and probe_rsp sent to Mlme_Req_Rsp while scaning, other types acknowledged (if unicast to this station) but ignored.
not import (mDisable)
RxIndicate (pdu,endRx, strTs,dAck)
RxC_Idle
ftype(pdu)
(beacon, probe_rsp)
else
MmIndicate (pdu,endRx, strTs,noerr)
RxC_Idle
chk_ sst
SsInquiry (addr2(pdu))
Wait_Sst_ _Response
At station Rx with toDs=1 discarded by Filter_MPDU. frDs=1 never sent by Sta, so explicit fromDs test not needed here.
SsResponse (,sas,sau)
*
ftype(pdu)
else
RxC_Idle
(null_frame, disasoc, asoc_req, reasoc_req, asoc_rsp, reasoc_rsp)
(pspoll)
(data_ack, data_poll, data_poll_ack, cfack, cfpoll, cfpoll_ack)
(sau = authOpen) or
(sau = authKey)
import (mActing_ AsAp)
(sas=asoc) and
sCfPollable
(false)
(true)
(true)
(true)
(false)
MmIndicate (pdu, , , class2)
snd_ clss3
(false)
sas=asoc
ftype(pdu)
else
(true)
(data_ack, data_poll, data_poll_ack)
ftype(pdu)
(null_frame)
else
MmIndicate (pdu, , , noerr)
PsPoll (pdu,endRx, rxrate)
(false)
Msdu_ Indicate (pdu)
RxC_Idle
RxC_Idle
Signal receipt of PsPoll to AP transmit coordination.
RxC_Idle
RxC_Idle
chk_ auth
PsPoll should not be received at station.


Process Rx_Coordination
rx_coord_3.1a(4)
snd_
clss3
ck_
auth
sau =
not_auth
(false)
(true)
MmIndicate
(pdu, , ,
class3)
MmIndicate
(pdu, , ,
class2)
RxC_Idle
RxC_Idle



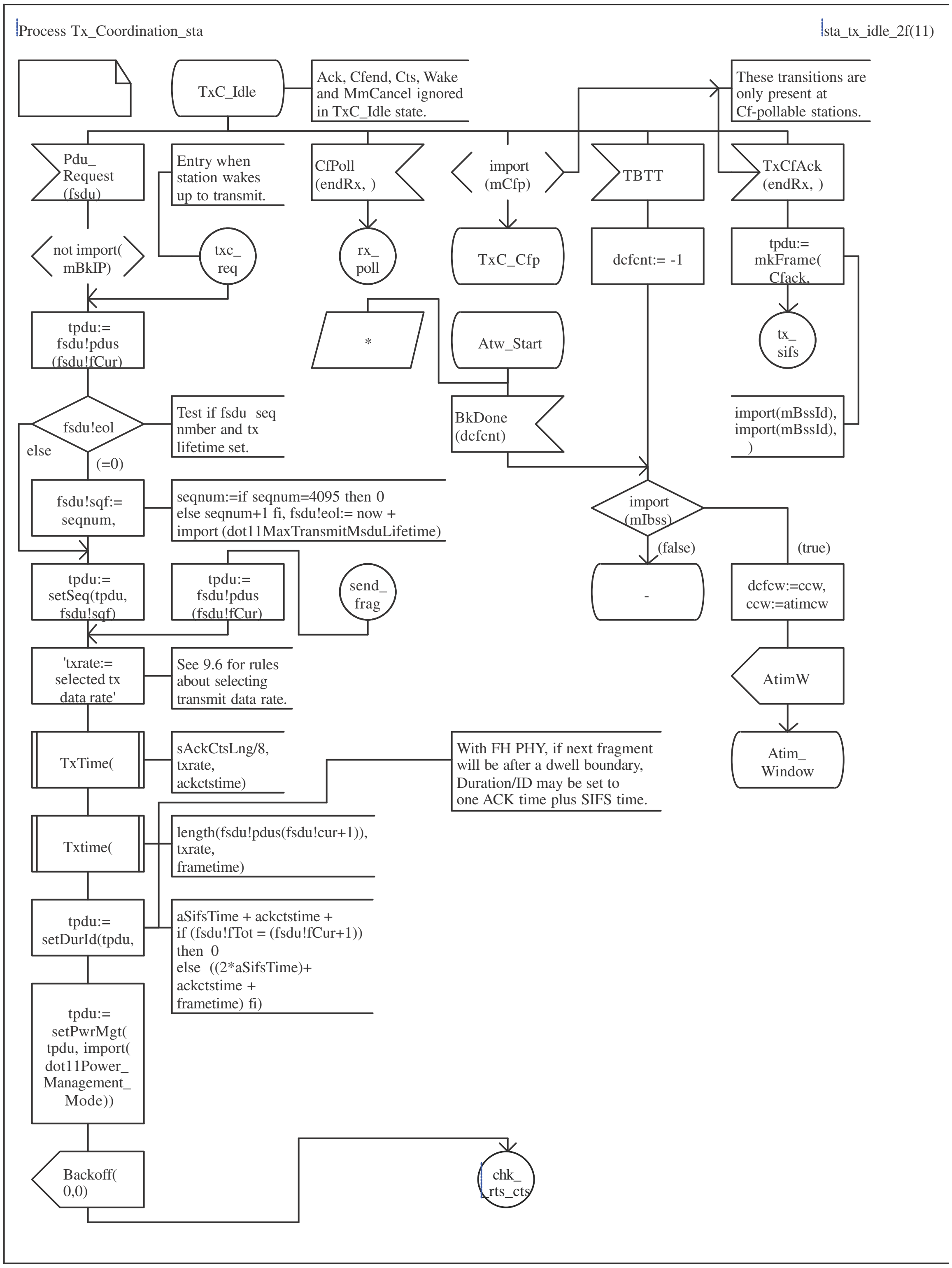
```
/* at start of frame exchange
sequence, when setting mFxIP,
check if dot11PowerManagementMode=curPsm,
if not, when indicating the new Psm,
also set psmChg boolean;
at end of frame exchange
sequence, when clearing FxIP,
test & reset PsmChg, if
true, send PsmDone to Mlme */
```

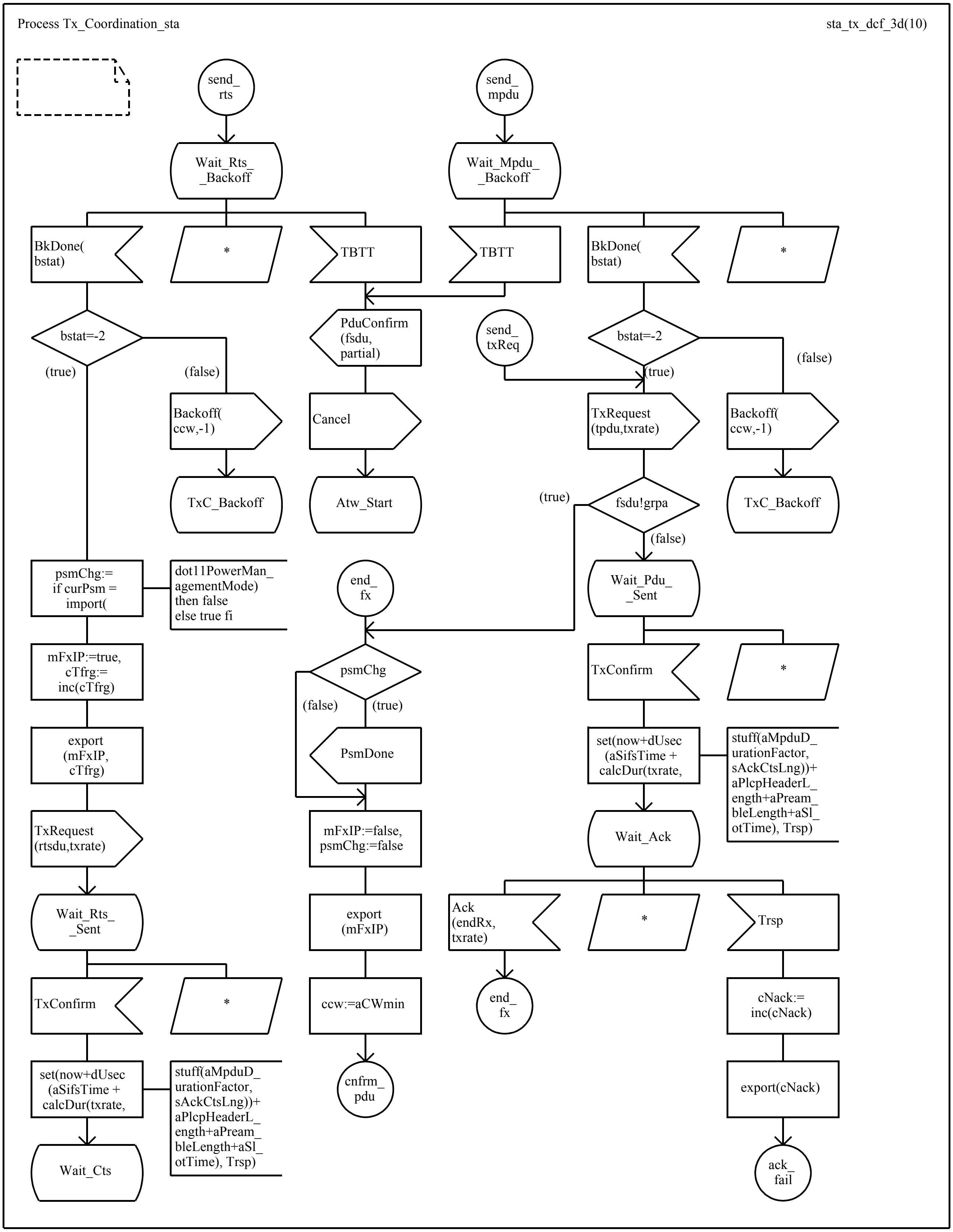
```
dcl ackctstime, atimcw, bstat, chan, dcfcnt,
    dcfcw, frametime, frametime2  Integer ;
dcl ccw  Integer:= aCwMin ;
dcl curPm  Bit ;
dcl doHop, psmChg, cont
  Boolean:= false ;
dcl dSifsDelay, endRx  Time ;
dcl fsdu  FragSdu ;
dcl rtype  Ftype ;
dcl seqnum, ssrc, slrc, n  Integer:= 0;
dcl tpdu  Frame ;
dcl txrate  Rate ;
```

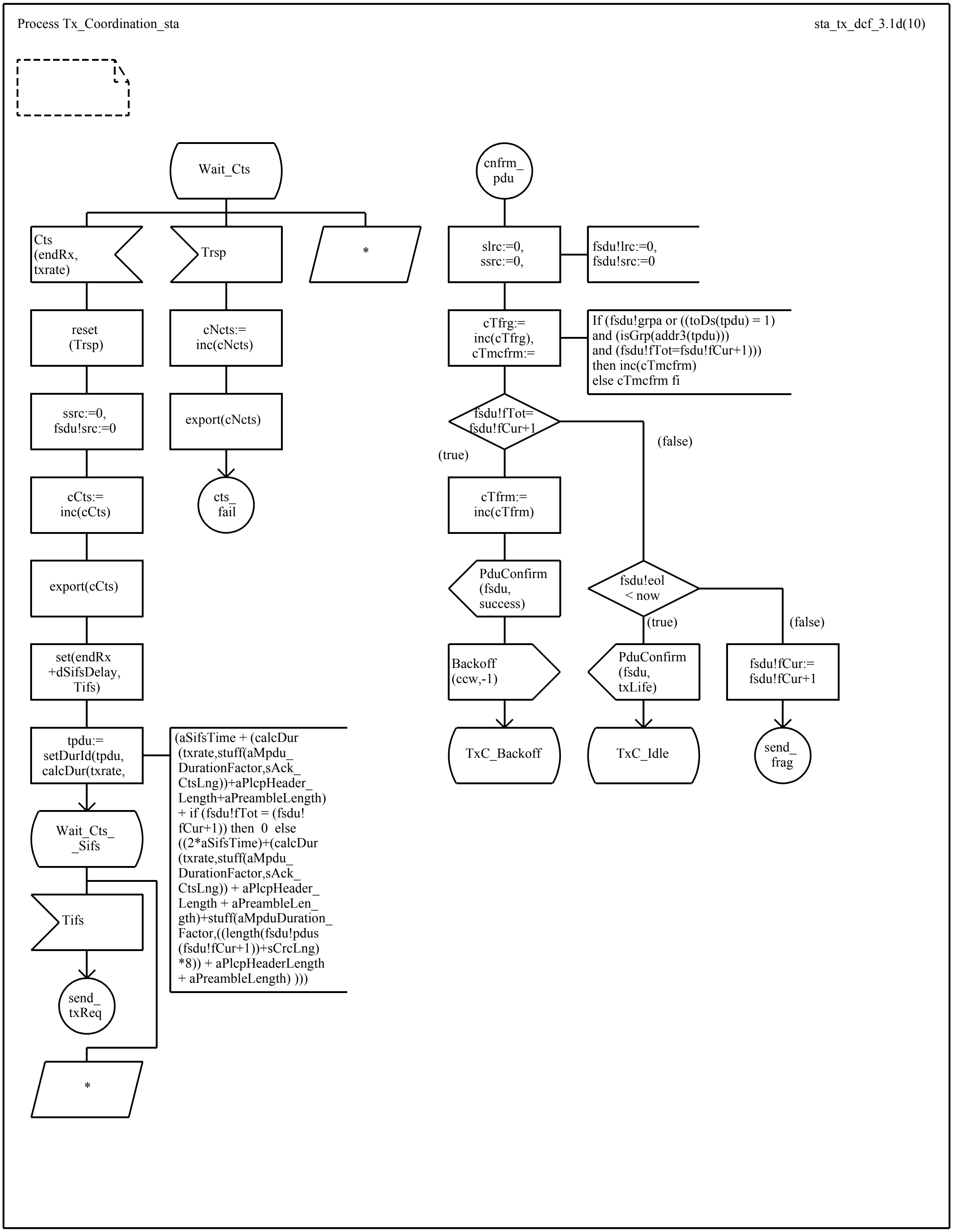
```
dcl exported FxIP  Boolean:= false ;
dcl  cTfrg exported as
    dot11TransmittedFragmentCount,
dcl cTfrm exported as
    dot11TransmittedFrameCount,
 cTmcfrm exported as
    dot11MulticastTransmittedFrameCount,
 cFail exported as dot11FailedCount,
 cRtry exported as dot11RetryCount,
 cMrtry exported as dot11MultipleRetryCount,
 cCts exported as dot11RtsSuccessCount,
 cNcts exported as dot11RtsFailureCount,
 cNack exported as dot11AckFailureCount
    Counter32:= 0 ;
```

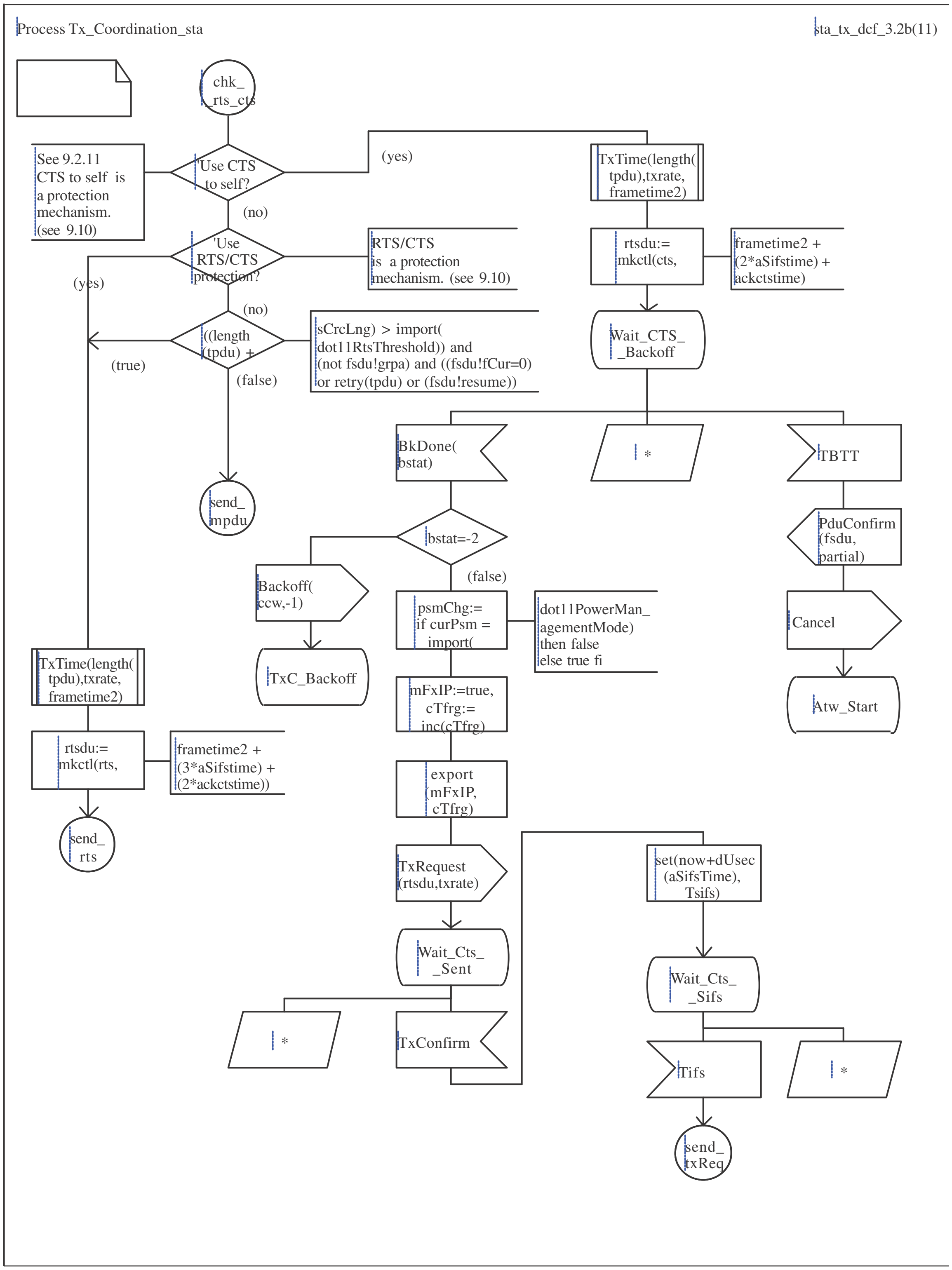
```
Imported dot11RtsThreshold,
dot11ShortRetryLimit,
dot11LongRetryLimit,
dot11FragmentationThreshold,
dot11MaxTransmitMsduLifetime  Integer,
mPdly  Usec ;
```

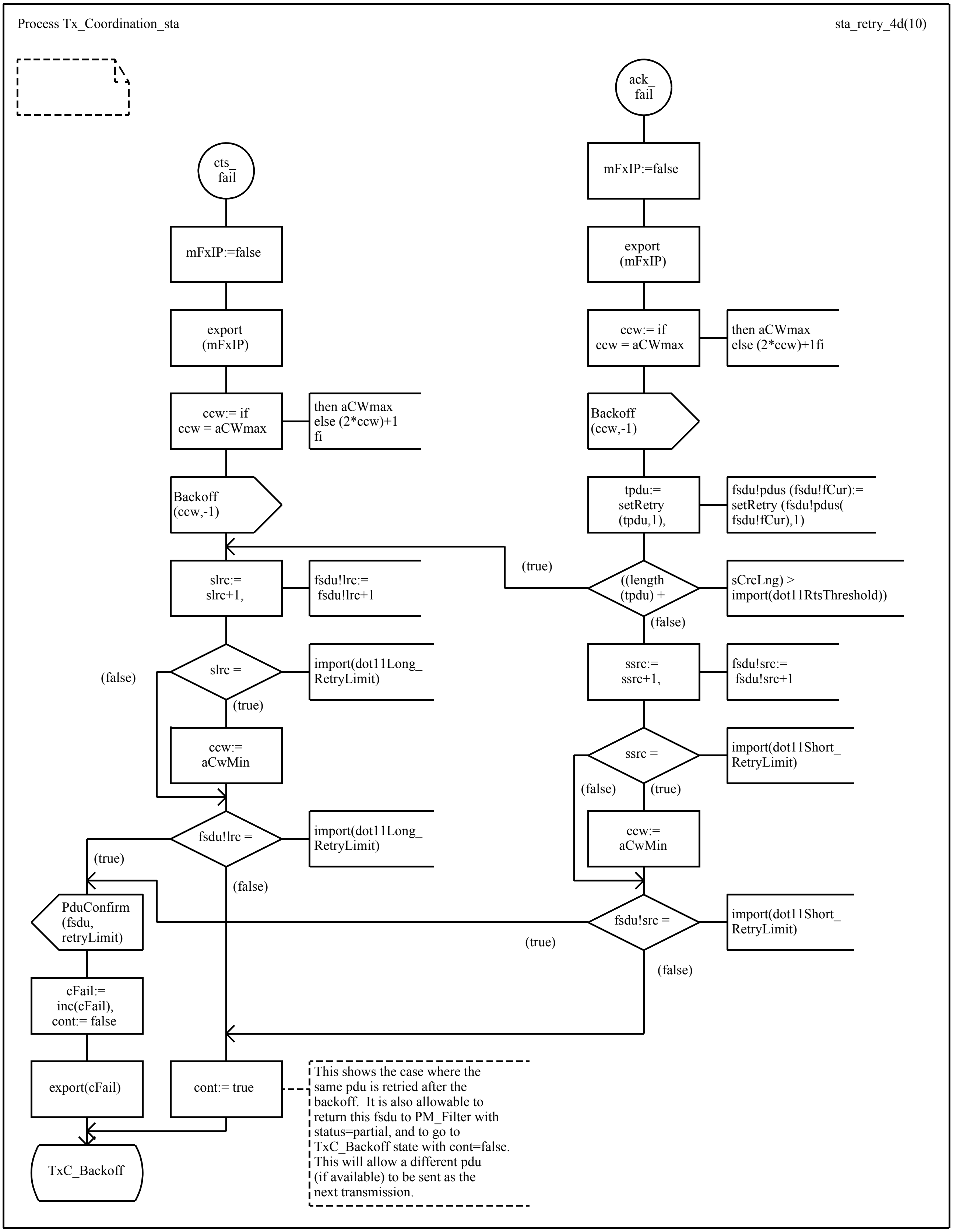
```
/* RANDOM NUMBER FUNCTION */
imported procedure Random ;
  fpar limit  Integer ;   returns Integer ;
```


Process Tx_Coordination_sta
sta_tx_idle_2f(11)
TxC_Idle
Ack, Cfend, Cts, Wake and MmCancel ignored in TxC_Idle state.
These transitions are only present at Cf-pollable stations.
Pdu_ Request (fsdu)
Entry when station wakes up to transmit.
CfPoll (endRx,)
import (mCfp)
TBTT
TxCfAck (endRx,)
not import(mBkIP)
txc_ req
rx_ poll
TxC_Cfp
dcfcnt:= -1
tpdu:= mkFrame(Cfack,
tpdu:= fsdu!pdus (fsdu!fCur)
*
Atw_Start
tx_ sifs
fsdu!eol
Test if fsdu seq nmber and tx lifetime set.
BkDone (dcfcnt)
import(mBssId), import(mBssId),)
else
(=0)
fsdu!sqf:= seqnum,
seqnum:=if seqnum=4095 then 0 else seqnum+1 fi, fsdu!eol:= now + import (dot11MaxTransmitMsduLifetime)
import (mIbss)
(false)
(true)
tpdu:= setSeq(tpdu, fsdu!sqf)
tpdu:= fsdu!pdus (fsdu!fCur)
send_ frag
-
dcfcw:=ccw, ccw:=atimcw
'txrate:= selected tx data rate'
See 9.6 for rules about selecting transmit data rate.
AtimW
TxTime(
sAckCtsLng/8, txrate, ackctstime)
With FH PHY, if next fragment will be after a dwell boundary, Duration/ID may be set to one ACK time plus SIFS time.
Atim_ Window
Txtime(
length(fsdu!pdus(fsdu!cur+1)), txrate, frametime)
tpdu:= setDurId(tpdu,
aSifsTime + ackctstime + if (fsdu!fTot = (fsdu!fCur+1)) then 0 else ((2*aSifsTime)+ ackctstime + frametime) fi)
tpdu:= setPwrMgt(tpdu, import(dot11Power_ Management_ Mode))
Backoff(0,0)
chk_ rts_cts

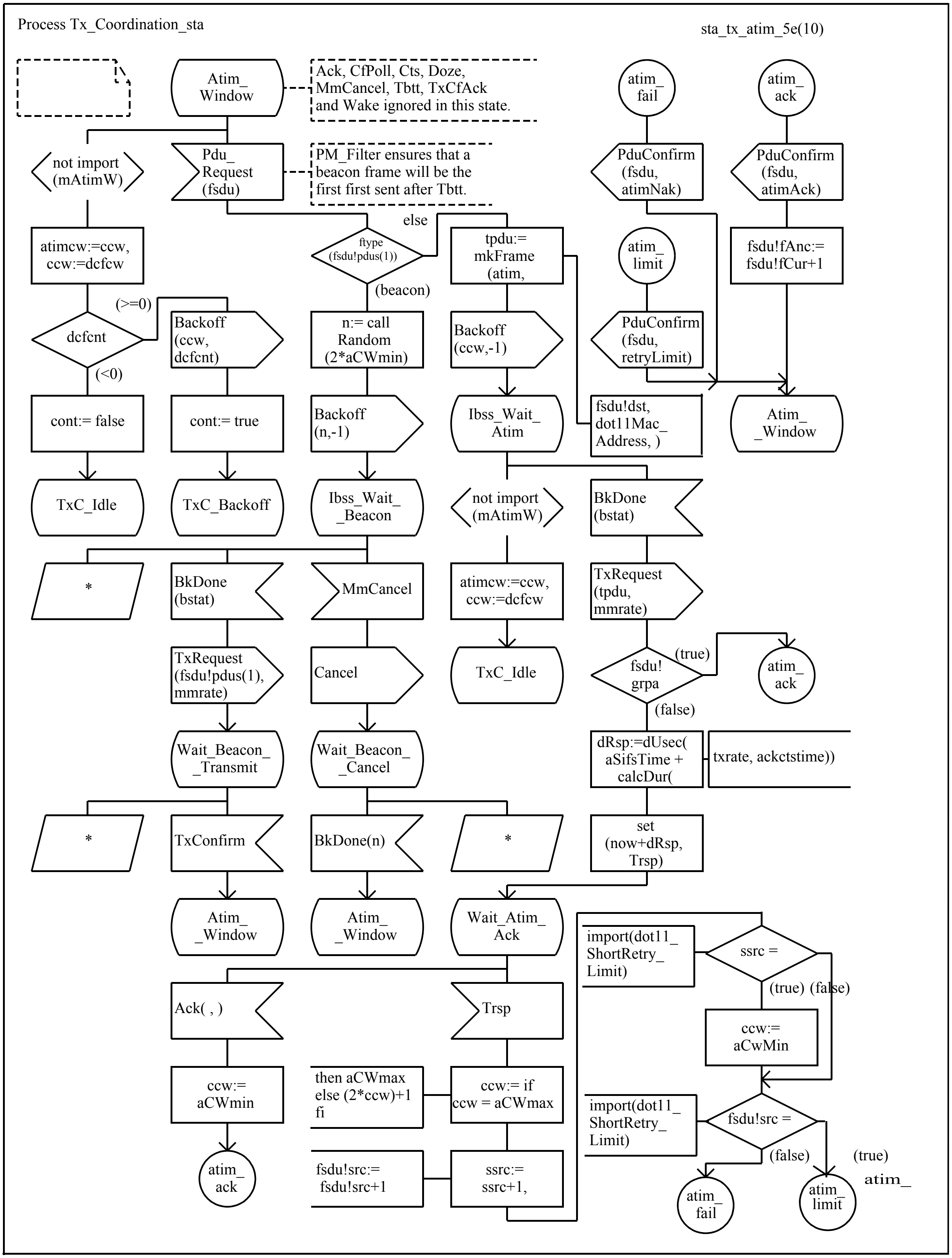

Process Tx_Coordination_sta
sta_tx_dcf_3d(10)
send_
rts
send_
mpdu
Wait_Rts_
_Backoff
Wait_Mpdu_
_Backoff
BkDone(
bstat)
*
TBTT
TBTT
BkDone(
bstat)
*
bstat=-2
(true)
(false)
PduConfirm
(fsdu,
partial)
send_
txReq
bstat=-2
(true)
(false)
Backoff(
ccw,-1)
Cancel
TxRequest
(tpdu,txrate)
Backoff(
ccw,-1)
TxC_Backoff
Atw_Start
(true)
fsdu!grpa
(false)
TxC_Backoff
psmChg:=
if curPsm =
import(
dot11PowerMan_
agementMode)
then false
else true fi
end_
fx
Wait_Pdu_
_Sent
mFxIP:=true,
cTfrg:=
inc(cTfrg)
psmChg
(false)
(true)
TxConfirm
*
export
(mFxIP,
cTfrg)
PsmDone
set(now+dUsec
(aSifsTime +
calcDur(txrate,
stuff(aMpduD_
urationFactor,
sAckCtsLng))+
aPlcpHeaderL_
ength+aPream_
bleLength+aSl_
otTime), Trsp)
TxRequest
(rtsdu,txrate)
mFxIP:=false,
psmChg:=false
Wait_Ack
Wait_Rts_
_Sent
export
(mFxIP)
Ack
(endRx,
txrate)
*
Trsp
TxConfirm
*
ccw:=aCWmin
end_
fx
cNack:=
inc(cNack)
set(now+dUsec
(aSifsTime +
calcDur(txrate,
stuff(aMpduD_
urationFactor,
sAckCtsLng))+
aPlcpHeaderL_
ength+aPream_
bleLength+aSl_
otTime), Trsp)
cnfrm_
pdu
export(cNack)
Wait_Cts
ack_
fail

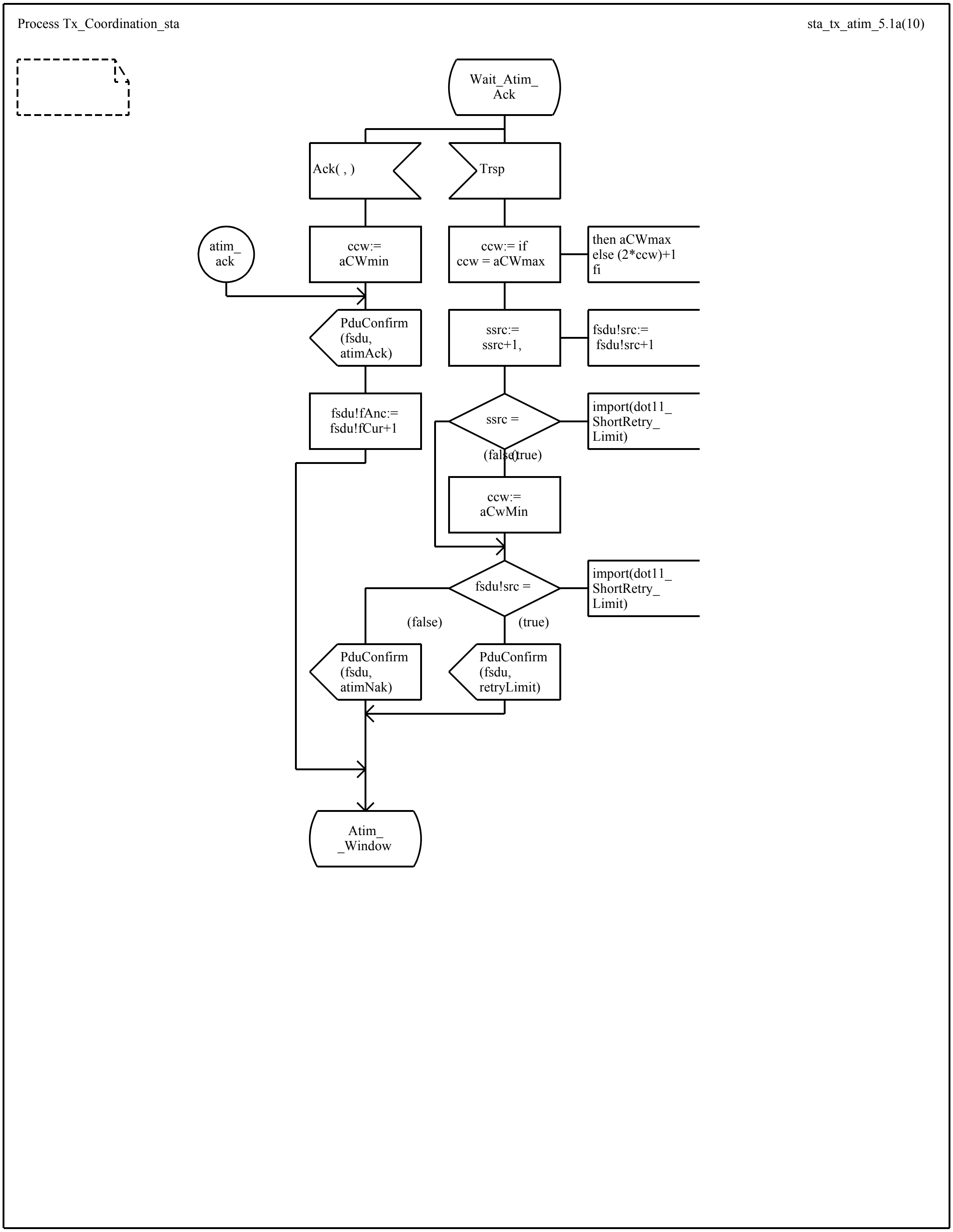

Process Tx_Coordination_sta
sta_tx_dcf_3.1d(10)
Wait_Cts
Cts (endRx, txrate)
Trsp
*
reset (Trsp)
ssrc:=0, fsdu!src:=0
cCts:= inc(cCts)
export(cCts)
set(endRx +dSifsDelay, Tifs)
tpdu:= setDurId(tpdu, calcDur(txrate,
(aSifsTime + (calcDur (txrate,stuff(aMpdu_ DurationFactor,sAck_ CtsLng))+aPlcpHeader_ Length+aPreambleLength) + if (fsdu!fTot = (fsdu! fCur+1)) then 0 else ((2*aSifsTime)+(calcDur (txrate,stuff(aMpdu_ DurationFactor,sAck_ CtsLng)) + aPlcpHeader_ Length + aPreambleLen_ gth)+stuff(aMpduDuration_ Factor,((length(fsdu!pdus (fsdu!fCur+1))+sCrcLng) *8)) + aPlcpHeaderLength + aPreambleLength))))
Wait_Cts_ _Sifs
Tifs
send_ txReq
*
cNcts:= inc(cNcts)
export(cNcts)
cts_ fail
cnfrm_ pdu
slrc:=0, ssrc:=0,
fsdu!lrc:=0, fsdu!src:=0
cTfrg:= inc(cTfrg), cTmcfrm:=
If (fsdu!grpa or ((toDs(tpdu) = 1) and (isGrp(addr3(tpdu))) and (fsdu!fTot=fsdu!fCur+1))) then inc(cTmcfrm) else cTmcfrm fi
fsdu!fTot= fsdu!fCur+1
(true)
(false)
cTfrm:= inc(cTfrm)
PduConfirm (fsdu, success)
Backoff (ccw,-1)
TxC_Backoff
fsdu!eol < now
(true)
(false)
PduConfirm (fsdu, txLife)
TxC_Idle
fsdu!fCur:= fsdu!fCur+1
send_ frag

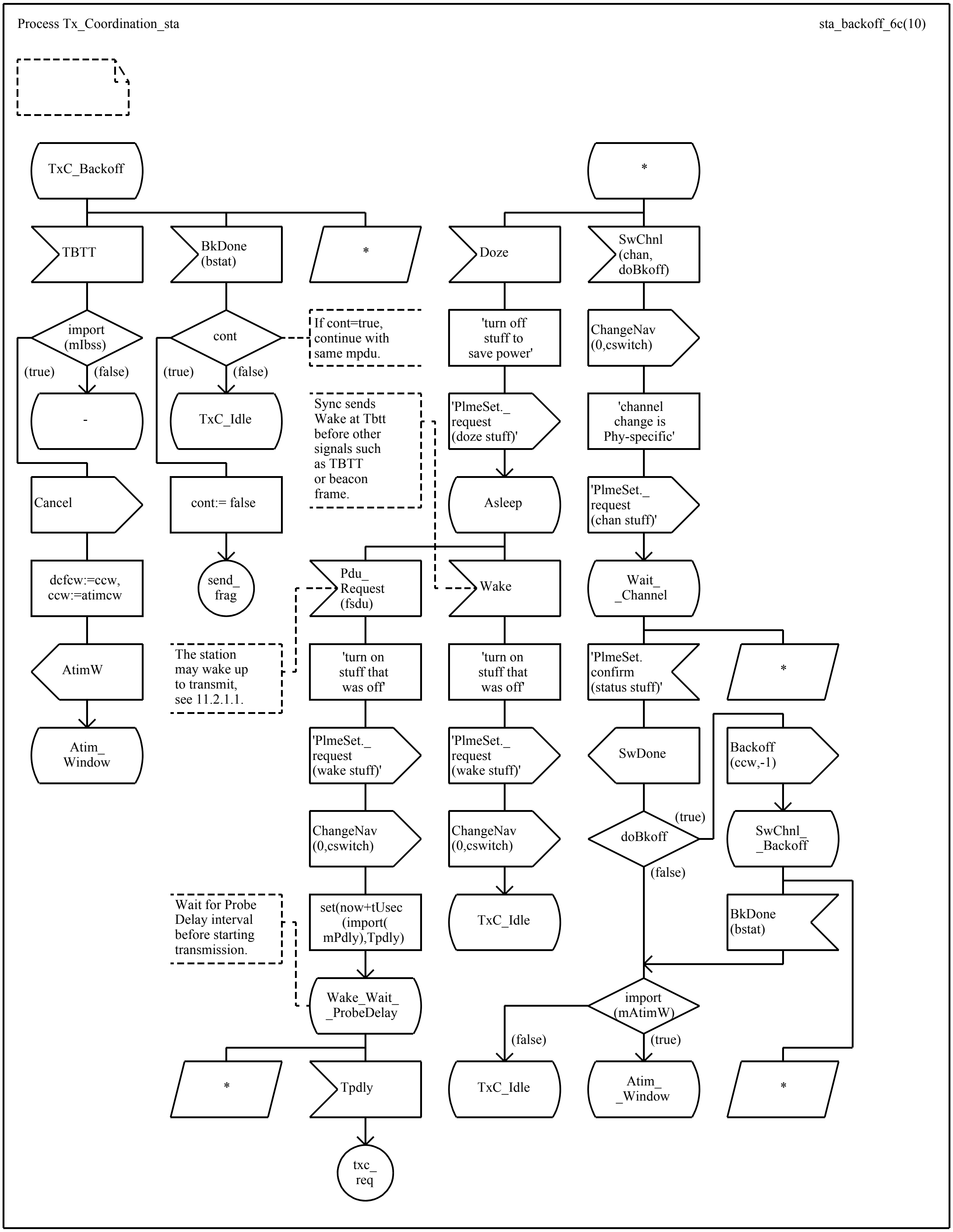

Process Tx_Coordination_sta
sta_tx_dcf_3.2b(11)
chk_
_rts_cts
See 9.2.11
CTS to self is
a protection
mechanism.
(see 9.10)
'Use CTS
to self?
(yes)
(no)
TxTime(length(
tpdu),txrate,
frametime2)
rtsdu:=
mkctl(cts,
frametime2 +
(2*aSifstime) +
ackctstime)
'Use
RTS/CTS
protection?
RTS/CTS
is a protection
mechanism. (see 9.10)
(yes)
(no)
((length
(tpdu) +
sCrcLng) > import(
dot11RtsThreshold)) and
(not fsdu!grpa) and ((fsdu!fCur=0)
or retry(tpdu) or (fsdu!resume))
(true)
(false)
Wait_CTS_
_Backoff
BkDone(
bstat)
*
TBTT
send_
mpdu
bstat=-2
(false)
PduConfirm
(fsdu,
partial)
Backoff(
ccw,-1)
psmChg:=
if curPsm =
import(
dot11PowerMan_
agementMode)
then false
else true fi
Cancel
TxC_Backoff
TxTime(length(
tpdu),txrate,
frametime2)
mFxIP:=true,
cTfrg:=
inc(cTfrg)
Atw_Start
rtsdu:=
mkctl(rts,
frametime2 +
(3*aSifstime) +
(2*ackctstime))
export
mFxIP,
cTfrg)
send_
rts
TxRequest
(rtsdu,txrate)
set(now+dUsec
(aSifsTime),
Tsifs)
Wait_Cts_
_Sent
Wait_Cts_
_Sifs
*
TxConfirm
Tifs
*
send_
txReq

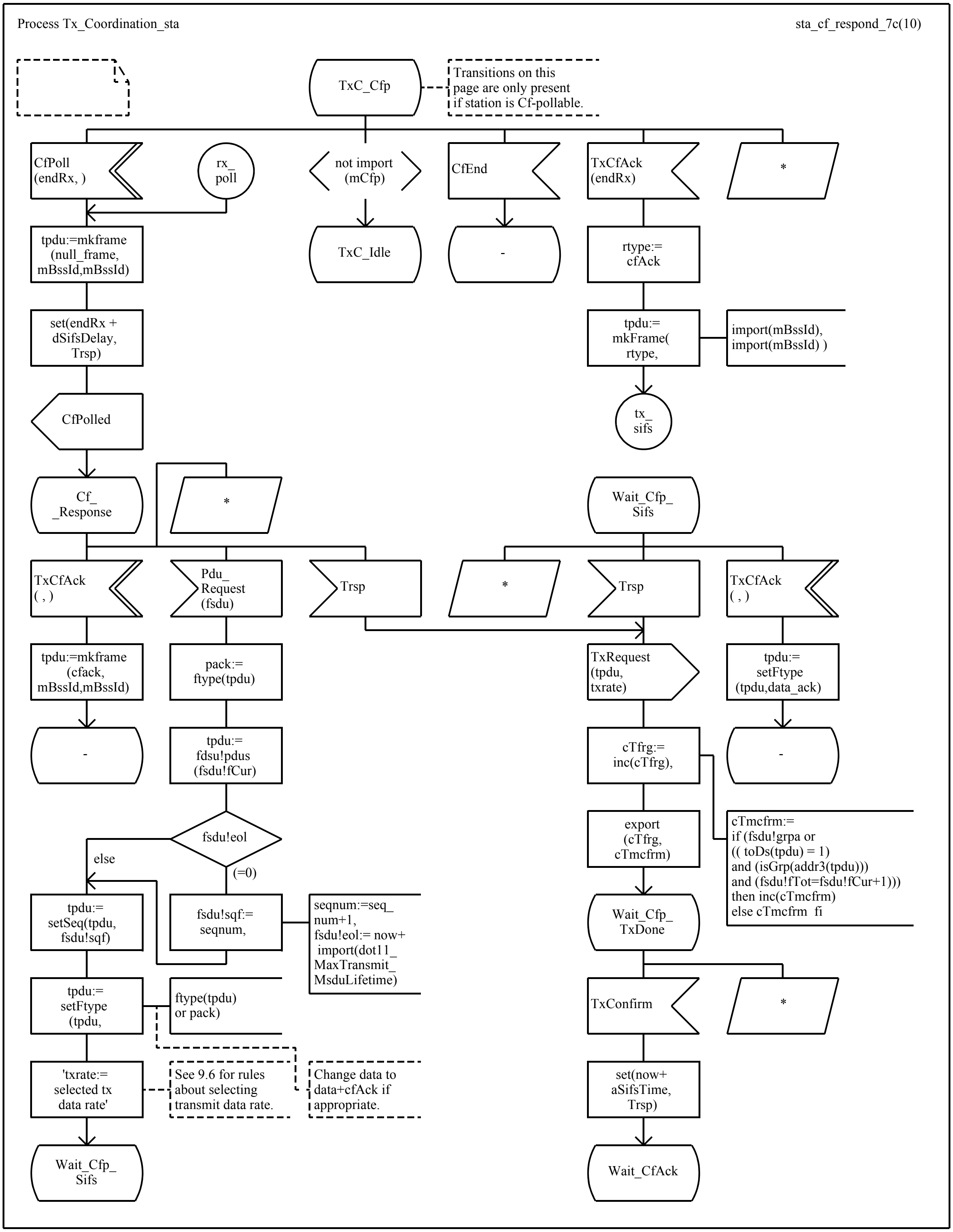


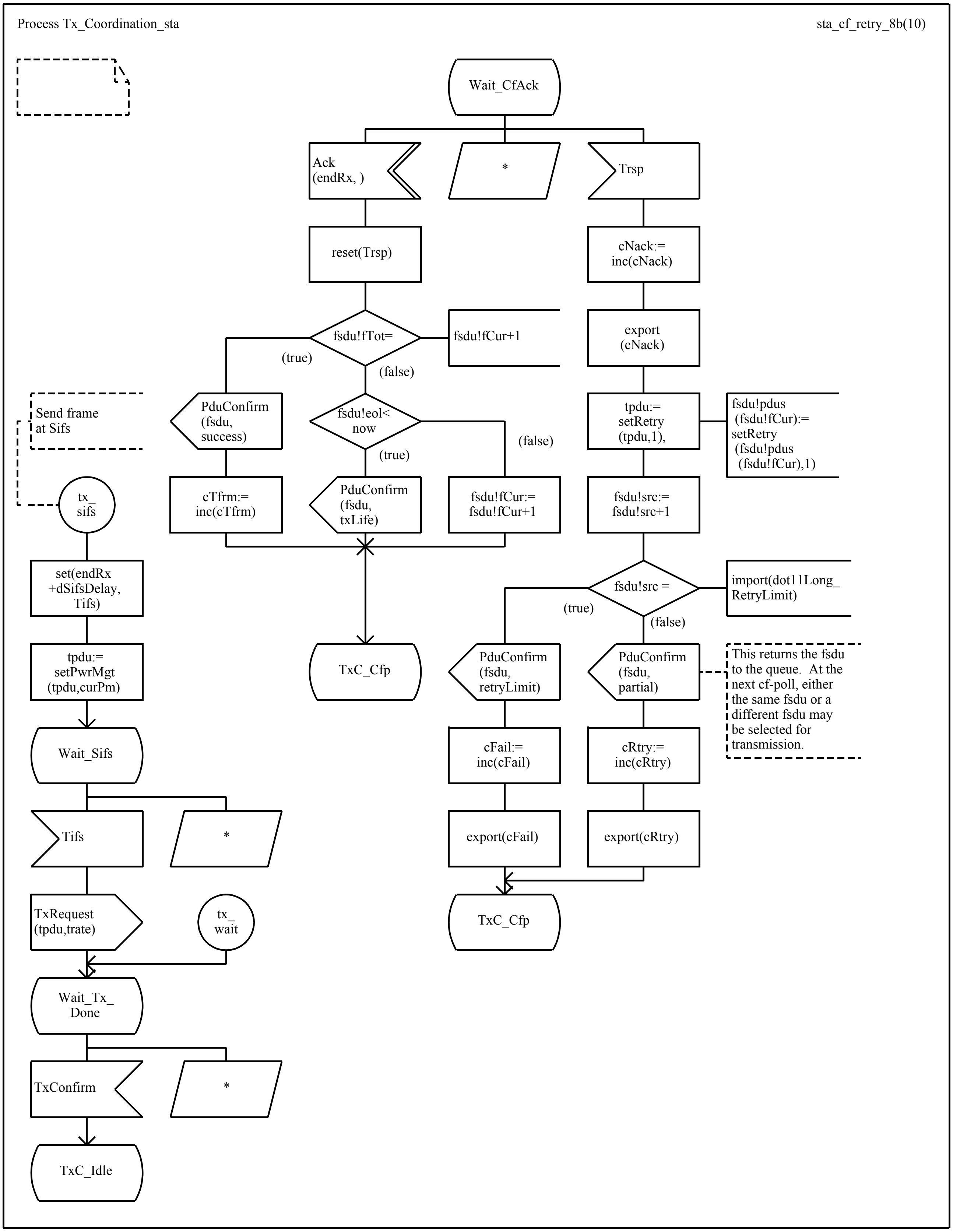
This shows the case where the same pdu is retried after the backoff. It is also allowable to return this fsdu to PM_Filter with status=partial, and to go to TxC_Backoff state with cont=false. This will allow a different pdu (if available) to be sent as the next transmission.

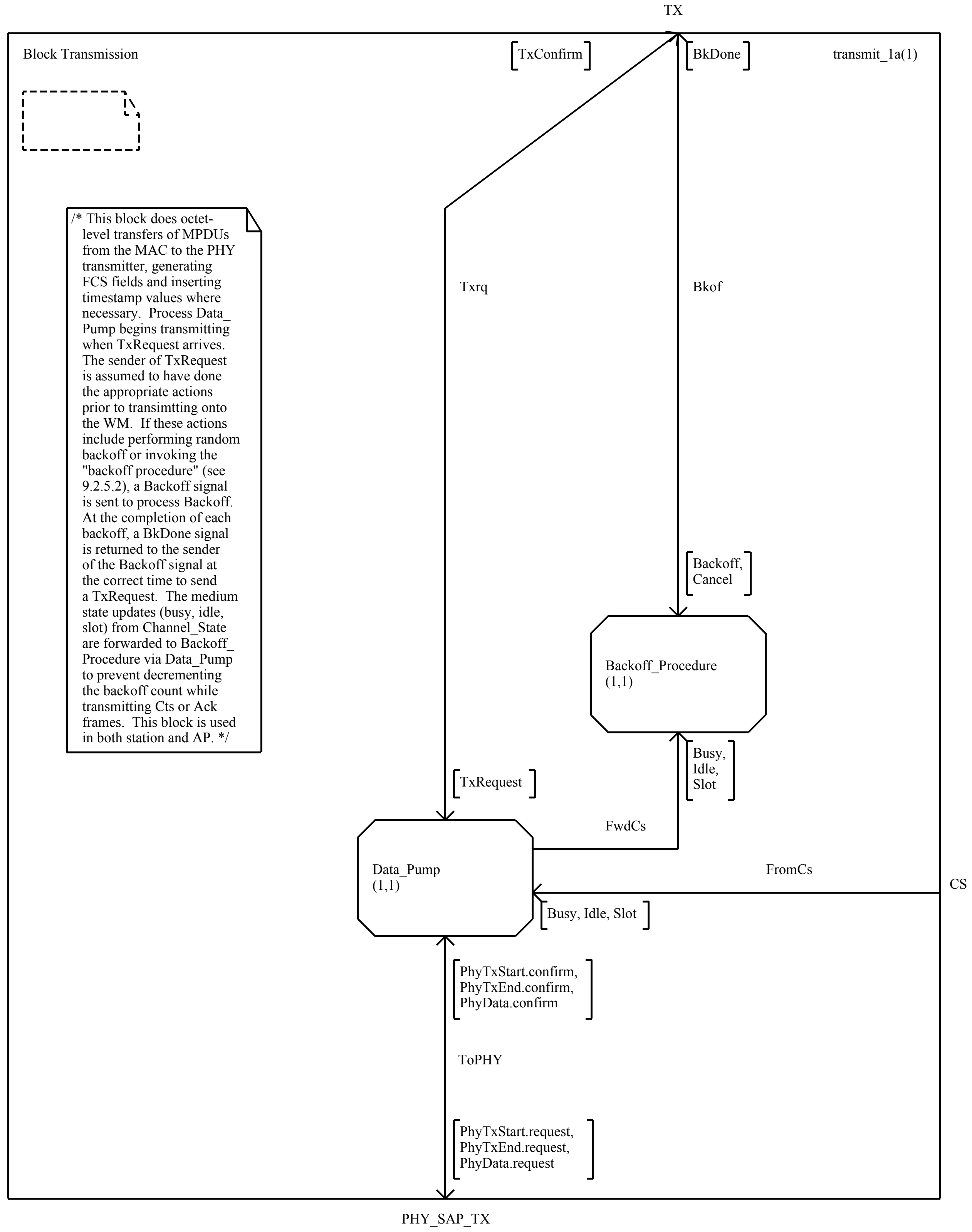

Process Tx_Coordination_sta
sta_tx_atim_5e(10)
Atim_ Window
Ack, CfPoll, Cts, Doze, MmCancel, Tbtt, TxCfAck and Wake ignored in this state.
atim_ fail
atim_ ack
not import (mAtimW)
Pdu_ Request (fsdu)
PM_Filter ensures that a beacon frame will be the first first sent after Tbtt.
PduConfirm (fsdu, atimNak)
PduConfirm (fsdu, atimAck)
else
atimcw:=ccw, ccw:=dcfcw
ftype (fsdu!pdus(1))
tpdu:= mkFrame (atim,
atim_ limit
fsdu!fAnc:= fsdu!fCur+1
(beacon)
(>=0)
dcfcnt
Backoff (ccw, dcfcnt)
n:= call Random (2*aCWmin)
Backoff (ccw,-1)
PduConfirm (fsdu, retryLimit)
(<0)
cont:= false
cont:= true
Backoff (n,-1)
Ibss_Wait_ Atim
fsdu!dst, dot11Mac_ Address,)
Atim_ _Window
TxC_Idle
TxC_Backoff
Ibss_Wait_ _Beacon
not import (mAtimW)
BkDone (bstat)
*
BkDone (bstat)
MmCancel
atimcw:=ccw, ccw:=dcfcw
TxRequest (tpdu, mmrate)
TxRequest (fsdu!pdus(1), mmrate)
Cancel
TxC_Idle
fsdu! grpa
(true)
atim_ ack
(false)
Wait_Beacon_ _Transmit
Wait_Beacon_ _Cancel
dRsp:=dUsec(aSifsTime + calcDur(
txrate, ackctstime))
*
TxConfirm
BkDone(n)
*
set (now+dRsp, Trsp)
Atim_ _Window
Atim_ _Window
Wait_Atim_ Ack
import(dot11_ ShortRetry_ Limit)
ssrc =
(true)
(false)
Ack(,)
Trsp
ccw:= aCwMin
ccw:= aCWmin
then aCWmax else (2*ccw)+1 fi
ccw:= if ccw = aCWmax
import(dot11_ ShortRetry_ Limit)
fsdu!src =
(false)
(true)
atim_ ack
fsdu!src:= fsdu!src+1
ssrc:= ssrc+1,
atim_ fail
atim_ limit
atim_


Process Tx_Coordination_sta
sta_tx_atim_5.1a(10)
Wait_Atim_
Ack
Ack(,)
Trsp
atim_
ack
ccw:=
aCWmin
ccw:= if
ccw = aCWmax
then aCWmax
else (2*ccw)+1
fi
PduConfirm
(fsdu,
atimAck)
ssrc:=
ssrc+1,
fsdu!src:=
fsdu!src+1
fsdu!fAnc:=
fsdu!fCur+1
ssrc =
import(dot11_
ShortRetry_
Limit)
(false)
(true)
ccw:=
aCwMin
fsdu!src =
import(dot11_
ShortRetry_
Limit)
(false)
(true)
PduConfirm
(fsdu,
atimNak)
PduConfirm
(fsdu,
retryLimit)
Atim_
_Window

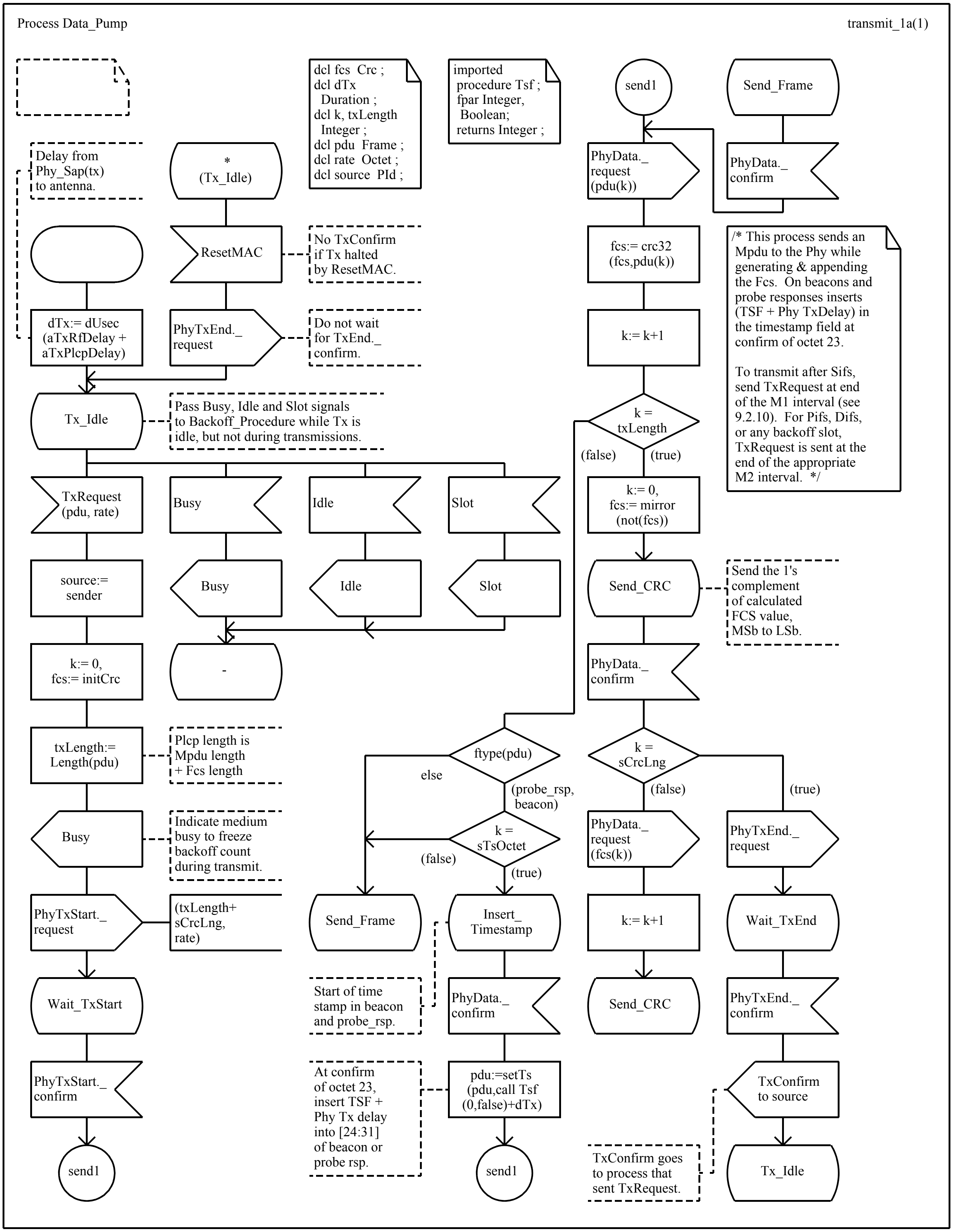

Process Tx_Coordination_sta
sta_backoff_6c(10)
TxC_Backoff
*
TBTT
BkDone (bstat)
*
Doze
SwChnl (chan, doBkoff)
import (mIbss)
(true)
(false)
cont
(true)
(false)
If cont=true, continue with same mpdu.
'turn off stuff to save power'
ChangeNav (0,cswitch)
-
TxC_Idle
Sync sends Wake at Tbtt before other signals such as TBTT or beacon frame.
'PlmeSet._ request (doze stuff)'
'channel change is Phy-specific'
Cancel
cont:= false
Asleep
'PlmeSet._ request (chan stuff)'
dcfcw:=ccw, ccw:=atimcw
send_ frag
Pdu_ Request (fsdu)
Wake
Wait_ _Channel
AtimW
The station may wake up to transmit, see 11.2.1.1.
'turn on stuff that was off'
'turn on stuff that was off'
'PlmeSet. confirm (status stuff)'
*
Atim_ Window
'PlmeSet._ request (wake stuff)'
'PlmeSet._ request (wake stuff)'
SwDone
Backoff (ccw,-1)
ChangeNav (0,cswitch)
ChangeNav (0,cswitch)
doBkoff
(true)
(false)
SwChnl_ _Backoff
Wait for Probe Delay interval before starting transmission.
set(now+tUsec (import(mPdly),Tpdly)
TxC_Idle
BkDone (bstat)
Wake_Wait_ _ProbeDelay
import (mAtimW)
(false)
(true)
*
Tpdly
TxC_Idle
Atim_ _Window
*
txc_ req

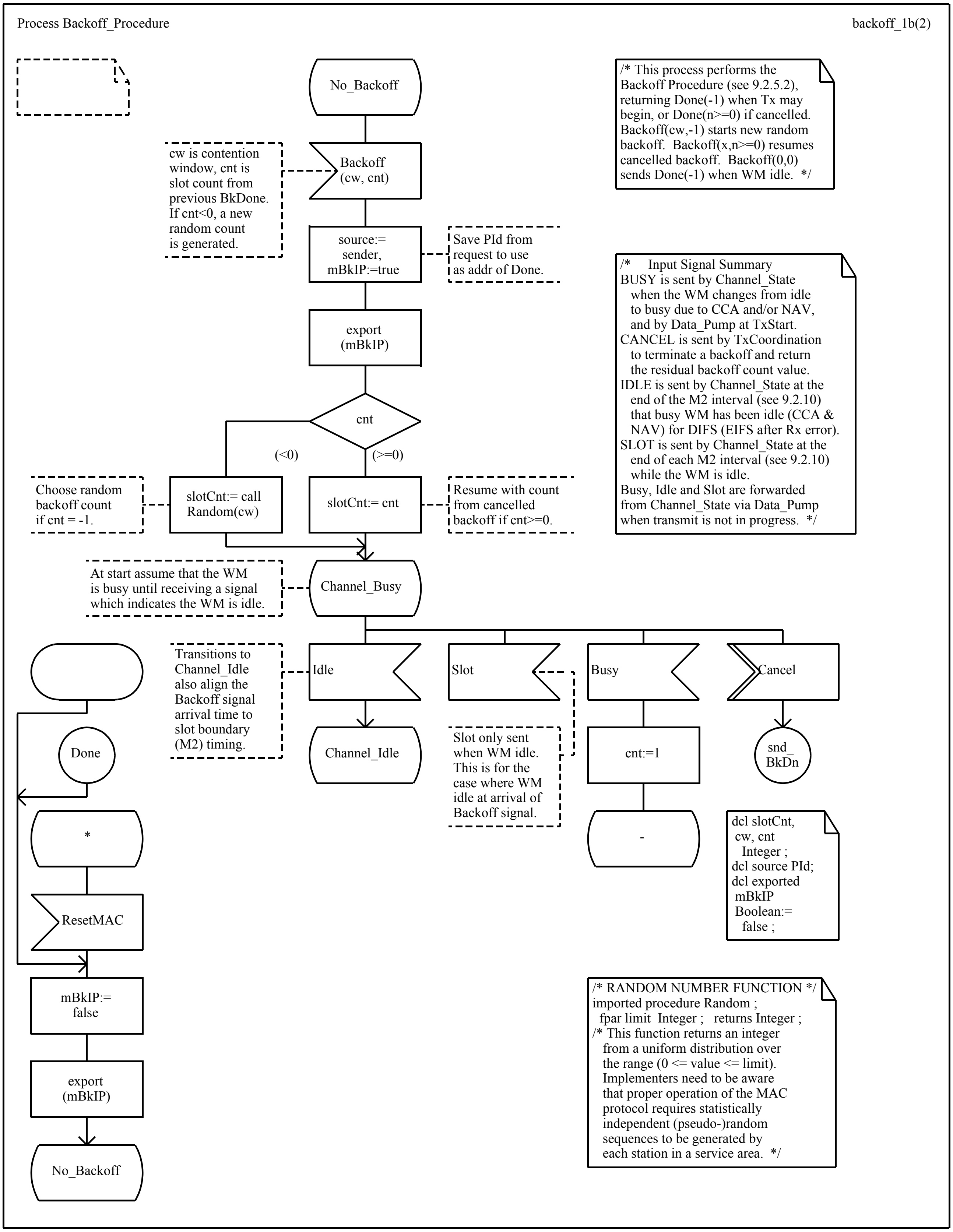

Process Tx_Coordination_sta
sta_cf_respond_7c(10)
TxC_Cfp
Transitions on this page are only present if station is Cf-pollable.
CfPoll (endRx,)
rx_ poll
not import (mCfp)
CfEnd
TxCfAck (endRx)
*
tpdu:=mkframe (null_frame, mBssId,mBssId)
TxC_Idle
-
rtype:= cfAck
set(endRx + dSifsDelay, Trsp)
tpdu:= mkFrame(rtype,
import(mBssId), import(mBssId))
CfPolled
tx_ sifs
Cf_ _Response
*
Wait_Cfp_ Sifs
TxCfAck (,)
Pdu_ Request (fsdu)
Trsp
*
Trsp
TxCfAck (,)
tpdu:=mkframe (cfack, mBssId,mBssId)
pack:= ftype(tpdu)
TxRequest (tpdu, txrate)
tpdu:= setFtype (tpdu,data_ack)
-
tpdu:= fdsu!pdus (fsdu!fCur)
cTfrg:= inc(cTfrg),
-
fsdu!eol
else
(=0)
export (cTfrg, cTmcfrm)
cTmcfrm:= if (fsdu!grpa or ((toDs(tpdu) = 1) and (isGrp(addr3(tpdu))) and (fsdu!fTot=fsdu!fCur+1))) then inc(cTmcfrm) else cTmcfrm fi
tpdu:= setSeq(tpdu, fsdu!sqf)
fsdu!sqf:= seqnum,
seqnum:=seq_ num+1, fsdu!eol:= now+ import(dot11_ MaxTransmit_ MsduLifetime)
Wait_Cfp_ TxDone
tpdu:= setFtype (tpdu,
ftype(tpdu) or pack)
TxConfirm
*
'txrate:= selected tx data rate'
See 9.6 for rules about selecting transmit data rate.
Change data to data+cfAck if appropriate.
set(now+ aSifsTime, Trsp)
Wait_Cfp_ Sifs
Wait_CfAck

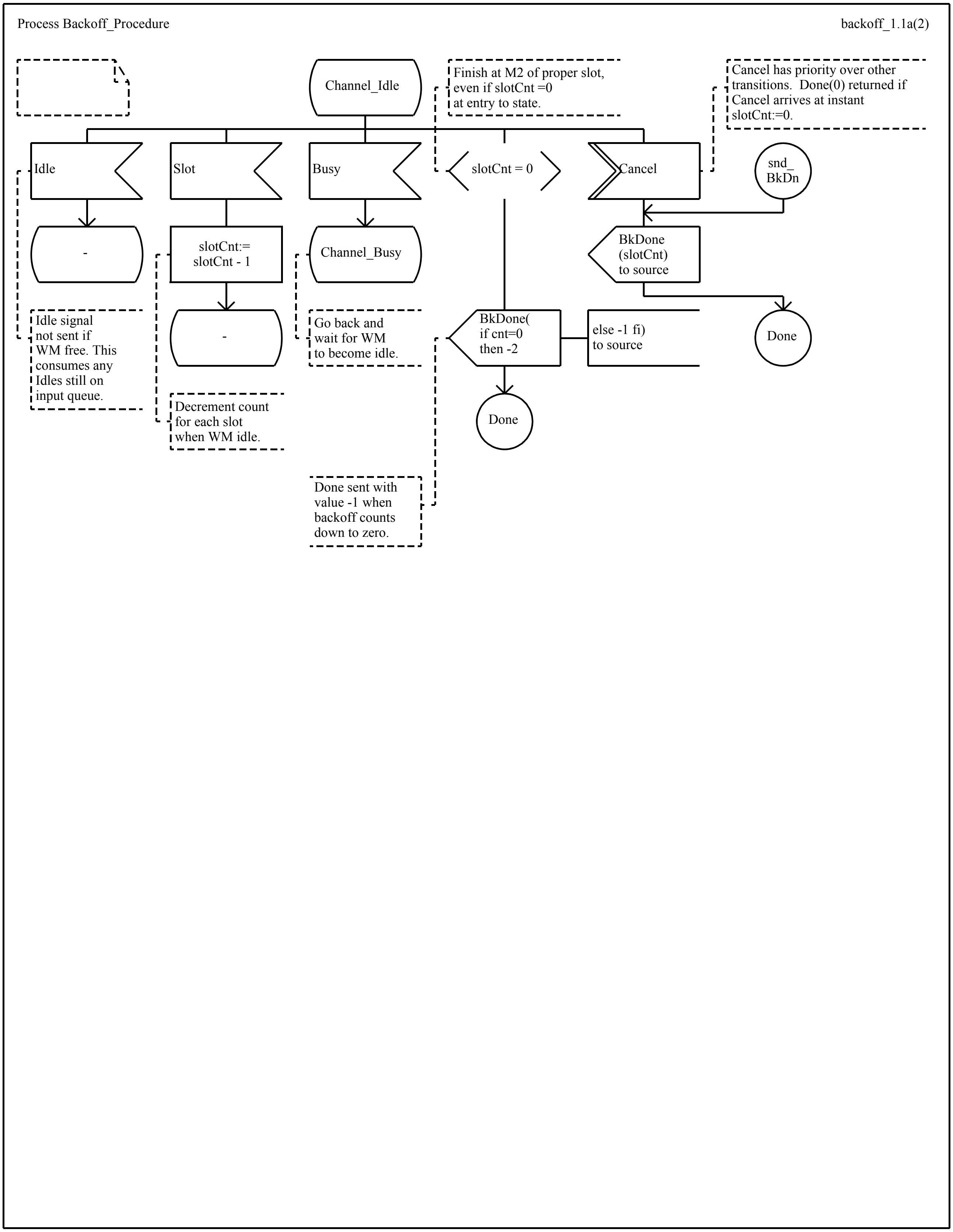

Process Tx_Coordination_sta
sta_cf_retry_8b(10)
Wait_CfAck
Ack
(endRx,)
*
Trsp
reset(Trsp)
cNack:=
inc(cNack)
fsdu!fTot=
fsdu!fCur+1
(true)
(false)
export
(cNack)
Send frame
at Sifs
PduConfirm
(fsdu,
success)
fsdu!eol<
now
(true)
(false)
tpdu:=
setRetry
(tpdu,1),
fsdu!pdus
(fsdu!fCur):=
setRetry
(fsdu!pdus
(fsdu!fCur),1)
tx_
sifs
cTfrm:=
inc(cTfrm)
PduConfirm
(fsdu,
txLife)
fsdu!fCur:=
fsdu!fCur+1
fsdu!src:=
fsdu!src+1
set(endRx
+dSifsDelay,
Tifs)
fsdu!src =
import(dot11Long_
RetryLimit)
(true)
(false)
tpdu:=
setPwrMgt
(tpdu,curPm)
TxC_Cfp
PduConfirm
(fsdu,
retryLimit)
PduConfirm
(fsdu,
partial)
This returns the fsdu
to the queue. At the
next cf-poll, either
the same fsdu or a
different fsdu may
be selected for
transmission.
Wait_Sifs
cFail:=
inc(cFail)
cRtry:=
inc(cRtry)
Tifs
*
export(cFail)
export(cRtry)
TxRequest
(tpdu,trate)
tx_
wait
TxC_Cfp
Wait_Tx_
Done
TxConfirm
*
TxC_Idle

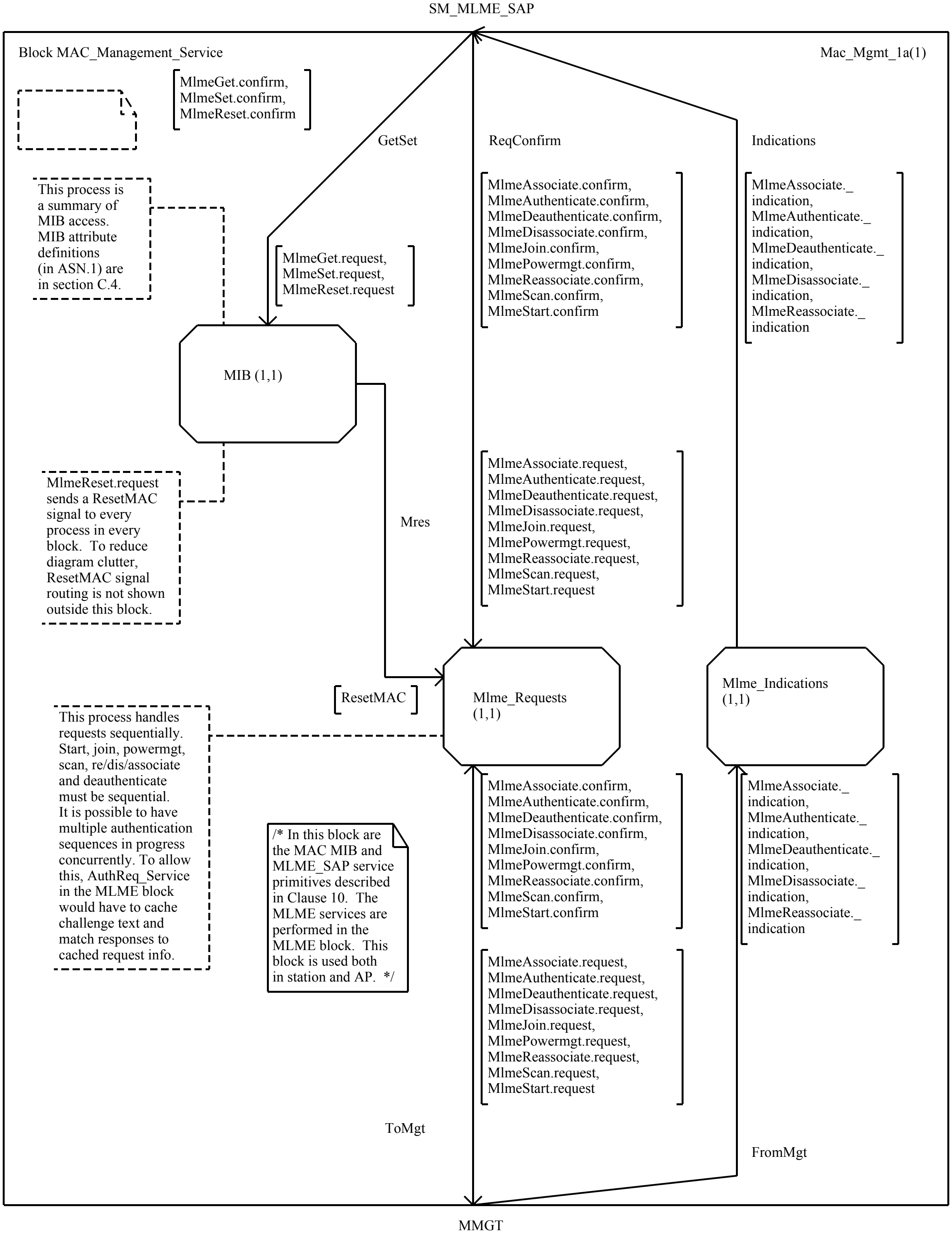


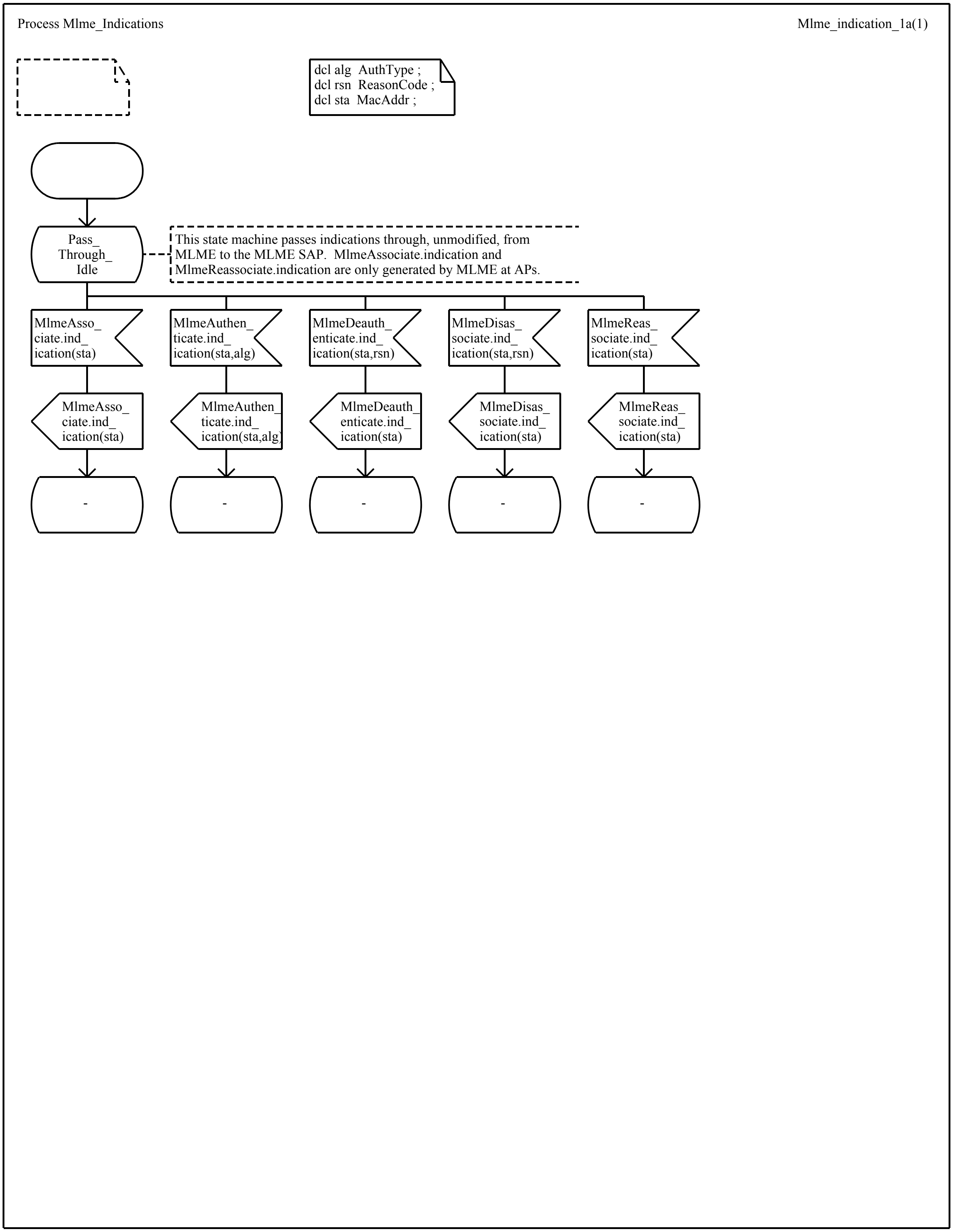
/* This block does octet-level transfers of MPDUs from the MAC to the PHY transmitter, generating FCS fields and inserting timestamp values where necessary. Process Data_ Pump begins transmitting when TxRequest arrives. The sender of TxRequest is assumed to have done the appropriate actions prior to transimtting onto the WM. If these actions include performing random backoff or invoking the "backoff procedure" (see 9.2.5.2), a Backoff signal is sent to process Backoff. At the completion of each backoff, a BkDone signal is returned to the sender of the Backoff signal at the correct time to send a TxRequest. The medium state updates (busy, idle, slot) from Channel_State are forwarded to Backoff_ Procedure via Data_Pump to prevent decrementing the backoff count while transmitting Cts or Ack frames. This block is used in both station and AP. */

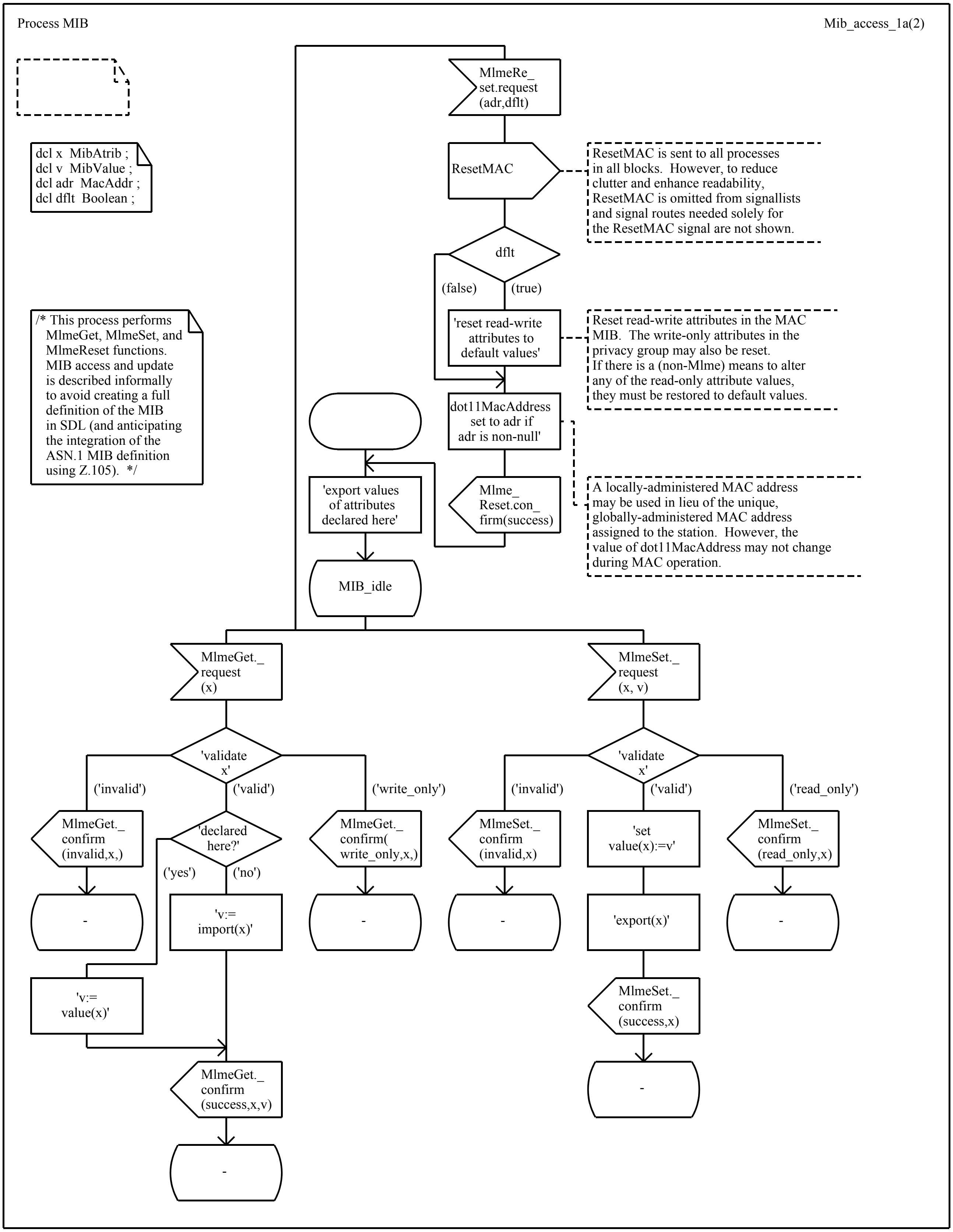

Process Data_Pump
transmit_1a(1)
dcl fcs Crc ;
dcl dTx Duration ;
dcl k, txLength Integer ;
dcl pdu Frame ;
dcl rate Octet ;
dcl source PId ;
imported procedure Tsf ;
fpar Integer, Boolean;
returns Integer ;
Delay from Phy_Sap(tx) to antenna.
dTx:= dUsec (aTxRfDelay + aTxPlcpDelay)
*
(Tx_Idle)
ResetMAC
No TxConfirm if Tx halted by ResetMAC.
PhyTxEnd._request
Do not wait for TxEnd._confirm.
Tx_Idle
Pass Busy, Idle and Slot signals to Backoff_Procedure while Tx is idle, but not during transmissions.
TxRequest (pdu, rate)
Busy
Idle
Slot
source:= sender
Busy
Idle
Slot
-
k:= 0, fcs:= initCrc
txLength:= Length(pdu)
Plcp length is Mpdu length + Fcs length
Busy
Indicate medium busy to freeze backoff count during transmit.
PhyTxStart._request
(txLength+ sCrcLng, rate)
Wait_TxStart
PhyTxStart._confirm
send1
send1
Send_Frame
PhyData._request (pdu(k))
PhyData._confirm
fcs:= crc32 (fcs,pdu(k))
k:= k+1
/* This process sends an Mpdu to the Phy while generating & appending the Fcs. On beacons and probe responses inserts (TSF + Phy TxDelay) in the timestamp field at confirm of octet 23.
To transmit after Sifs, send TxRequest at end of the M1 interval (see 9.2.10). For Pifs, Difs, or any backoff slot, TxRequest is sent at the end of the appropriate M2 interval. */
k = txLength
(false)
(true)
k:= 0, fcs:= mirror (not(fcs))
Send_CRC
Send the 1's complement of calculated FCS value, MSb to LSb.
PhyData._confirm
ftype(pdu)
else
(probe_rsp, beacon)
k = sTsOctet
(false)
(true)
Send_Frame
Insert_ Timestamp
Start of time stamp in beacon and probe_rsp.
PhyData._confirm
pdu:=setTs (pdu,call Tsf (0,false)+dTx)
At confirm of octet 23, insert TSF + Phy Tx delay into [24:31] of beacon or probe rsp.
send1
k = sCrcLng
(false)
(true)
PhyData._request (fcs(k))
k:= k+1
Send_CRC
PhyTxEnd._request
Wait_TxEnd
PhyTxEnd._confirm
TxConfirm to source
TxConfirm goes to process that sent TxRequest.
Tx_Idle


Process Backoff_Procedure
backoff_1b(2)
No_Backoff
Backoff
(cw, cnt)
cw is contention window, cnt is slot count from previous BkDone. If cnt<0, a new random count is generated.
/* This process performs the Backoff Procedure (see 9.2.5.2), returning Done(-1) when Tx may begin, or Done(n>=0) if cancelled. Backoff(cw,-1) starts new random backoff. Backoff(x,n>=0) resumes cancelled backoff. Backoff(0,0) sends Done(-1) when WM idle. */
source:= sender, mBkIP:=true
Save PId from request to use as addr of Done.
export (mBkIP)
/* Input Signal Summary
BUSY is sent by Channel_State when the WM changes from idle to busy due to CCA and/or NAV, and by Data_Pump at TxStart.
CANCEL is sent by TxCoordination to terminate a backoff and return the residual backoff count value.
IDLE is sent by Channel_State at the end of the M2 interval (see 9.2.10) that busy WM has been idle (CCA & NAV) for DIFS (EIFS after Rx error).
SLOT is sent by Channel_State at the end of each M2 interval (see 9.2.10) while the WM is idle.
Busy, Idle and Slot are forwarded from Channel_State via Data_Pump when transmit is not in progress. */
cnt
(<0)
(>=0)
Choose random backoff count if cnt = -1.
slotCnt:= call Random(cw)
slotCnt:= cnt
Resume with count from cancelled backoff if cnt>=0.
At start assume that the WM is busy until receiving a signal which indicates the WM is idle.
Channel_Busy
Transitions to Channel_Idle also align the Backoff signal arrival time to slot boundary (M2) timing.
Idle
Slot
Busy
Cancel
Channel_Idle
Slot only sent when WM idle. This is for the case where WM idle at arrival of Backoff signal.
cnt:=1
-
snd_ BkDn
Done
*
ResetMAC
mBkIP:= false
export (mBkIP)
No_Backoff
dcl slotCnt, cw, cnt Integer ; dcl source PId; dcl exported mBkIP Boolean:= false ;
/* RANDOM NUMBER FUNCTION */
imported procedure Random ;
fpar limit Integer ; returns Integer ;
/* This function returns an integer from a uniform distribution over the range (0 <= value <= limit). Implementers need to be aware that proper operation of the MAC protocol requires statistically independent (pseudo-)random sequences to be generated by each station in a service area. */


Process Backoff_Procedure
backoff_1.1a(2)
Channel_Idle
Finish at M2 of proper slot,
even if slotCnt =0
at entry to state.
Cancel has priority over other
transitions. Done(0) returned if
Cancel arrives at instant
slotCnt:=0.
Idle
Slot
Busy
slotCnt = 0
Cancel
snd_
BkDn
-
slotCnt:=
slotCnt - 1
Channel_Busy
BkDone
(slotCnt)
to source
Idle signal
not sent if
WM free. This
consumes any
Idles still on
input queue.
-
Go back and
wait for WM
to become idle.
BkDone(
if cnt=0
then -2
else -1 fi)
to source
Done
Decrement count
for each slot
when WM idle.
Done
Done sent with
value -1 when
backoff counts
down to zero.


SM_MLME_SAP
Block MAC_Management_Service
Mac_Mgmt_1a(1)
MlmeGet.confirm,
MlmeSet.confirm,
MlmeReset.confirm
GetSet
ReqConfirm
Indications
This process is
a summary of
MIB access.
MIB attribute
definitions
(in ASN.1) are
in section C.4.
MlmeAssociate.confirm,
MlmeAuthenticate.confirm,
MlmeDeauthenticate.confirm,
MlmeDisassociate.confirm,
MlmeJoin.confirm,
MlmePowermgt.confirm,
MlmeReassociate.confirm,
MlmeScan.confirm,
MlmeStart.confirm
MlmeAssociate._
indication,
MlmeAuthenticate._
indication,
MlmeDeauthenticate._
indication,
MlmeDisassociate._
indication,
MlmeReassociate._
indication
MlmeGet.request,
MlmeSet.request,
MlmeReset.request
MIB (1,1)
MlmeReset.request
sends a ResetMAC
signal to every
process in every
block. To reduce
diagram clutter,
ResetMAC signal
routing is not shown
outside this block.
Mres
MlmeAssociate.request,
MlmeAuthenticate.request,
MlmeDeauthenticate.request,
MlmeDisassociate.request,
MlmeJoin.request,
MlmePowermgt.request,
MlmeReassociate.request,
MlmeScan.request,
MlmeStart.request
ResetMAC
Mlme_Requests
(1,1)
Mlme_Indications
(1,1)
This process handles
requests sequentially.
Start, join, powermgt,
scan, re/dis/associate
and deauthenticate
must be sequential.
It is possible to have
multiple authentication
sequences in progress
concurrently. To allow
this, AuthReq_Service
in the MLME block
would have to cache
challenge text and
match responses to
cached request info.
/* In this block are
the MAC MIB and
MLME_SAP service
primitives described
in Clause 10. The
MLME services are
performed in the
MLME block. This
block is used both
in station and AP. */
MlmeAssociate.confirm,
MlmeAuthenticate.confirm,
MlmeDeauthenticate.confirm,
MlmeDisassociate.confirm,
MlmeJoin.confirm,
MlmePowermgt.confirm,
MlmeReassociate.confirm,
MlmeScan.confirm,
MlmeStart.confirm
MlmeAssociate._
indication,
MlmeAuthenticate._
indication,
MlmeDeauthenticate._
indication,
MlmeDisassociate._
indication,
MlmeReassociate._
indication
MlmeAssociate.request,
MlmeAuthenticate.request,
MlmeDeauthenticate.request,
MlmeDisassociate.request,
MlmeJoin.request,
MlmePowermgt.request,
MlmeReassociate.request,
MlmeScan.request,
MlmeStart.request
ToMgt
FromMgt
MMGT


Process Mlme_Indications
Mlme_indication_1a(1)
dcl alg AuthType ;
dcl rsn ReasonCode ;
dcl sta MacAddr ;
Pass_ Through_ Idle
This state machine passes indications through, unmodified, from MLME to the MLME SAP. MlmeAssociate.indication and MlmeReassociate.indication are only generated by MLME at APs.
MlmeAsso_ ciate.ind_ ication(sta)
MlmeAuthen_ ticate.ind_ ication(sta,alg)
MlmeDeauth_ enticate.ind_ ication(sta,rsn)
MlmeDisas_ sociate.ind_ ication(sta,rsn)
MlmeReas_ sociate.ind_ ication(sta)
MlmeAsso_ ciate.ind_ ication(sta)
MlmeAuthen_ ticate.ind_ ication(sta,alg
MlmeDeauth_ enticate.ind_ ication(sta)
MlmeDisas_ sociate.ind_ ication(sta)
MlmeReas_ sociate.ind_ ication(sta)
-
-
-
-
-


Process MIB
Mib_access_1a(2)
dcl x MibAtrib ;
dcl v MibValue ;
dcl adr MacAddr ;
dcl dflt Boolean ;
/* This process performs MlmeGet, MlmeSet, and MlmeReset functions. MIB access and update is described informally to avoid creating a full definition of the MIB in SDL (and anticipating the integration of the ASN.1 MIB definition using Z.105). */
MlmeRe_ set.request (adr,dflt)
ResetMAC
ResetMAC is sent to all processes in all blocks. However, to reduce clutter and enhance readability, ResetMAC is omitted from signallists and signal routes needed solely for the ResetMAC signal are not shown.
dflt
(false)
(true)
'reset read-write attributes to default values'
Reset read-write attributes in the MAC MIB. The write-only attributes in the privacy group may also be reset. If there is a (non-Mlme) means to alter any of the read-only attribute values, they must be restored to default values.
dot11MacAddress set to adr if adr is non-null'
A locally-administered MAC address may be used in lieu of the unique, globally-administered MAC address assigned to the station. However, the value of dot11MacAddress may not change during MAC operation.
Mlme_ Reset.con_ firm(success)
'export values of attributes declared here'
MIB_idle
MlmeGet._ request (x)
'validate x'
('invalid')
('valid')
('write_only')
MlmeGet._ confirm (invalid,x,)
'declared here?'
('yes')
('no')
MlmeGet._ confirm(write_only,x,)
-
'v:= import(x)'
-
'v:= value(x)'
MlmeGet._ confirm (success,x,v)
-
MlmeSet._ request (x, v)
'validate x'
('invalid')
('valid')
('read_only')
MlmeSet._ confirm (invalid,x)
'set value(x):=v'
MlmeSet._ confirm (read_only,x)
-
'export(x)'
-
MlmeSet._ confirm (success,x)
-

Process MIB

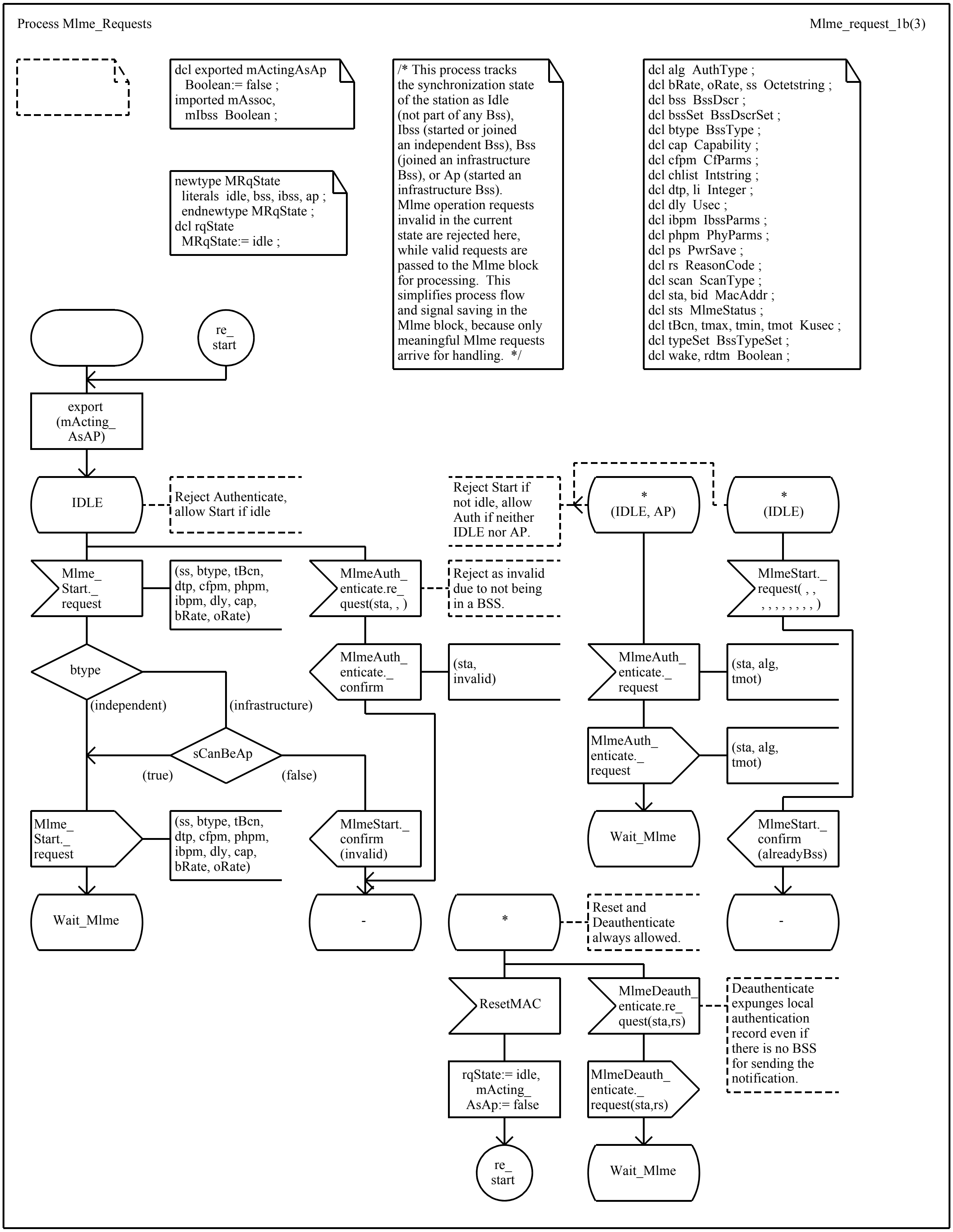
Mib_import_export_2b(2)

```
/* Import of {Read-Only} MIB counter
   values exported from other processes */
imported
  dot11AckFailureCount,
  dot11FailedCount,
  dot11FcsErrorCount,
  dot11FrameDuplicateCount,
  dot11MulticastReceivedFrameCount,
  dot11MulticastTransmittedFrameCount,
  dot11MultipleRetryCount,
  dot11ReceivedFragmentCount,
  dot11RetryCount,
  dot11RtsFailureCount,
  dot11RtsSuccessCount,
  dot11TransmittedFragmentCount,
  dot11WepExcludedCount,
  dot11WepIcvErrorCount,
  dot11WepUndecryptableCount  Counter32 ;
```

```
/* Declarations of MIB attributes exported from
    this process */

        /* Read-Write attributes */
dcl exported
   dot11AuthenticationAlgorithms  AuthTypeSet:=
     incl(open_system, shared_key),
  dot11ExcludeUnencrypted  Boolean:= false,
  dot11FragmentationThreshold  Integer:= 2346,
  dot11GroupAddresses  MacAddrSet:= empty,
  dot11LongRetryLimit  Integer:= 4,
  dot11MaxReceiveLifetime  Kusec:= 512,
  dot11MaxTransmitMsduLifetime  Kusec:= 512,
  dot11MediumOccupancyLimit  Kusec:= 100,
  dot11PrivacyInvoked  Boolean:= false,
  mReceiveDTIMs  Boolean:= true,
  dot11CfpPeriod  Integer:= 1,
  dot11CfpMaxDuration  Kusec:= 200,
  dot11AuthenticationResponseTimeout  Kusec:= 512,
  dot11RtsThreshold  Integer:= 3000,
  dot11ShortRetryLimit  Integer:= 7,
  dot11WepDefaultKeyId  KeyIndex:= 0,
  dot11CurrentChannelNumber  Integer:= 0,
  dot11CurrentSet  Integer:= 0,
  dot11CurrentPattern  Integer:= 0,
  dot11CurrentIndex  Integer:= 0 ;

        /* Write-Only attributes */
dcl exported
  dot11WepDefaultKeys  KeyVector:= nullKey,
  dot11WepKeyMappings
     KeyMapArray:= (. nullAddr, false, nullKey .) ;
```

```
/* The following Read-Only attributes in the
   MAC MIB are defined as synonyms (named
   constants) rather than remote variables
   because they describe properties of the
   station which are static, at least during
   any single instance of MAC operation:
      dot11AuthenticationAlgorithms  AuthTypeSet,
      dot11CfPollable  Boolean,
      dot11MacAddress  MacAddr,
      dot11ManufacturerID  Octetstring,
      dot11PrivacyOptionImplemented  Boolean,
      dot11ProductID  Octetstring,
      aStationID  MacAddr,
      dot11WepKeyMappingLength  Integer ;

   In addition, all Read-Only attributes in the
   PHY MIB which are accessed by the MAC
   are defined as synonyms.
*/
```

```
/* NOTE:
   The values listed for MAC MIB attributes are the
   specified default values for those attributes.
   The values listed for PHY MIB attributes are either
   the default values for the FH PHY, or arbitrary
   values within the specified range.  The specific
   values for PHY attributes in this SDL description
   of the MAC do not have normative significance.
*/
```



Process Mlme_Requests

Mlme_request_1b(3)

```
dcl exported mActingAsAp
  Boolean:= false ;
imported mAssoc,
  mIbss  Boolean ;
```

```
newtype MRqState
  literals  idle, bss, ibss, ap ;
  endnewtype MRqState ;
dcl rqState
  MRqState:= idle ;
```

/* This process tracks the synchronization state of the station as Idle (not part of any Bss), Ibss (started or joined an independent Bss), Bss (joined an infrastructure Bss), or Ap (started an infrastructure Bss). Mlme operation requests invalid in the current state are rejected here, while valid requests are passed to the Mlme block for processing. This simplifies process flow and signal saving in the Mlme block, because only meaningful Mlme requests arrive for handling. */

```
dcl alg  AuthType ;
dcl bRate, oRate, ss  Octetstring ;
dcl bss  BssDscr ;
dcl bssSet  BssDscrSet ;
dcl btype  BssType ;
dcl cap  Capability ;
dcl cfpm  CfParms ;
dcl chlist  Intstring ;
dcl dtp, li  Integer ;
dcl dly  Usec ;
dcl ibpm  IbssParms ;
dcl phpm  PhyParms ;
dcl ps  PwrSave ;
dcl rs  ReasonCode ;
dcl scan  ScanType ;
dcl sta, bid  MacAddr ;
dcl sts  MlmeStatus ;
dcl tBcn, tmax, tmin, tmot  Kusec ;
dcl typeSet  BssTypeSet ;
dcl wake, rdtm  Boolean ;
```

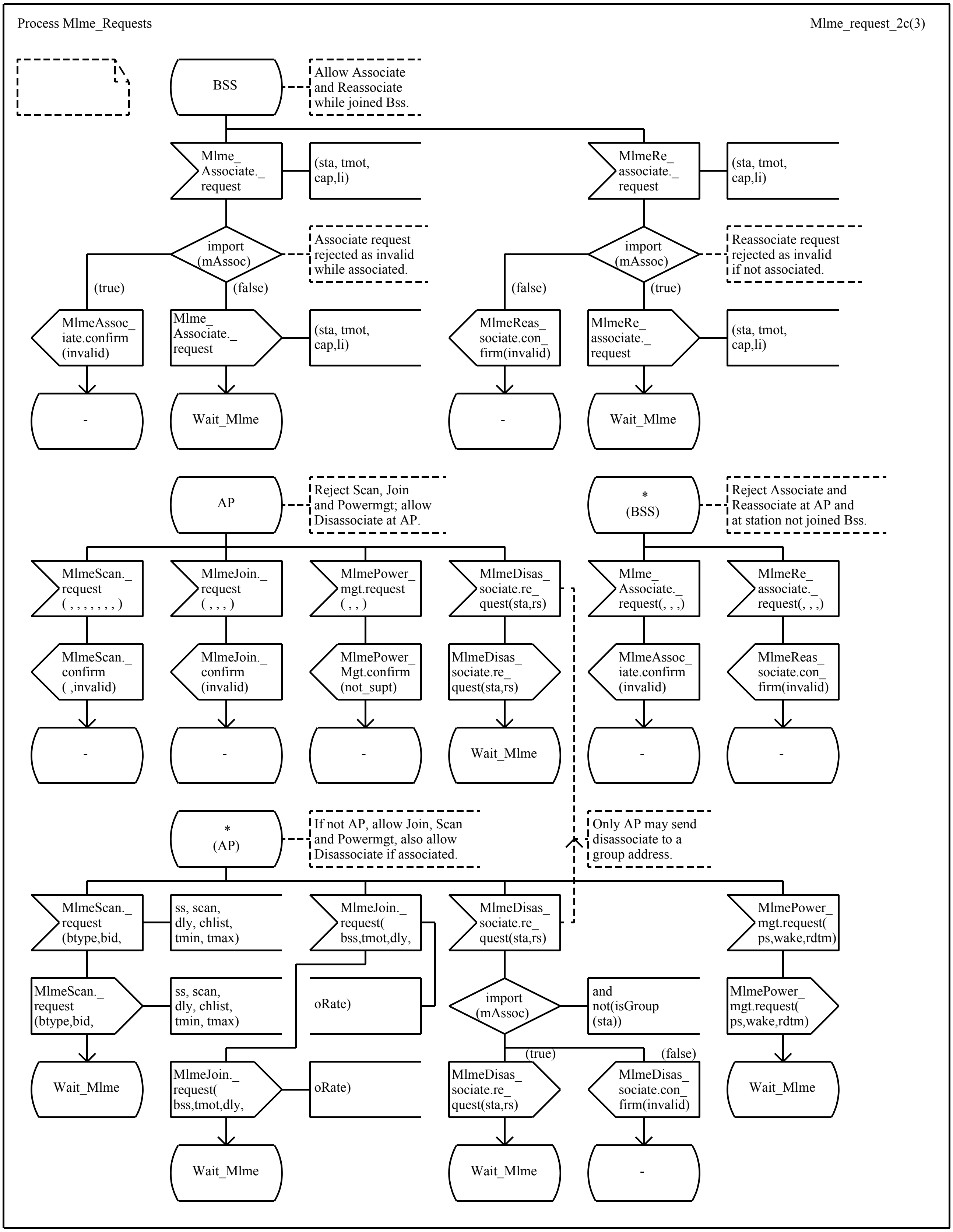
Reject Authenticate, allow Start if idle

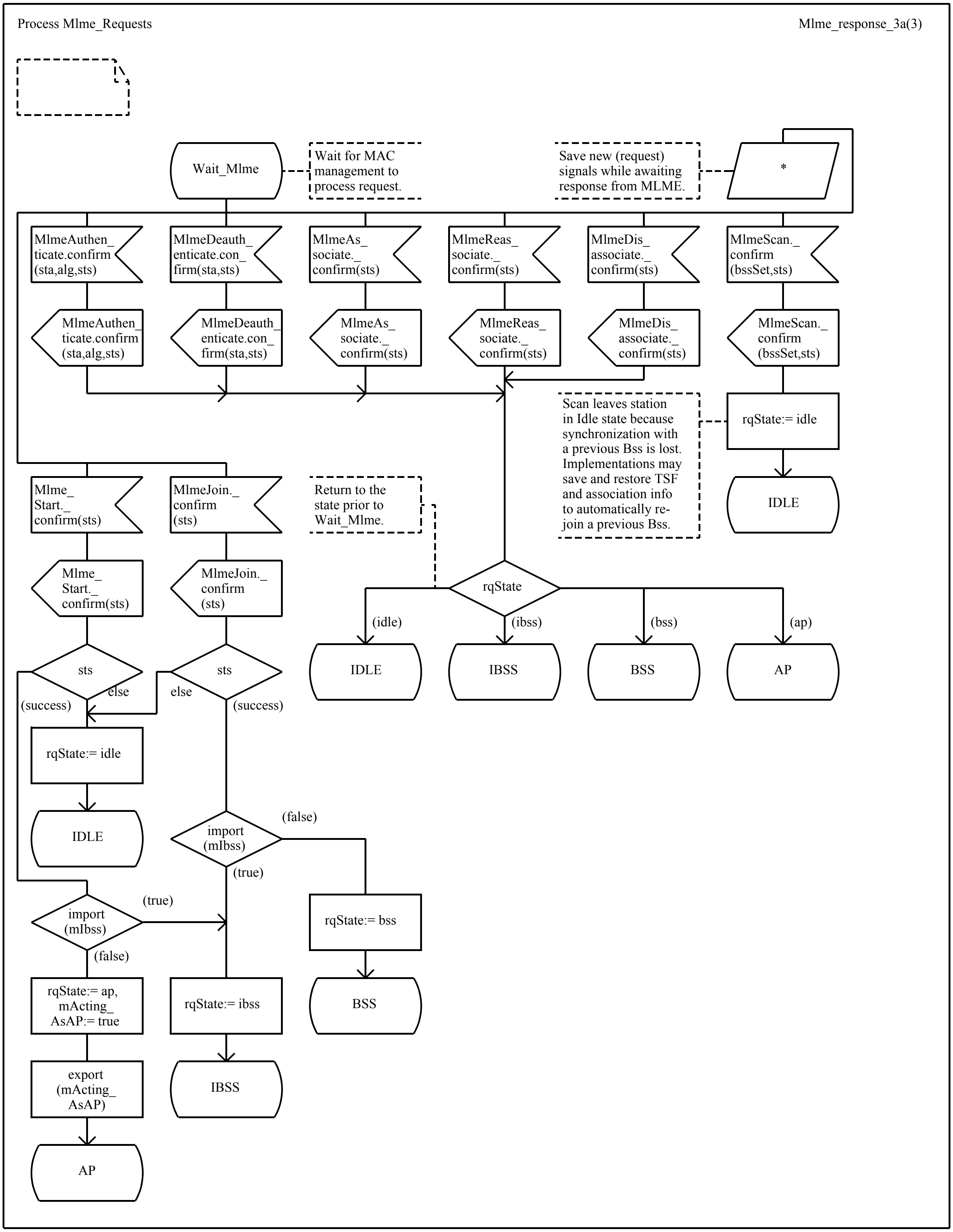
Reject Start if not idle, allow Auth if neither IDLE nor AP.

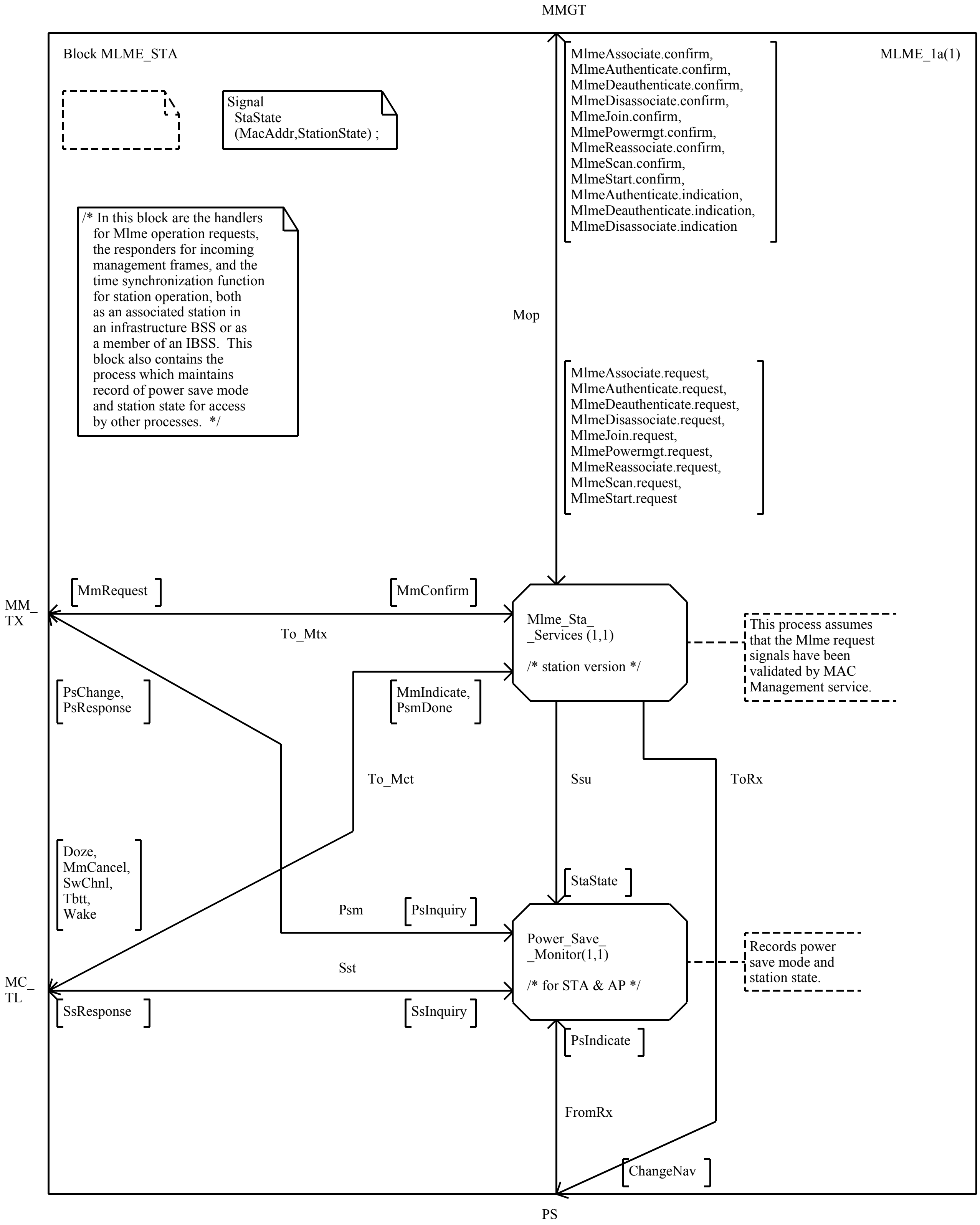
Reject as invalid due to not being in a BSS.

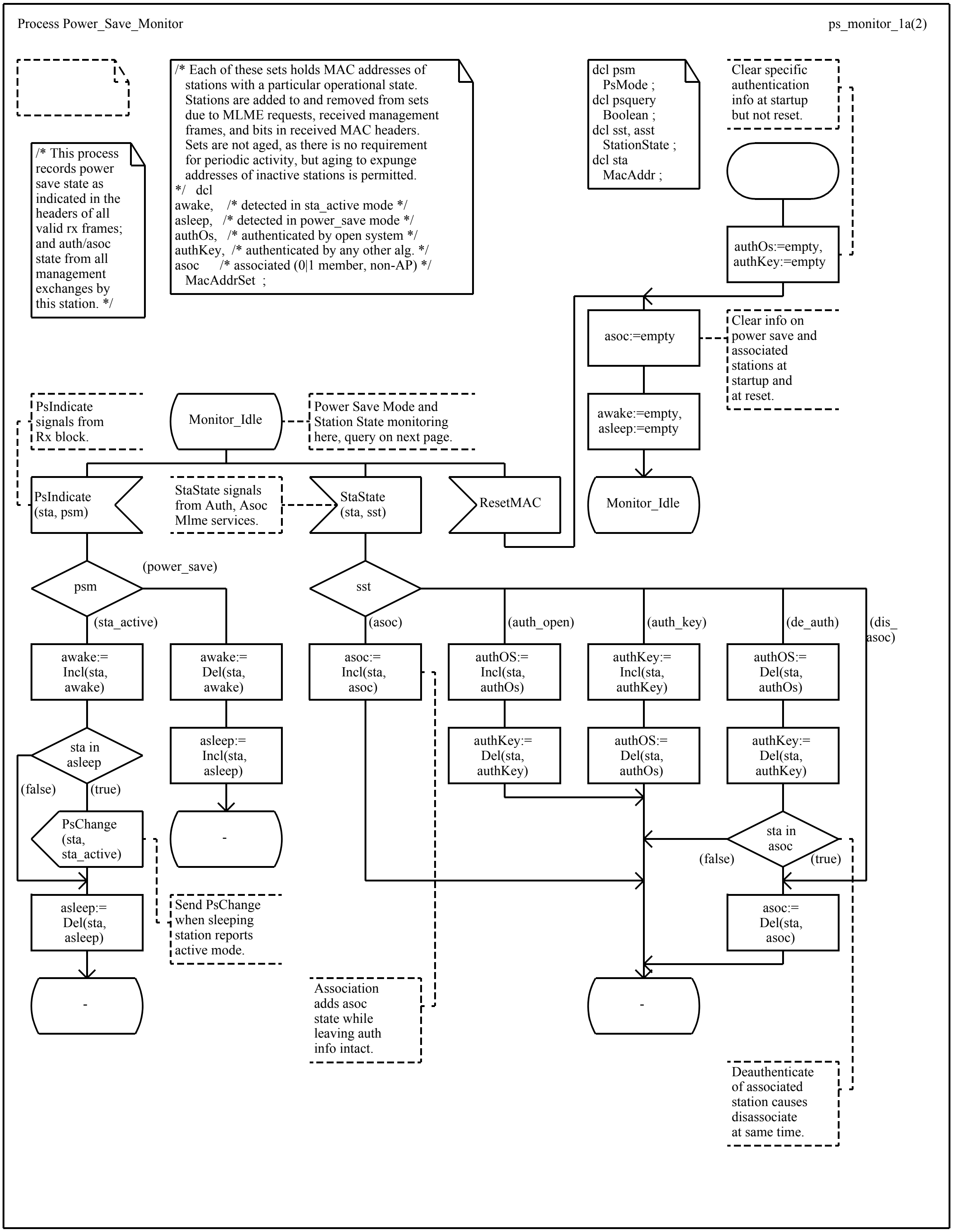
Reset and Deauthenticate always allowed.

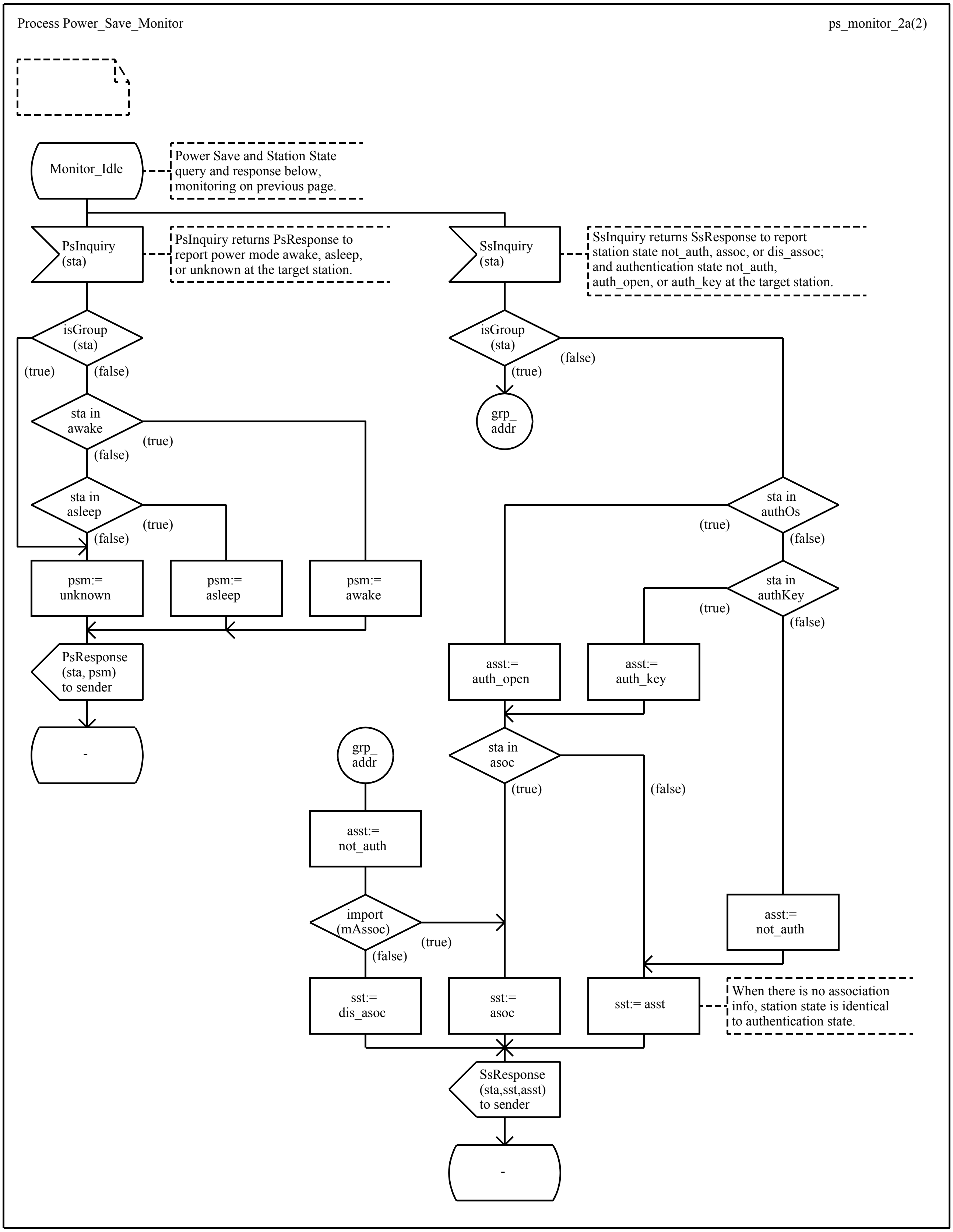
Deauthenticate expunges local authentication record even if there is no BSS for sending the notification.

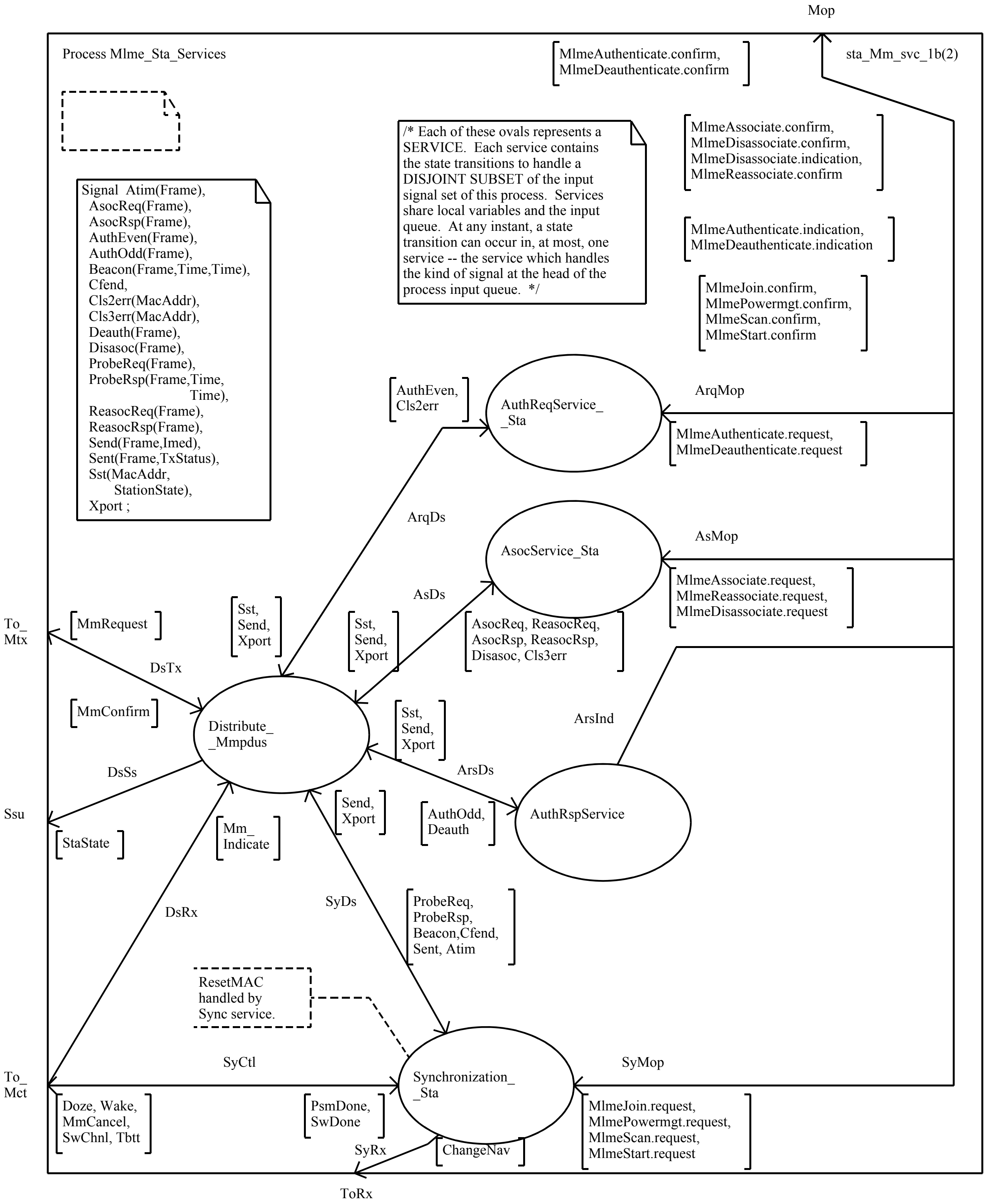

Process Mlme_Requests
Mlme_request_2c(3)
BSS
Allow Associate and Reassociate while joined Bss.
Mlme_ Associate._ request
(sta, tmot, cap,li)
MlmeRe_ associate._ request
(sta, tmot, cap,li)
import (mAssoc)
Associate request rejected as invalid while associated.
(true)
(false)
MlmeAssoc_ iate.confirm (invalid)
Mlme_ Associate._ request
(sta, tmot, cap,li)
-
Wait_Mlme
import (mAssoc)
Reassociate request rejected as invalid if not associated.
(false)
(true)
MlmeReas_ sociate.con_ firm(invalid)
MlmeRe_ associate._ request
(sta, tmot, cap,li)
-
Wait_Mlme
AP
Reject Scan, Join and Powermgt; allow Disassociate at AP.
MlmeScan._ request (, , , , , , ,)
MlmeJoin._ request (, , ,)
MlmePower_ mgt.request (, ,)
MlmeDisas_ sociate.re_ quest(sta,rs)
MlmeScan._ confirm (,invalid)
MlmeJoin._ confirm (invalid)
MlmePower_ Mgt.confirm (not_supt)
MlmeDisas_ sociate.re_ quest(sta,rs)
-
-
-
Wait_Mlme
* (BSS)
Reject Associate and Reassociate at AP and at station not joined Bss.
Mlme_ Associate._ request(, ,)
MlmeRe_ associate._ request(, ,)
MlmeAssoc_ iate.confirm (invalid)
MlmeReas_ sociate.con_ firm(invalid)
-
-
* (AP)
If not AP, allow Join, Scan and Powermgt, also allow Disassociate if associated.
Only AP may send disassociate to a group address.
MlmeScan._ request (btype,bid,
ss, scan, dly, chlist, tmin, tmax)
MlmeJoin._ request(bss,tmot,dly,
oRate)
MlmeDisas_ sociate.re_ quest(sta,rs)
MlmePower_ mgt.request(ps,wake,rdtm)
MlmeScan._ request (btype,bid,
ss, scan, dly, chlist, tmin, tmax)
import (mAssoc)
and not(isGroup (sta))
MlmePower_ mgt.request(ps,wake,rdtm)
(true)
(false)
Wait_Mlme
MlmeJoin._ request(bss,tmot,dly,
oRate)
MlmeDisas_ sociate.re_ quest(sta,rs)
MlmeDisas_ sociate.con_ firm(invalid)
Wait_Mlme
Wait_Mlme
Wait_Mlme
Wait_Mlme
-


Process Mlme_Requests
Mlme_response_3a(3)
Wait_Mlme
Wait for MAC management to process request.
Save new (request) signals while awaiting response from MLME.
*
MlmeAuthen_ ticate.confirm (sta,alg,sts)
MlmeDeauth_ enticate.con_ firm(sta,sts)
MlmeAs_ sociate._ confirm(sts)
MlmeReas_ sociate._ confirm(sts)
MlmeDis_ associate._ confirm(sts)
MlmeScan._ confirm (bssSet,sts)
MlmeAuthen_ ticate.confirm (sta,alg,sts)
MlmeDeauth_ enticate.con_ firm(sta,sts)
MlmeAs_ sociate._ confirm(sts)
MlmeReas_ sociate._ confirm(sts)
MlmeDis_ associate._ confirm(sts)
MlmeScan._ confirm (bssSet,sts)
Scan leaves station in Idle state because synchronization with a previous Bss is lost. Implementations may save and restore TSF and association info to automatically re-join a previous Bss.
rqState:= idle
IDLE
Mlme_ Start._ confirm(sts)
MlmeJoin._ confirm (sts)
Return to the state prior to Wait_Mlme.
Mlme_ Start._ confirm(sts)
MlmeJoin._ confirm (sts)
rqState
(idle)
(ibss)
(bss)
(ap)
IDLE
IBSS
BSS
AP
sts
else
(success)
sts
else
(success)
rqState:= idle
IDLE
import (mIbss)
(false)
(true)
import (mIbss)
(true)
(false)
rqState:= bss
BSS
rqState:= ap, mActing_ AsAP:= true
rqState:= ibss
export (mActing_ AsAP)
IBSS
AP


MMGT
Block MLME_STA
MLME_1a(1)
Signal
StaState
(MacAddr,StationState) ;
/* In this block are the handlers for Mlme operation requests, the responders for incoming management frames, and the time synchronization function for station operation, both as an associated station in an infrastructure BSS or as a member of an IBSS. This block also contains the process which maintains record of power save mode and station state for access by other processes. */
MlmeAssociate.confirm,
MlmeAuthenticate.confirm,
MlmeDeauthenticate.confirm,
MlmeDisassociate.confirm,
MlmeJoin.confirm,
MlmePowermgt.confirm,
MlmeReassociate.confirm,
MlmeScan.confirm,
MlmeStart.confirm,
MlmeAuthenticate.indication,
MlmeDeauthenticate.indication,
MlmeDisassociate.indication
Mop
MlmeAssociate.request,
MlmeAuthenticate.request,
MlmeDeauthenticate.request,
MlmeDisassociate.request,
MlmeJoin.request,
MlmePowermgt.request,
MlmeReassociate.request,
MlmeScan.request,
MlmeStart.request
MM_
TX
MmRequest
MmConfirm
To_Mtx
Mlme_Sta_
_Services (1,1)
/* station version */
This process assumes that the Mlme request signals have been validated by MAC Management service.
PsChange,
PsResponse
MmIndicate,
PsmDone
To_Mct
Ssu
ToRx
Doze,
MmCancel,
SwChnl,
Tbtt,
Wake
StaState
Psm
PsInquiry
Power_Save_
_Monitor(1,1)
/* for STA & AP */
Records power save mode and station state.
Sst
MC_
TL
SsResponse
SsInquiry
PsIndicate
FromRx
ChangeNav
PS


Process Power_Save_Monitor
ps_monitor_1a(2)
/* Each of these sets holds MAC addresses of stations with a particular operational state. Stations are added to and removed from sets due to MLME requests, received management frames, and bits in received MAC headers. Sets are not aged, as there is no requirement for periodic activity, but aging to expunge addresses of inactive stations is permitted. */ dcl
awake, /* detected in sta_active mode */
asleep, /* detected in power_save mode */
authOs, /* authenticated by open system */
authKey, /* authenticated by any other alg. */
asoc /* associated (0|1 member, non-AP) */
MacAddrSet ;
dcl psm PsMode ;
dcl psquery Boolean ;
dcl sst, asst StationState ;
dcl sta MacAddr ;
Clear specific authentication info at startup but not reset.
authOs:=empty,
authKey:=empty
/* This process records power save state as indicated in the headers of all valid rx frames; and auth/asoc state from all management exchanges by this station. */
asoc:=empty
Clear info on power save and associated stations at startup and at reset.
awake:=empty,
asleep:=empty
Monitor_Idle
PsIndicate signals from Rx block.
Monitor_Idle
Power Save Mode and Station State monitoring here, query on next page.
PsIndicate
(sta, psm)
StaState signals from Auth, Asoc Mlme services.
StaState
(sta, sst)
ResetMAC
psm
(power_save)
(sta_active)
sst
(asoc)
(auth_open)
(auth_key)
(de_auth)
(dis_
asoc)
awake:=
Incl(sta,
awake)
awake:=
Del(sta,
awake)
asoc:=
Incl(sta,
asoc)
authOS:=
Incl(sta,
authOs)
authKey:=
Incl(sta,
authKey)
authOS:=
Del(sta,
authOs)
sta in
asleep
asleep:=
Incl(sta,
asleep)
authKey:=
Del(sta,
authKey)
authOS:=
Del(sta,
authOs)
authKey:=
Del(sta,
authKey)
(false)
(true)
PsChange
(sta,
sta_active)
-
sta in
asoc
(false)
(true)
asleep:=
Del(sta,
asleep)
Send PsChange when sleeping station reports active mode.
asoc:=
Del(sta,
asoc)
-
Association adds asoc state while leaving auth info intact.
-
Deauthenticate of associated station causes disassociate at same time.


Process Power_Save_Monitor
ps_monitor_2a(2)
Monitor_Idle
Power Save and Station State query and response below, monitoring on previous page.
PsInquiry (sta)
PsInquiry returns PsResponse to report power mode awake, asleep, or unknown at the target station.
SsInquiry (sta)
SsInquiry returns SsResponse to report station state not_auth, assoc, or dis_assoc; and authentication state not_auth, auth_open, or auth_key at the target station.
isGroup (sta)
(true)
(false)
sta in awake
(true)
(false)
sta in asleep
(true)
(false)
psm:= unknown
psm:= asleep
psm:= awake
PsResponse (sta, psm) to sender
-
isGroup (sta)
(false)
(true)
grp_ addr
sta in authOs
(true)
(false)
sta in authKey
(true)
(false)
asst:= auth_open
asst:= auth_key
sta in asoc
(true)
(false)
grp_ addr
asst:= not_auth
import (mAssoc)
(true)
(false)
asst:= not_auth
sst:= dis_asoc
sst:= asoc
sst:= asst
When there is no association info, station state is identical to authentication state.
SsResponse (sta,sst,asst) to sender
-

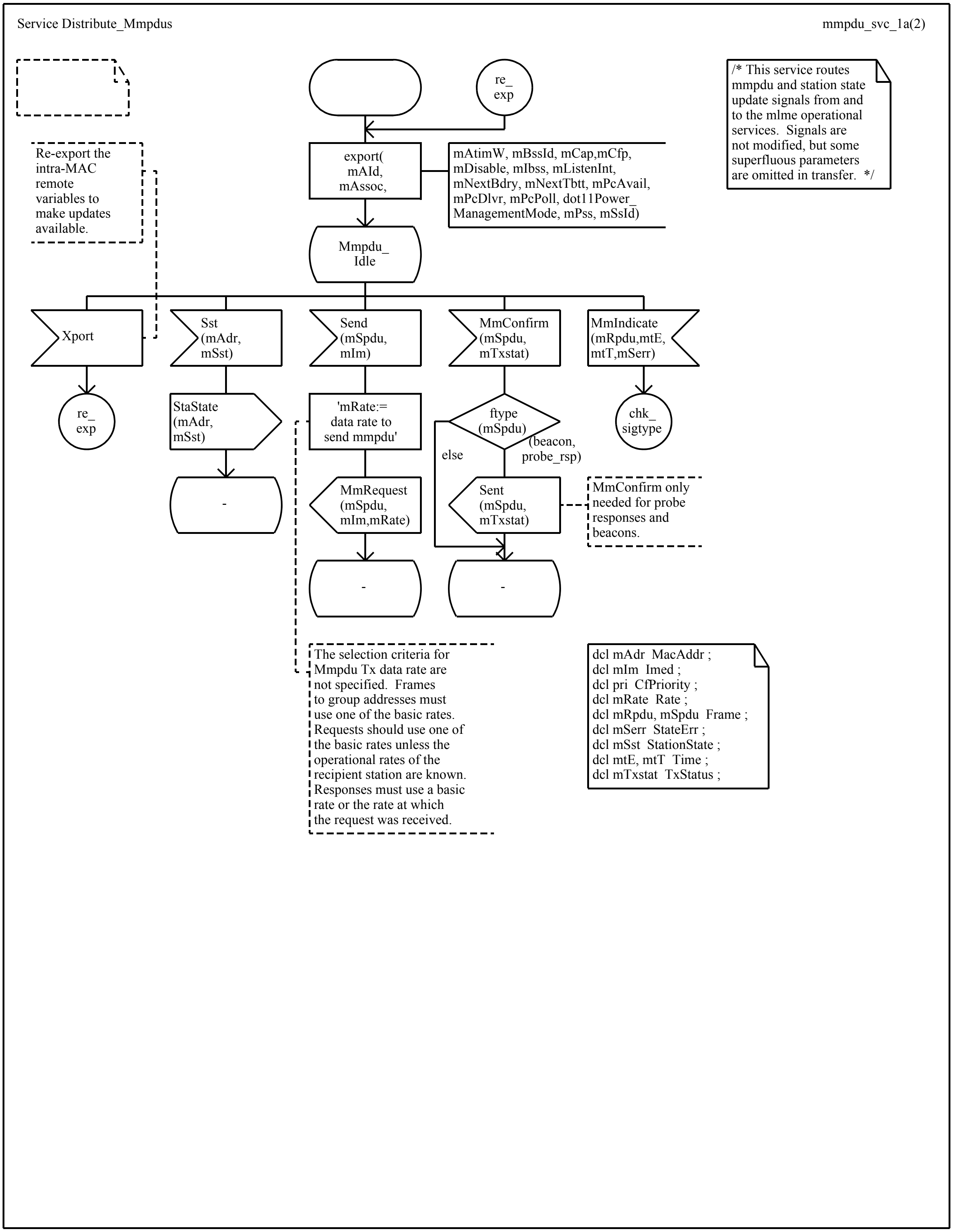

Mop
Process Mlme_Sta_Services
sta_Mm_svc_1b(2)
MlmeAuthenticate.confirm,
MlmeDeauthenticate.confirm
/* Each of these ovals represents a SERVICE. Each service contains the state transitions to handle a DISJOINT SUBSET of the input signal set of this process. Services share local variables and the input queue. At any instant, a state transition can occur in, at most, one service -- the service which handles the kind of signal at the head of the process input queue. */
MlmeAssociate.confirm,
MlmeDisassociate.confirm,
MlmeDisassociate.indication,
MlmeReassociate.confirm
MlmeAuthenticate.indication,
MlmeDeauthenticate.indication
MlmeJoin.confirm,
MlmePowermgt.confirm,
MlmeScan.confirm,
MlmeStart.confirm
Signal Atim(Frame),
AsocReq(Frame),
AsocRsp(Frame),
AuthEven(Frame),
AuthOdd(Frame),
Beacon(Frame,Time,Time),
Cfend,
Cls2err(MacAddr),
Cls3err(MacAddr),
Deauth(Frame),
Disasoc(Frame),
ProbeReq(Frame),
ProbeRsp(Frame,Time,
Time),
ReasocReq(Frame),
ReasocRsp(Frame),
Send(Frame,Imed),
Sent(Frame,TxStatus),
Sst(MacAddr,
StationState),
Xport ;
AuthEven,
Cls2err
AuthReqService_
_Sta
ArqMop
MlmeAuthenticate.request,
MlmeDeauthenticate.request
ArqDs
AsocService_Sta
AsMop
MlmeAssociate.request,
MlmeReassociate.request,
MlmeDisassociate.request
AsDs
Sst,
Send,
Xport
Sst,
Send,
Xport
AsocReq, ReasocReq,
AsocRsp, ReasocRsp,
Disasoc, Cls3err
To_
Mtx
MmRequest
DsTx
MmConfirm
Distribute_
_Mmpdus
Sst,
Send,
Xport
ArsInd
ArsDs
DsSs
Ssu
StaState
Mm_
Indicate
Send,
Xport
AuthOdd,
Deauth
AuthRspService
DsRx
SyDs
ProbeReq,
ProbeRsp,
Beacon,Cfend,
Sent, Atim
ResetMAC
handled by
Sync service.
SyCtl
SyMop
To_
Mct
Synchronization_
_Sta
Doze, Wake,
MmCancel,
SwChnl, Tbtt
PsmDone,
SwDone
MlmeJoin.request,
MlmePowermgt.request,
MlmeScan.request,
MlmeStart.request
SyRx
ChangeNav
ToRx

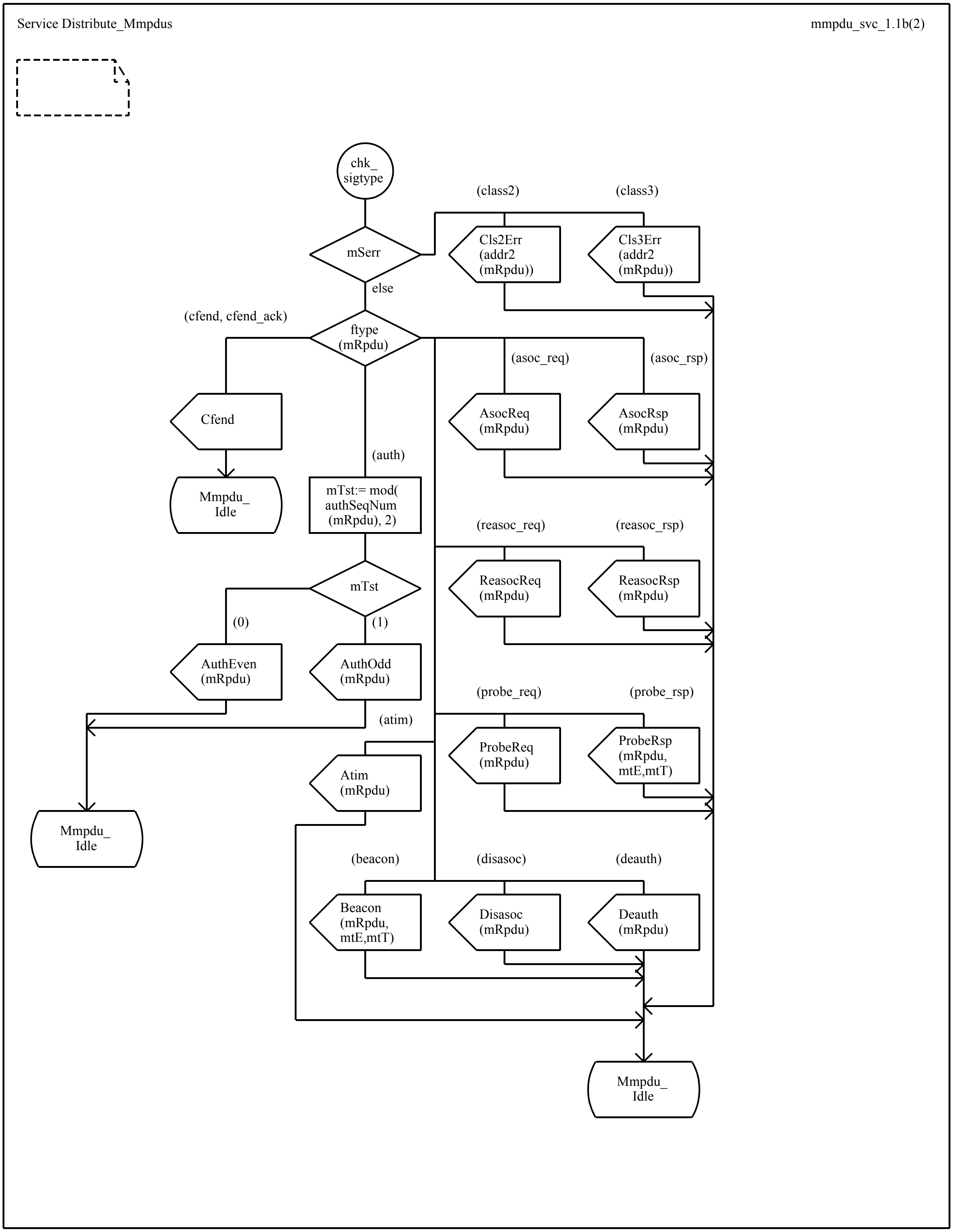
Process Mlme_Sta_Services

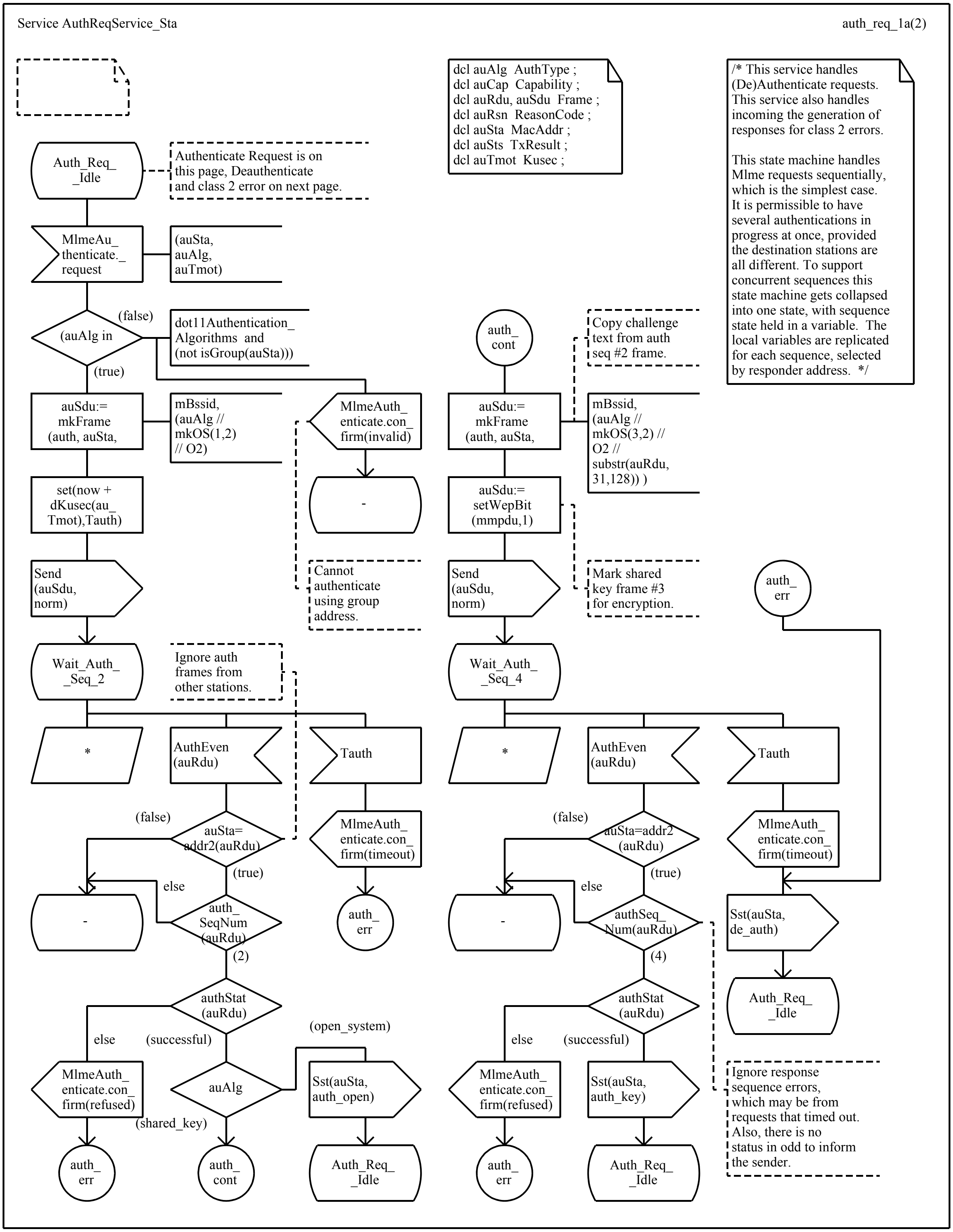
sta_Mm_svc_1.1b(2)

```
Timer Tasoc,
  Tauth, Tchal,
  Tbcn, Tatim ;
```

```
/* Intra-MAC remote variables */
dcl exported
dot11PowerManagementMode  PwrSave:= sta_active,
dot11DesiredSsid  Octetstring,
dot11DesiredBssType,
dot11OperationalRateSet  Ratestring:= mkOS(2,1),
dot11BeaconPeriod  TU,
dot11DtimPeriod  Integer:= 1,
dot11AssociationResponseTimeOut  TU,
dot11MultiDomainCapabilityEnabled  Boolean:= false,
mAId  AsocId:= 0,
mAssoc  Boolean:= false,
mAtimW  Boolean:= false,
mBrates Ratestring:= mkOS(2,1),
mBssId  MacAddr:= nullAddr,
mCap  Octetstring:= O2,
mCfp  Boolean:= false,
mDisable  Boolean:= true,
mDtimCount  Integer:= 1,
mIbss  Boolean:= false,
mNextBdry  Time:= 0,
mNextTbtt  Time:= 0,
mPcAvail  Boolean:= false,
mPcPoll  Boolean:= false,
mPdly  Usec:= 0,
mPss  PsState:= awake,
mSsId  Octetstring:= null;
```


Service Distribute_Mmpdus
mmpdu_svc_1a(2)
re_
exp
/* This service routes mmpdu and station state update signals from and to the mlme operational services. Signals are not modified, but some superfluous parameters are omitted in transfer. */
Re-export the intra-MAC remote variables to make updates available.
export(
mAId,
mAssoc,
mAtimW, mBssId, mCap,mCfp, mDisable, mIbss, mListenInt, mNextBdry, mNextTbtt, mPcAvail, mPcDlvr, mPcPoll, dot11Power_ ManagementMode, mPss, mSsId)
Mmpdu_
Idle
Xport
Sst
(mAdr,
mSst)
Send
(mSpdu,
mIm)
MmConfirm
(mSpdu,
mTxstat)
MmIndicate
(mRpdu,mtE,
mtT,mSerr)
re_
exp
StaState
(mAdr,
mSst)
'mRate:=
data rate to
send mmpdu'
ftype
(mSpdu)
else
(beacon,
probe_rsp)
chk_
sigtype
-
MmRequest
(mSpdu,
mIm,mRate)
Sent
(mSpdu,
mTxstat)
MmConfirm only needed for probe responses and beacons.
-
-
The selection criteria for Mmpdu Tx data rate are not specified. Frames to group addresses must use one of the basic rates. Requests should use one of the basic rates unless the operational rates of the recipient station are known. Responses must use a basic rate or the rate at which the request was received.
dcl mAdr MacAddr ;
dcl mIm Imed ;
dcl pri CfPriority ;
dcl mRate Rate ;
dcl mRpdu, mSpdu Frame ;
dcl mSerr StateErr ;
dcl mSst StationState ;
dcl mtE, mtT Time ;
dcl mTxstat TxStatus ;


Service Distribute_Mmpdus
mmpdu_svc_1.1b(2)
chk_ sigtype
mSerr
(class2)
(class3)
Cls2Err (addr2 (mRpdu))
Cls3Err (addr2 (mRpdu))
else
(cfend, cfend_ack)
ftype (mRpdu)
(asoc_req)
(asoc_rsp)
Cfend
AsocReq (mRpdu)
AsocRsp (mRpdu)
(auth)
Mmpdu_ Idle
mTst:= mod(authSeqNum (mRpdu), 2)
(reasoc_req)
(reasoc_rsp)
mTst
ReasocReq (mRpdu)
ReasocRsp (mRpdu)
(0)
(1)
AuthEven (mRpdu)
AuthOdd (mRpdu)
(probe_req)
(probe_rsp)
(atim)
ProbeReq (mRpdu)
ProbeRsp (mRpdu, mtE,mtT)
Atim (mRpdu)
Mmpdu_ Idle
(beacon)
(disasoc)
(deauth)
Beacon (mRpdu, mtE,mtT)
Disasoc (mRpdu)
Deauth (mRpdu)
Mmpdu_ Idle



Authenticate Request is on this page, Deauthenticate and class 2 error on next page.

Cannot authenticate using group address.

Ignore auth frames from other stations.

dcl auAlg AuthType ;
dcl auCap Capability ;
dcl auRdu, auSdu Frame ;
dcl auRsn ReasonCode ;
dcl auSta MacAddr ;
dcl auSts TxResult ;
dcl auTmot Kusec ;

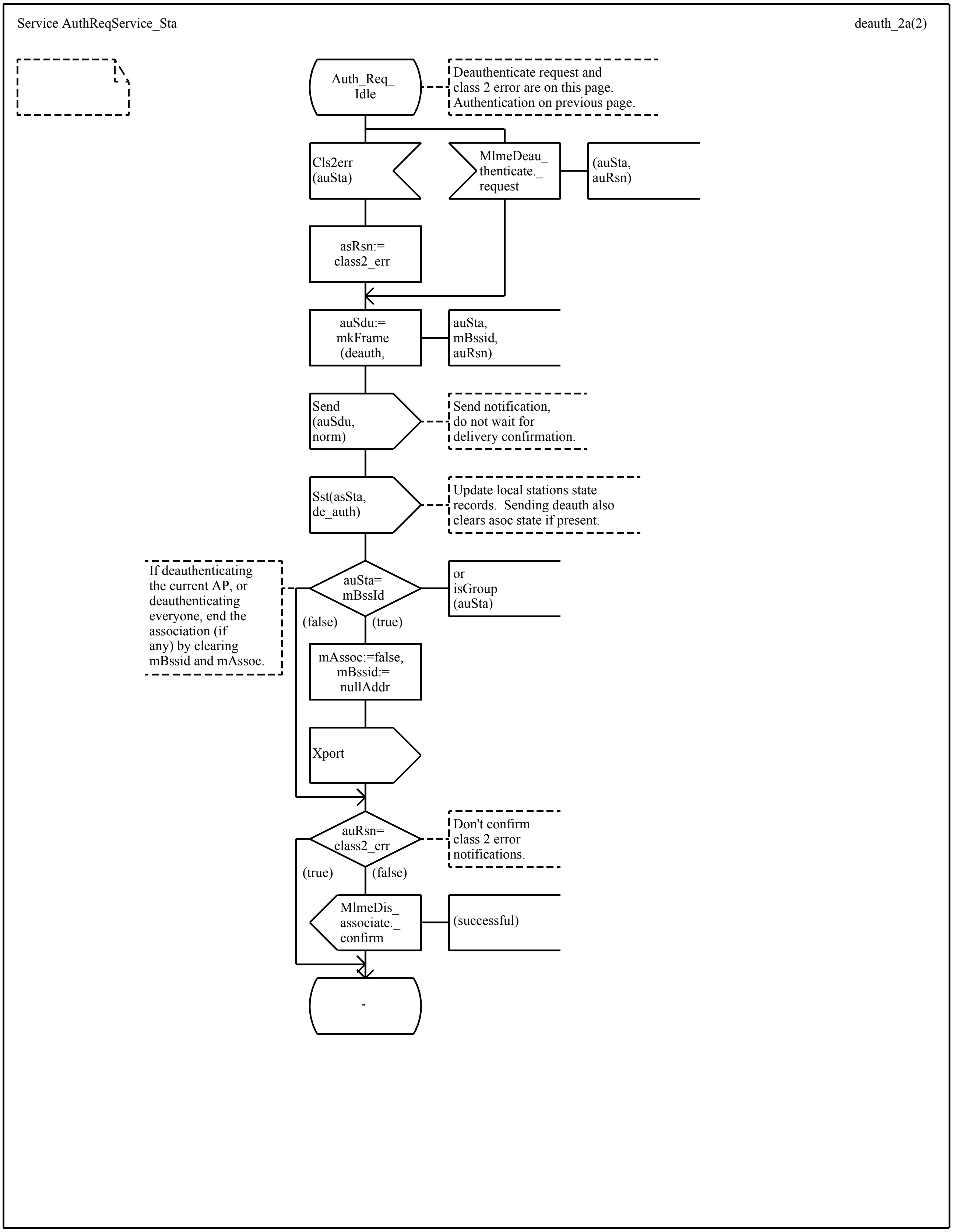
/* This service handles (De)Authenticate requests. This service also handles incoming the generation of responses for class 2 errors.

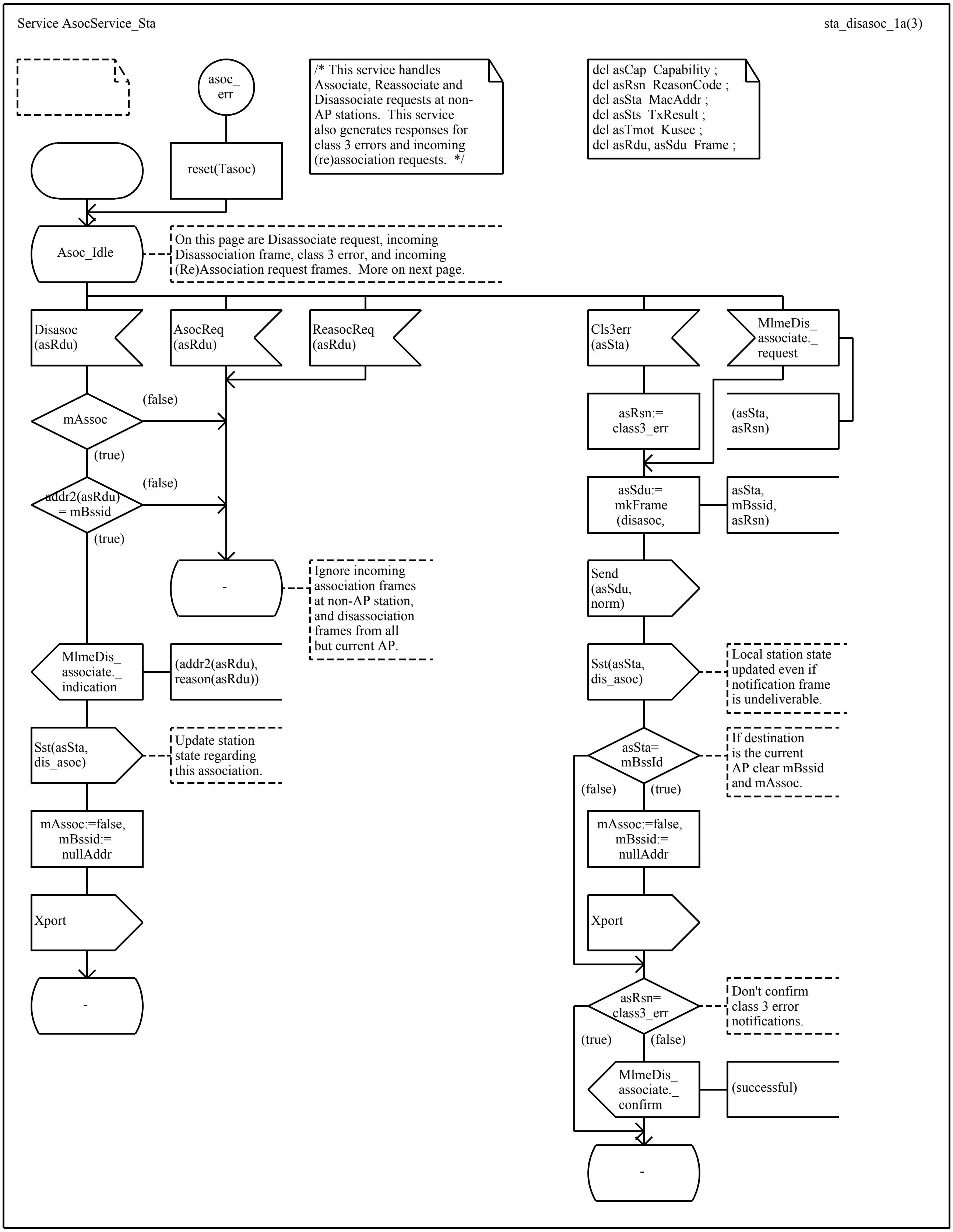
This state machine handles Mlme requests sequentially, which is the simplest case. It is permissible to have several authentications in progress at once, provided the destination stations are all different. To support concurrent sequences this state machine gets collapsed into one state, with sequence state held in a variable. The local variables are replicated for each sequence, selected by responder address. */

Copy challenge text from auth seq #2 frame.

Mark shared key frame #3 for encryption.

Ignore response sequence errors, which may be from requests that timed out. Also, there is no status in odd to inform the sender.


Service AuthReqService_Sta
deauth_2a(2)
Auth_Req_
Idle
Deauthenticate request and
class 2 error are on this page.
Authentication on previous page.
Cls2err
(auSta)
MlmeDeau_
thenticate._
request
(auSta,
auRsn)
asRsn:=
class2_err
auSdu:=
mkFrame
(deauth,
auSta,
mBssid,
auRsn)
Send
(auSdu,
norm)
Send notification,
do not wait for
delivery confirmation.
Sst(asSta,
de_auth)
Update local stations state
records. Sending deauth also
clears asoc state if present.
If deauthenticating
the current AP, or
deauthenticating
everyone, end the
association (if
any) by clearing
mBssid and mAssoc.
auSta=
mBssId
or
isGroup
(auSta)
(false)
(true)
mAssoc:=false,
mBssid:=
nullAddr
Xport
auRsn=
class2_err
Don't confirm
class 2 error
notifications.
(true)
(false)
MlmeDis_
associate._
confirm
(successful)
-



/* This service handles Associate, Reassociate and Disassociate requests at non-AP stations. This service also generates responses for class 3 errors and incoming (re)association requests. */

```
dcl asCap  Capability ;
dcl asRsn  ReasonCode ;
dcl asSta  MacAddr ;
dcl asSts  TxResult ;
dcl asTmot  Kusec ;
dcl asRdu, asSdu  Frame ;
```

On this page are Disassociate request, incoming Disassociation frame, class 3 error, and incoming (Re)Association request frames. More on next page.

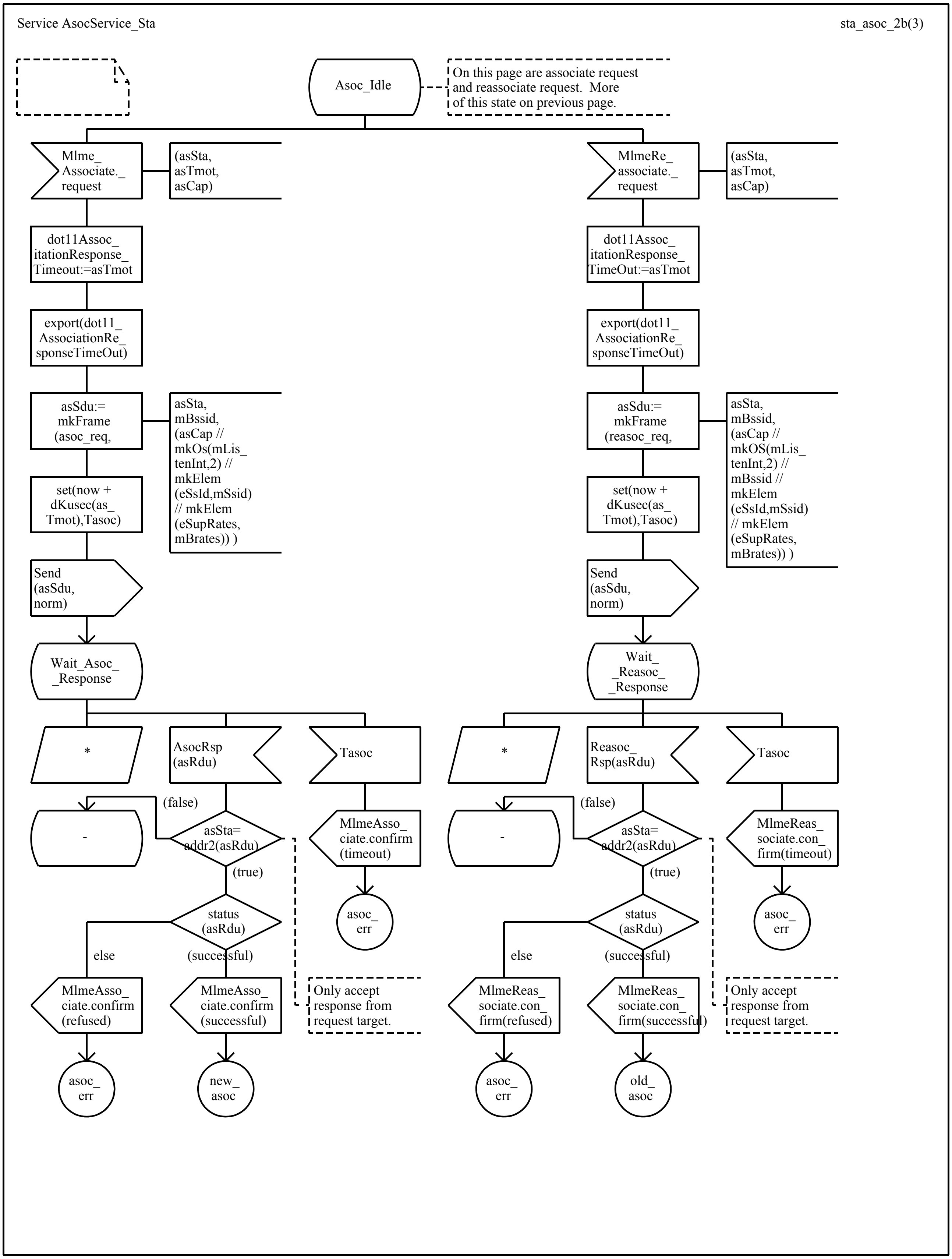
Ignore incoming association frames at non-AP station, and disassociation frames from all but current AP.

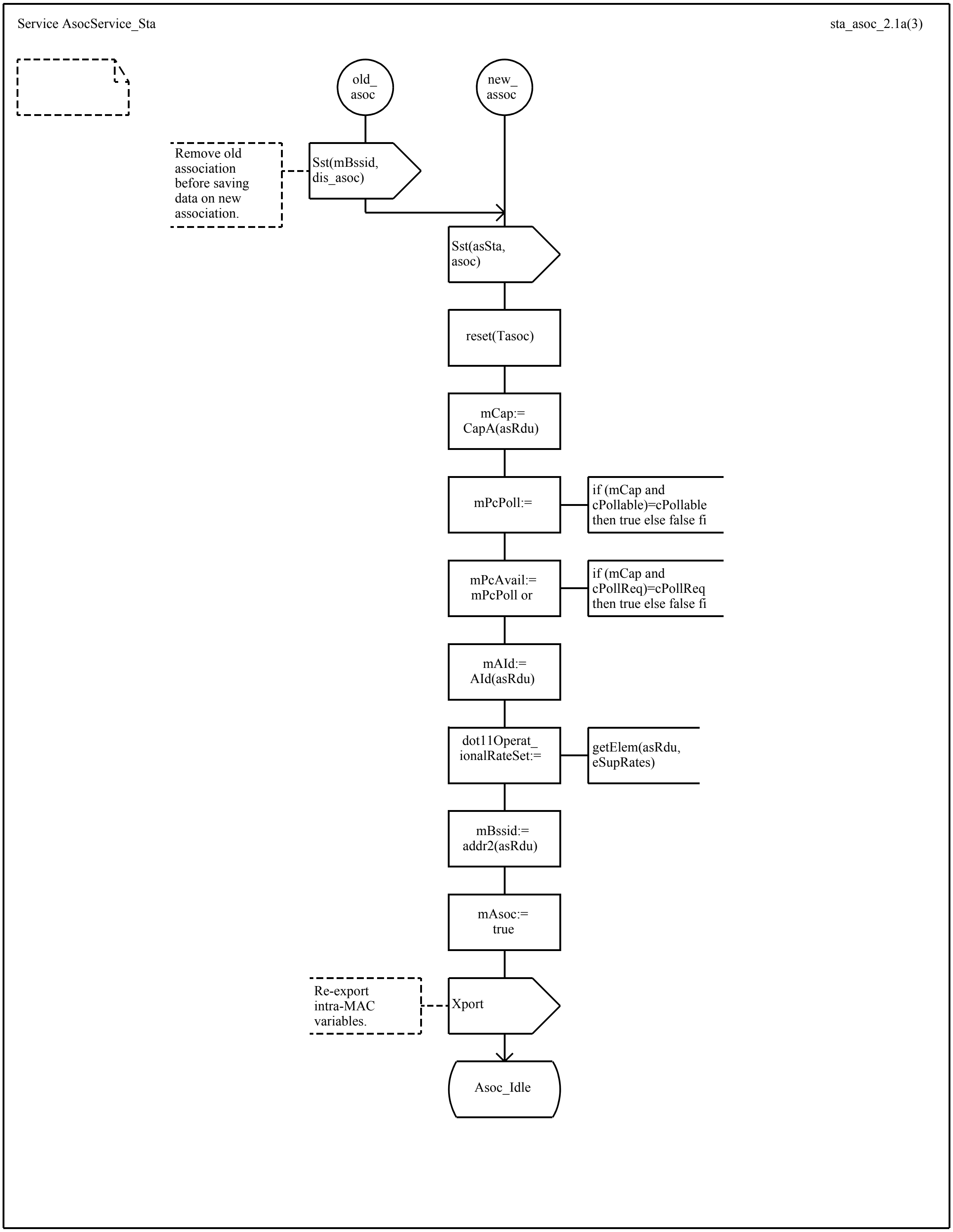
Update station state regarding this association.

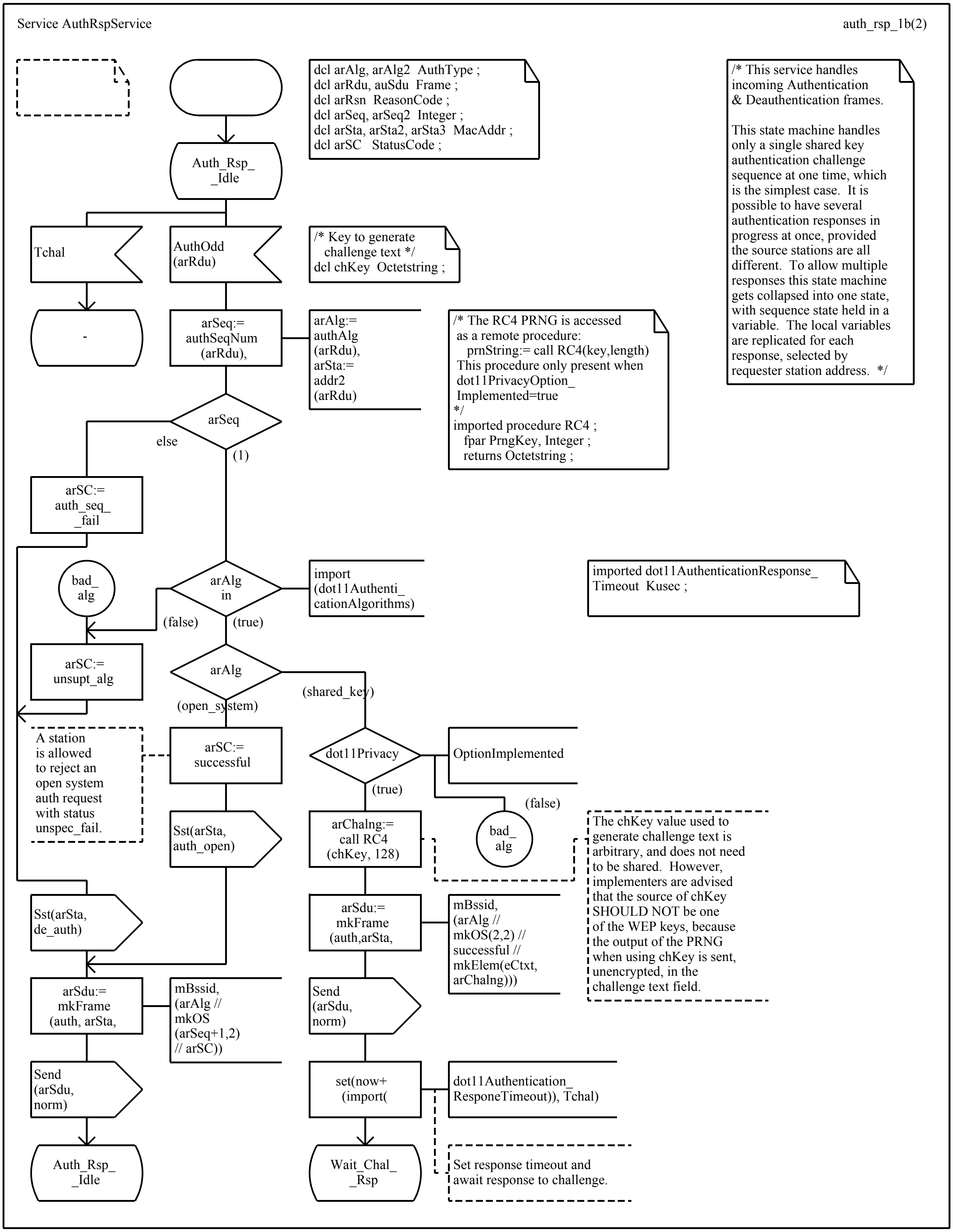
Local station state updated even if notification frame is undeliverable.

If destination is the current AP clear mBssid and mAssoc.

Don't confirm class 3 error notifications.


Service AsocService_Sta
sta_asoc_2b(3)
Asoc_Idle
On this page are associate request and reassociate request. More of this state on previous page.
Mlme_ Associate._ request
(asSta, asTmot, asCap)
dot11Assoc_ itationResponse_ Timeout:=asTmot
export(dot11_ AssociationRe_ sponseTimeOut)
asSdu:= mkFrame (asoc_req,
asSta, mBssid, (asCap // mkOs(mLis_ tenInt,2) // mkElem (eSsId,mSsid) // mkElem (eSupRates, mBrates)))
set(now + dKusec(as_ Tmot),Tasoc)
Send (asSdu, norm)
Wait_Asoc_ _Response
*
-
AsocRsp (asRdu)
(false)
asSta= addr2(asRdu)
(true)
status (asRdu)
else
(successful)
MlmeAsso_ ciate.confirm (refused)
MlmeAsso_ ciate.confirm (successful)
asoc_ err
new_ asoc
Tasoc
MlmeAsso_ ciate.confirm (timeout)
asoc_ err
Only accept response from request target.
MlmeRe_ associate._ request
(asSta, asTmot, asCap)
dot11Assoc_ itationResponse_ TimeOut:=asTmot
export(dot11_ AssociationRe_ sponseTimeOut)
asSdu:= mkFrame (reasoc_req,
asSta, mBssid, (asCap // mkOS(mLis_ tenInt,2) // mBssid // mkElem (eSsId,mSsid) // mkElem (eSupRates, mBrates)))
set(now + dKusec(as_ Tmot),Tasoc)
Send (asSdu, norm)
Wait_ _Reasoc_ _Response
*
-
Reasoc_ Rsp(asRdu)
(false)
asSta= addr2(asRdu)
(true)
status (asRdu)
else
(successful)
MlmeReas_ sociate.con_ firm(refused)
MlmeReas_ sociate.con_ firm(successful)
asoc_ err
old_ asoc
Tasoc
MlmeReas_ sociate.con_ firm(timeout)
asoc_ err
Only accept response from request target.


Service AsocService_Sta
sta_asoc_2.1a(3)
old_ asoc
new_ assoc
Remove old association before saving data on new association.
Sst(mBssid, dis_asoc)
Sst(asSta, asoc)
reset(Tasoc)
mCap:= CapA(asRdu)
mPcPoll:=
if (mCap and cPollable)=cPollable then true else false fi
mPcAvail:= mPcPoll or
if (mCap and cPollReq)=cPollReq then true else false fi
mAId:= AId(asRdu)
dot11Operat_ ionalRateSet:=
getElem(asRdu, eSupRates)
mBssid:= addr2(asRdu)
mAsoc:= true
Re-export intra-MAC variables.
Xport
Asoc_Idle



dcl arAlg, arAlg2 AuthType ;
dcl arRdu, auSdu Frame ;
dcl arRsn ReasonCode ;
dcl arSeq, arSeq2 Integer ;
dcl arSta, arSta2, arSta3 MacAddr ;
dcl arSC StatusCode ;

/* This service handles incoming Authentication & Deauthentication frames.

This state machine handles only a single shared key authentication challenge sequence at one time, which is the simplest case. It is possible to have several authentication responses in progress at once, provided the source stations are all different. To allow multiple responses this state machine gets collapsed into one state, with sequence state held in a variable. The local variables are replicated for each response, selected by requester station address. */

/* Key to generate
challenge text */
dcl chKey Octetstring ;

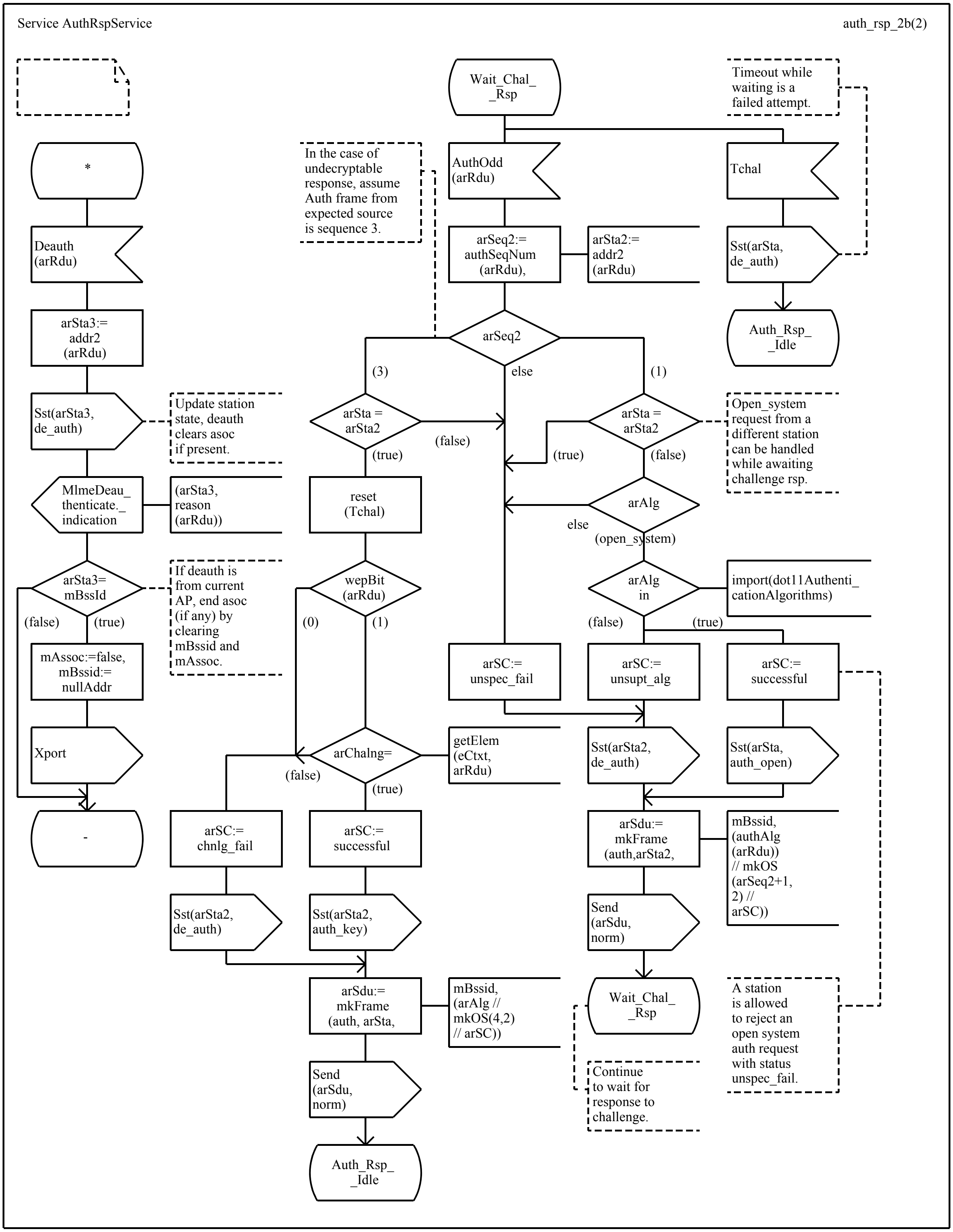
/* The RC4 PRNG is accessed as a remote procedure:
prnString:= call RC4(key,length)
This procedure only present when dot11PrivacyOption_ Implemented=true
*/
imported procedure RC4 ;
fpar PrngKey, Integer ;
returns Octetstring ;

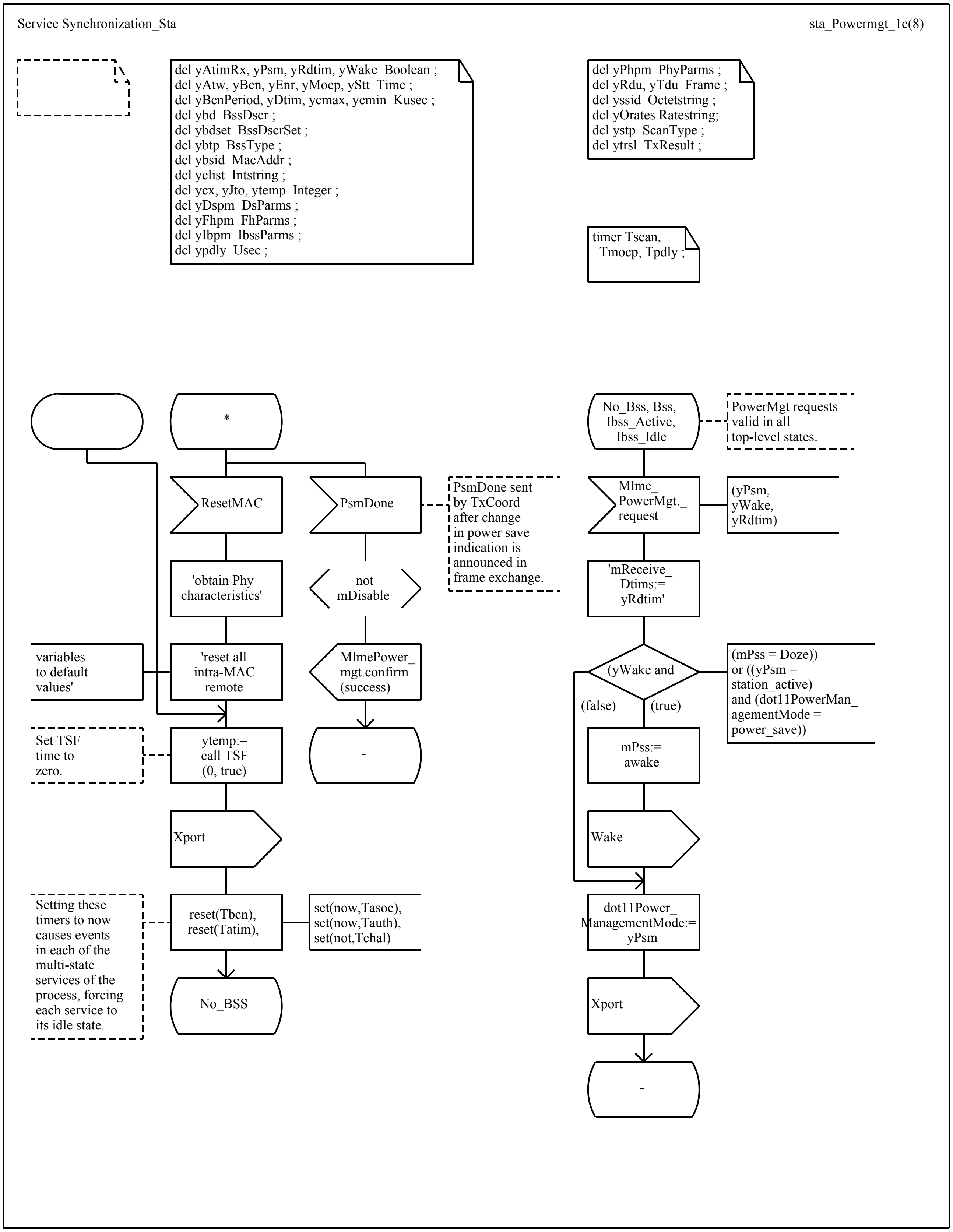
imported dot11AuthenticationResponse_
Timeout Kusec ;

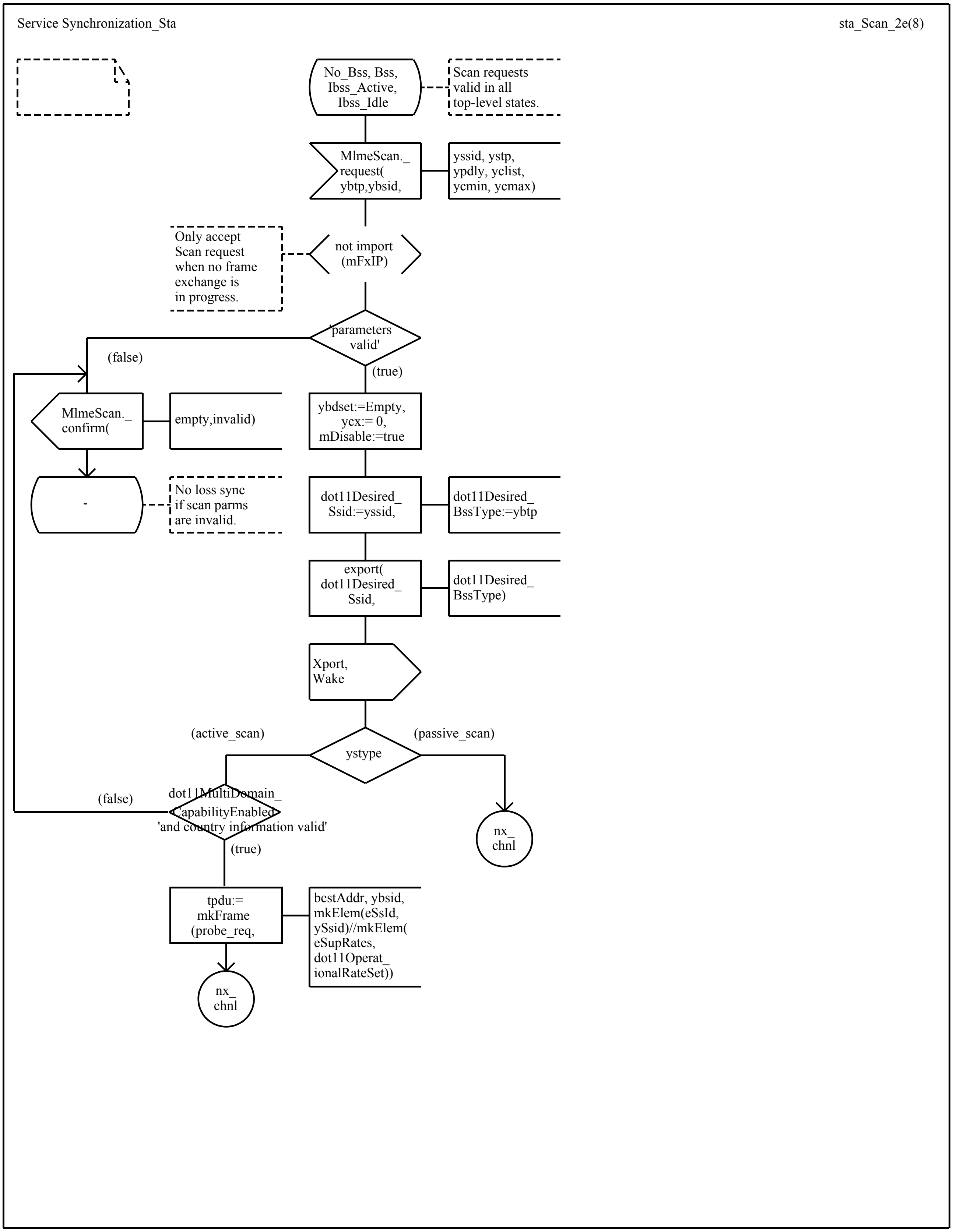
A station is allowed to reject an open system auth request with status unspec_fail.

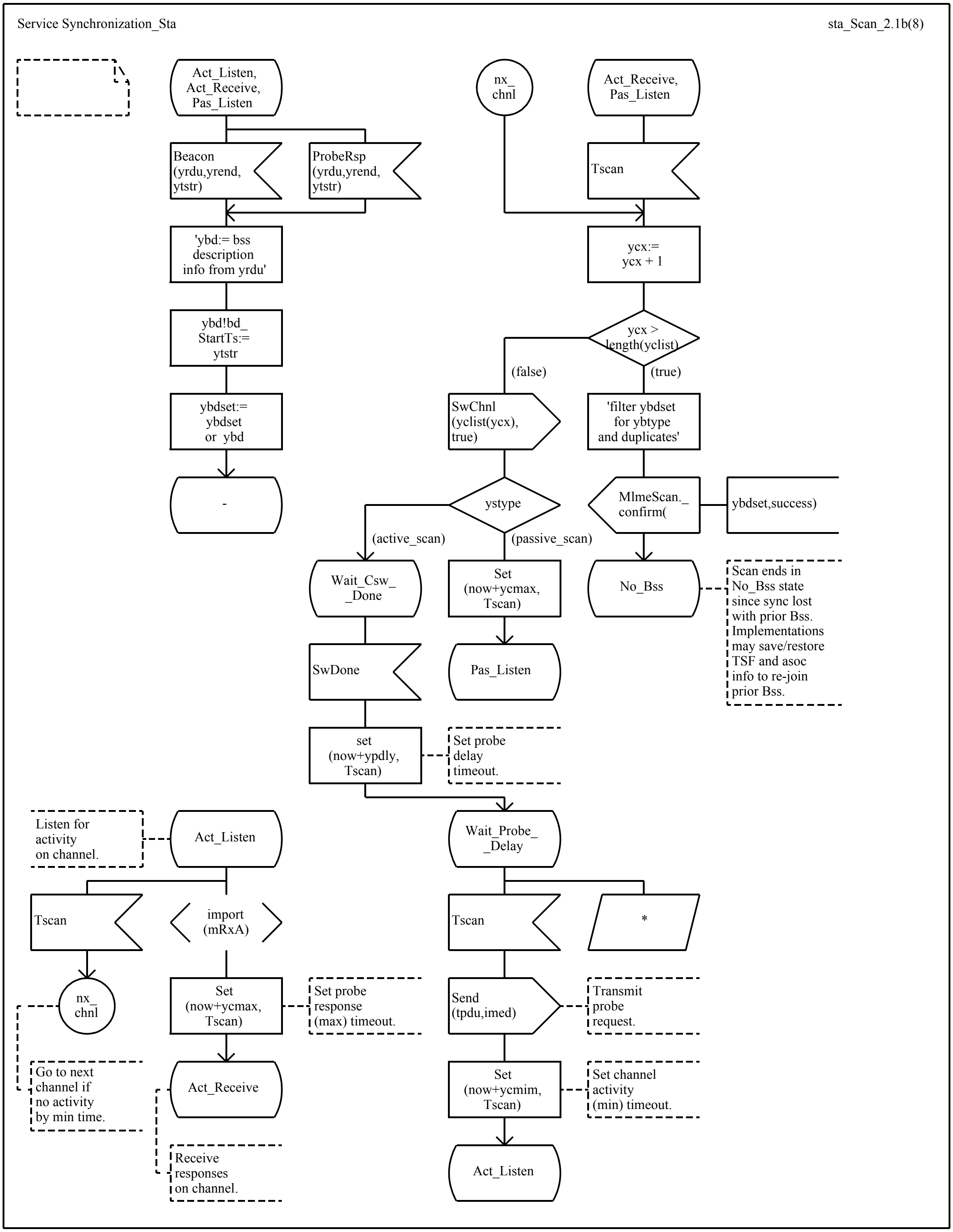
The chKey value used to generate challenge text is arbitrary, and does not need to be shared. However, implementers are advised that the source of chKey SHOULD NOT be one of the WEP keys, because the output of the PRNG when using chKey is sent, unencrypted, in the challenge text field.

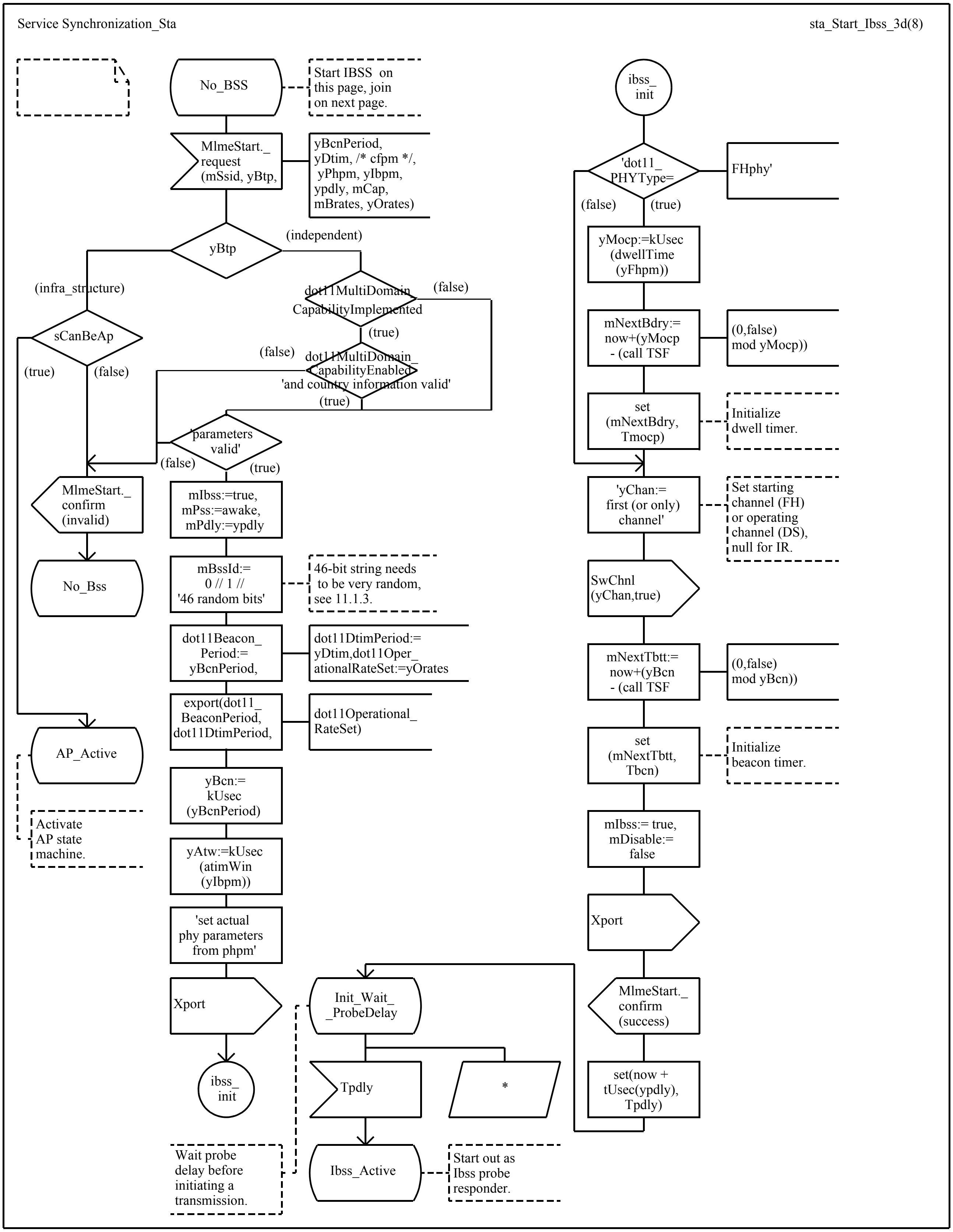
Set response timeout and await response to challenge.

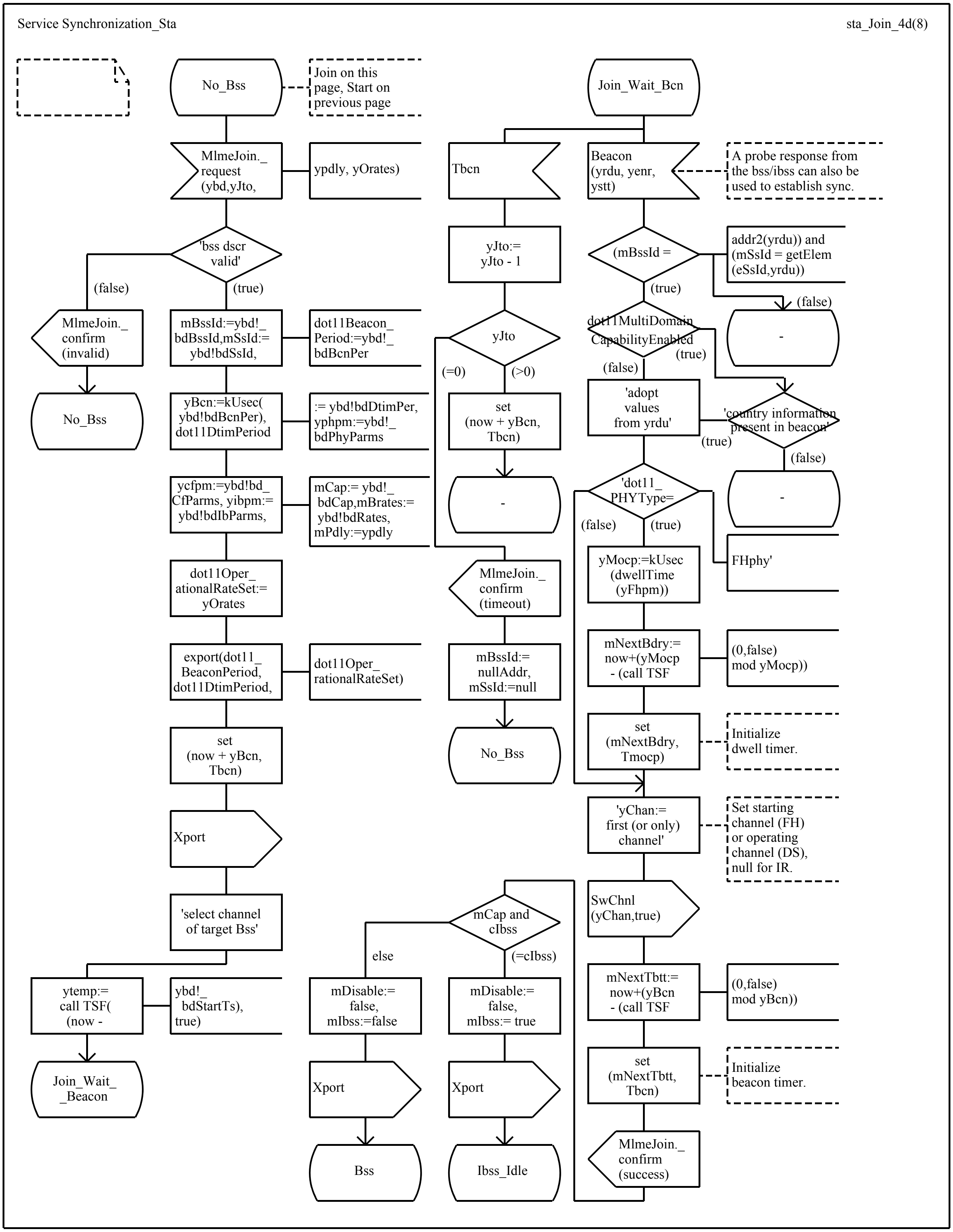

Service AuthRspService
auth_rsp_2b(2)
*
Deauth (arRdu)
arSta3:= addr2 (arRdu)
Sst(arSta3, de_auth)
Update station state, deauth clears asoc if present.
MlmeDeau_ thenticate._ indication
(arSta3, reason (arRdu))
arSta3= mBssId
If deauth is from current AP, end asoc (if any) by clearing mBssid and mAssoc.
(false)
(true)
mAssoc:=false, mBssid:= nullAddr
Xport
-
Wait_Chal_ _Rsp
Timeout while waiting is a failed attempt.
In the case of undecryptable response, assume Auth frame from expected source is sequence 3.
AuthOdd (arRdu)
Tchal
arSeq2:= authSeqNum (arRdu),
arSta2:= addr2 (arRdu)
Sst(arSta, de_auth)
Auth_Rsp_ _Idle
arSeq2
(3)
else
(1)
arSta = arSta2
(false)
(true)
arSta = arSta2
(true)
(false)
Open_system request from a different station can be handled while awaiting challenge rsp.
reset (Tchal)
arAlg
else
(open_system)
wepBit (arRdu)
(0)
(1)
arAlg in
import(dot11Authenti_ cationAlgorithms)
(false)
(true)
arSC:= unspec_fail
arSC:= unsupt_alg
arSC:= successful
arChalng=
getElem (eCtxt, arRdu)
(false)
(true)
Sst(arSta2, de_auth)
Sst(arSta, auth_open)
arSC:= chnlg_fail
arSC:= successful
arSdu:= mkFrame (auth,arSta2,
mBssid, (authAlg (arRdu)) // mkOS (arSeq2+1, 2) // arSC))
Sst(arSta2, de_auth)
Sst(arSta2, auth_key)
Send (arSdu, norm)
arSdu:= mkFrame (auth, arSta,
mBssid, (arAlg // mkOS(4,2) // arSC))
Wait_Chal_ _Rsp
A station is allowed to reject an open system auth request with status unspec_fail.
Continue to wait for response to challenge.
Send (arSdu, norm)
Auth_Rsp_ _Idle

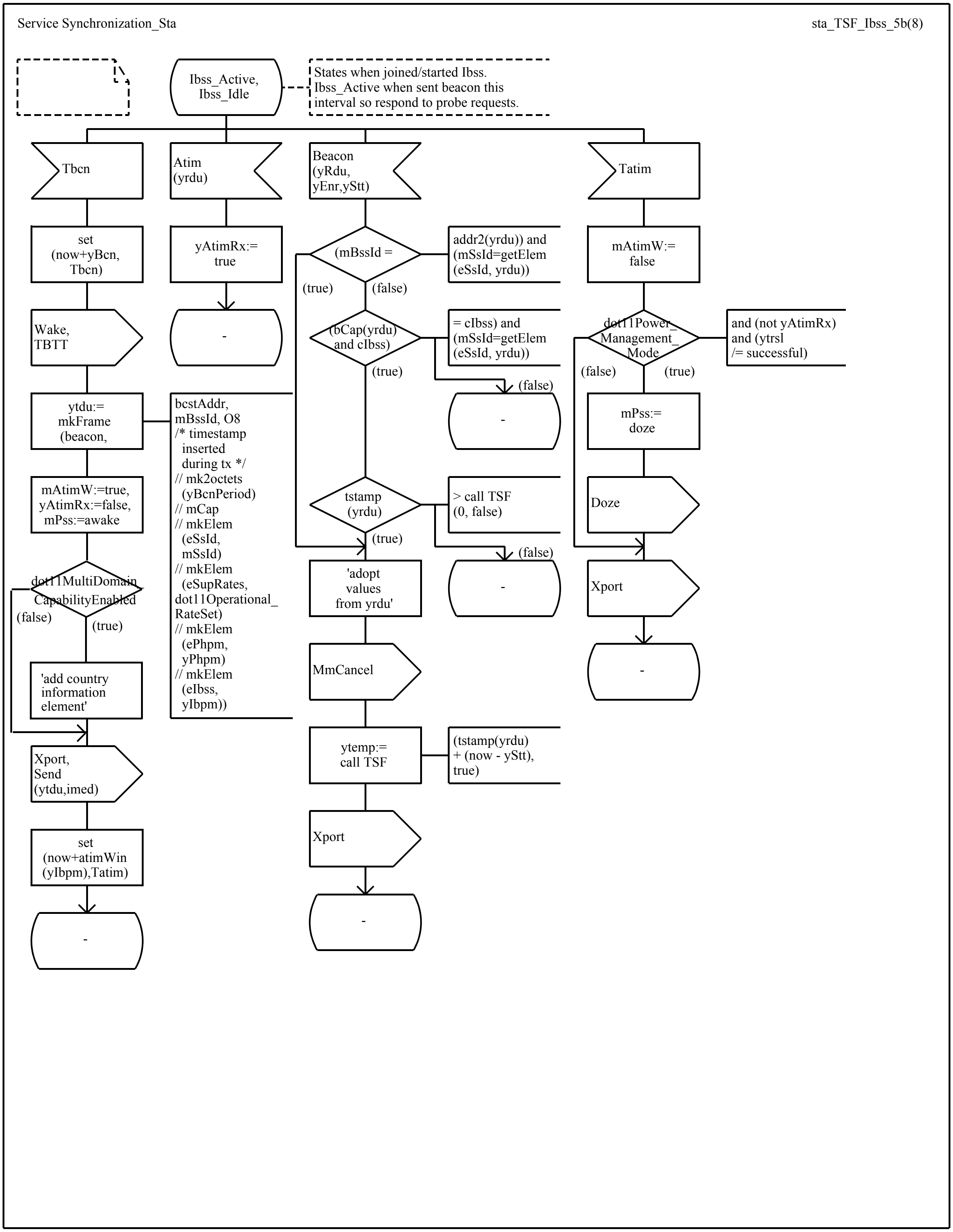

Service Synchronization_Sta
sta_Powermgt_1c(8)
dcl yAtimRx, yPsm, yRdtim, yWake Boolean ;
dcl yAtw, yBcn, yEnr, yMocp, yStt Time ;
dcl yBcnPeriod, yDtim, ycmax, ycmin Kusec ;
dcl ybd BssDscr ;
dcl ybdset BssDscrSet ;
dcl ybtp BssType ;
dcl ybsid MacAddr ;
dcl yclist Intstring ;
dcl ycx, yJto, ytemp Integer ;
dcl yDspm DsParms ;
dcl yFhpm FhParms ;
dcl yIbpm IbssParms ;
dcl ypdly Usec ;
dcl yPhpm PhyParms ;
dcl yRdu, yTdu Frame ;
dcl yssid Octetstring ;
dcl yOrates Ratestring;
dcl ystp ScanType ;
dcl ytrsl TxResult ;
timer Tscan,
Tmocp, Tpdly ;
*
ResetMAC
'obtain Phy
characteristics'
variables
to default
values'
'reset all
intra-MAC
remote
Set TSF
time to
zero.
ytemp:=
call TSF
(0, true)
Xport
Setting these
timers to now
causes events
in each of the
multi-state
services of the
process, forcing
each service to
its idle state.
reset(Tbcn),
reset(Tatim),
set(now,Tasoc),
set(now,Tauth),
set(not,Tchal)
No_BSS
PsmDone
PsmDone sent
by TxCoord
after change
in power save
indication is
announced in
frame exchange.
not
mDisable
MlmePower_
mgt.confirm
(success)
-
No_Bss, Bss,
Ibss_Active,
Ibss_Idle
PowerMgt requests
valid in all
top-level states.
Mlme_
PowerMgt._
request
(yPsm,
yWake,
yRdtim)
'mReceive_
Dtims:=
yRdtim'
(yWake and
(mPss = Doze))
or ((yPsm =
station_active)
and (dot11PowerMan_
agementMode =
power_save))
(false)
(true)
mPss:=
awake
Wake
dot11Power_
ManagementMode:=
yPsm
Xport
-

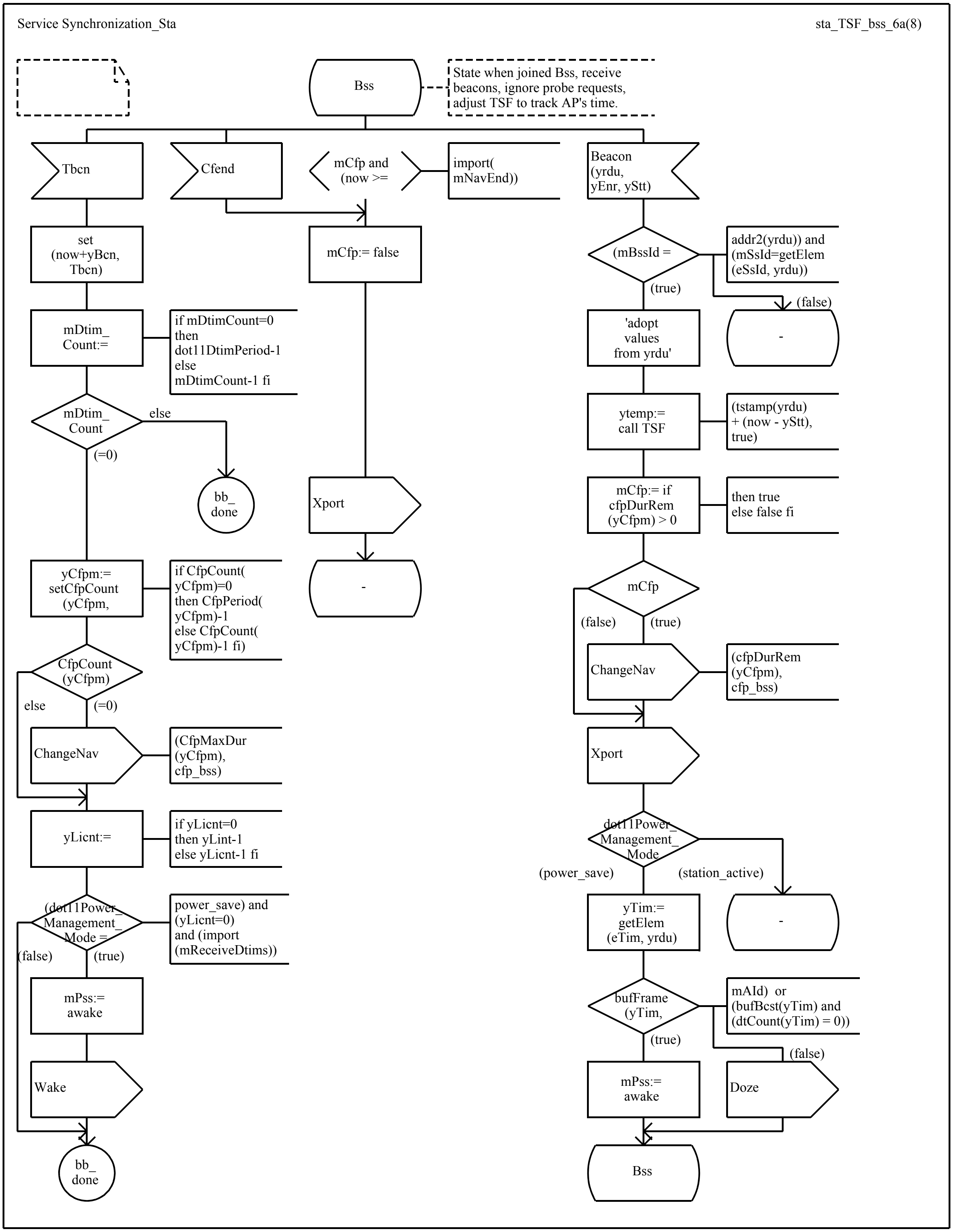

Service Synchronization_Sta
sta_Scan_2e(8)
No_Bss, Bss, Ibss_Active, Ibss_Idle
Scan requests valid in all top-level states.
MlmeScan._ request(ybtp,ybsid,
yssid, ystp, ypdly, yclist, ycmin, ycmax)
Only accept Scan request when no frame exchange is in progress.
not import (mFxIP)
'parameters valid'
(false)
(true)
MlmeScan._ confirm(
empty,invalid)
-
No loss sync if scan parms are invalid.
ybdset:=Empty, ycx:= 0, mDisable:=true
dot11Desired_ Ssid:=yssid,
dot11Desired_ BssType:=ybtp
export(dot11Desired_ Ssid,
dot11Desired_ BssType)
Xport, Wake
(active_scan)
ystype
(passive_scan)
(false)
dot11MultiDomain_ CapabilityEnabled 'and country information valid'
(true)
nx_ chnl
tpdu:= mkFrame (probe_req,
bcstAddr, ybsid, mkElem(eSsId, ySsid)//mkElem(eSupRates, dot11Operat_ ionalRateSet))
nx_ chnl

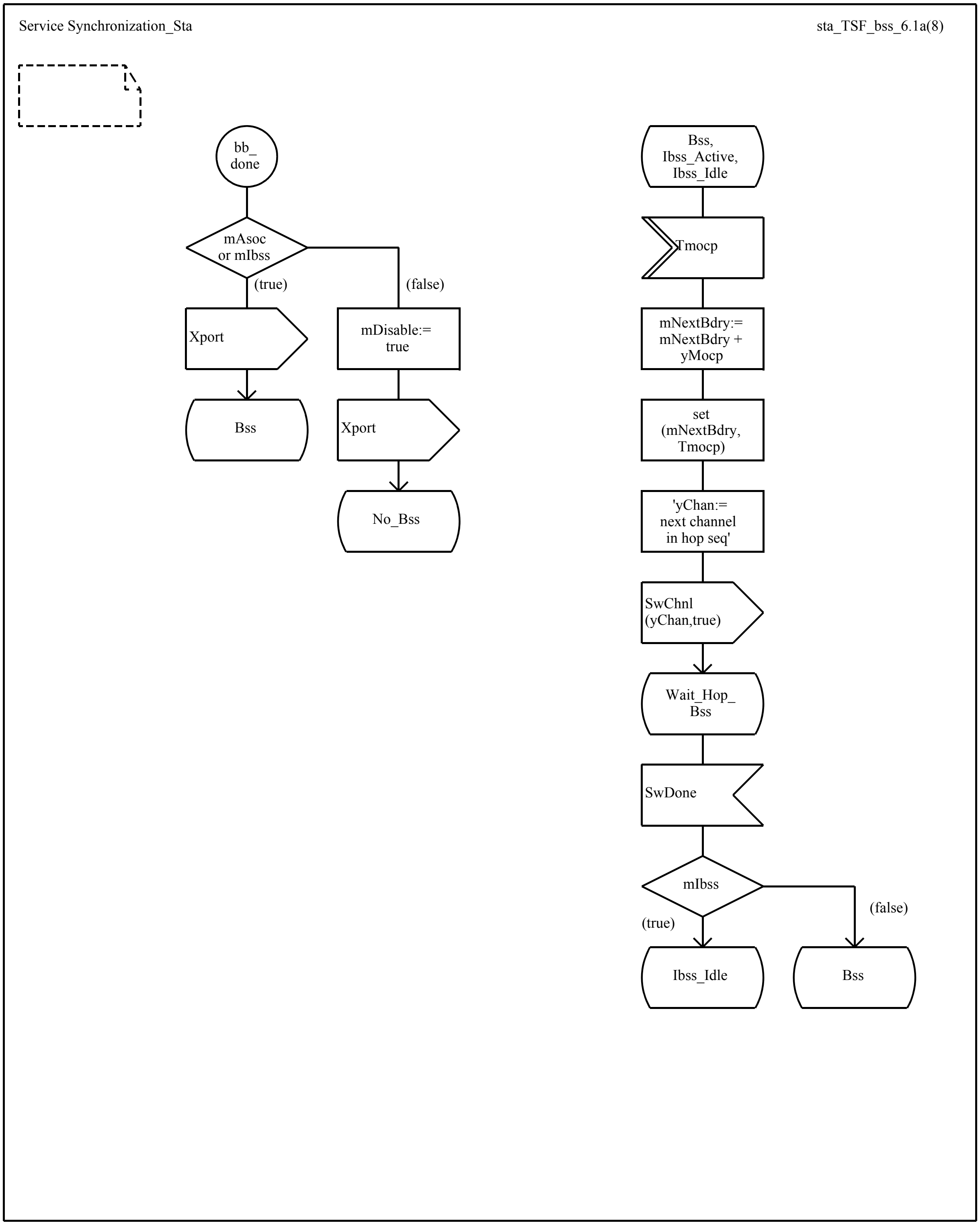

Service Synchronization_Sta
sta_Scan_2.1b(8)
Act_Listen, Act_Receive, Pas_Listen
Beacon (yrdu,yrend, ytstr)
ProbeRsp (yrdu,yrend, ytstr)
'ybd:= bss description info from yrdu'
ybd!bd_ StartTs:= ytstr
ybdset:= ybdset or ybd
-
nx_ chnl
Act_Receive, Pas_Listen
Tscan
ycx:= ycx + 1
ycx > length(yclist)
(false)
(true)
SwChnl (yclist(ycx), true)
'filter ybdset for ybtype and duplicates'
ystype
(active_scan)
(passive_scan)
MlmeScan._ confirm(
ybdset,success)
Wait_Csw_ _Done
Set (now+ycmax, Tscan)
No_Bss
Scan ends in No_Bss state since sync lost with prior Bss. Implementations may save/restore TSF and asoc info to re-join prior Bss.
SwDone
Pas_Listen
set (now+ypdly, Tscan)
Set probe delay timeout.
Listen for activity on channel.
Act_Listen
Wait_Probe_ _Delay
Tscan
import (mRxA)
Tscan
*
nx_ chnl
Set (now+ycmax, Tscan)
Set probe response (max) timeout.
Send (tpdu,imed)
Transmit probe request.
Go to next channel if no activity by min time.
Act_Receive
Set (now+ycmim, Tscan)
Set channel activity (min) timeout.
Receive responses on channel.
Act_Listen

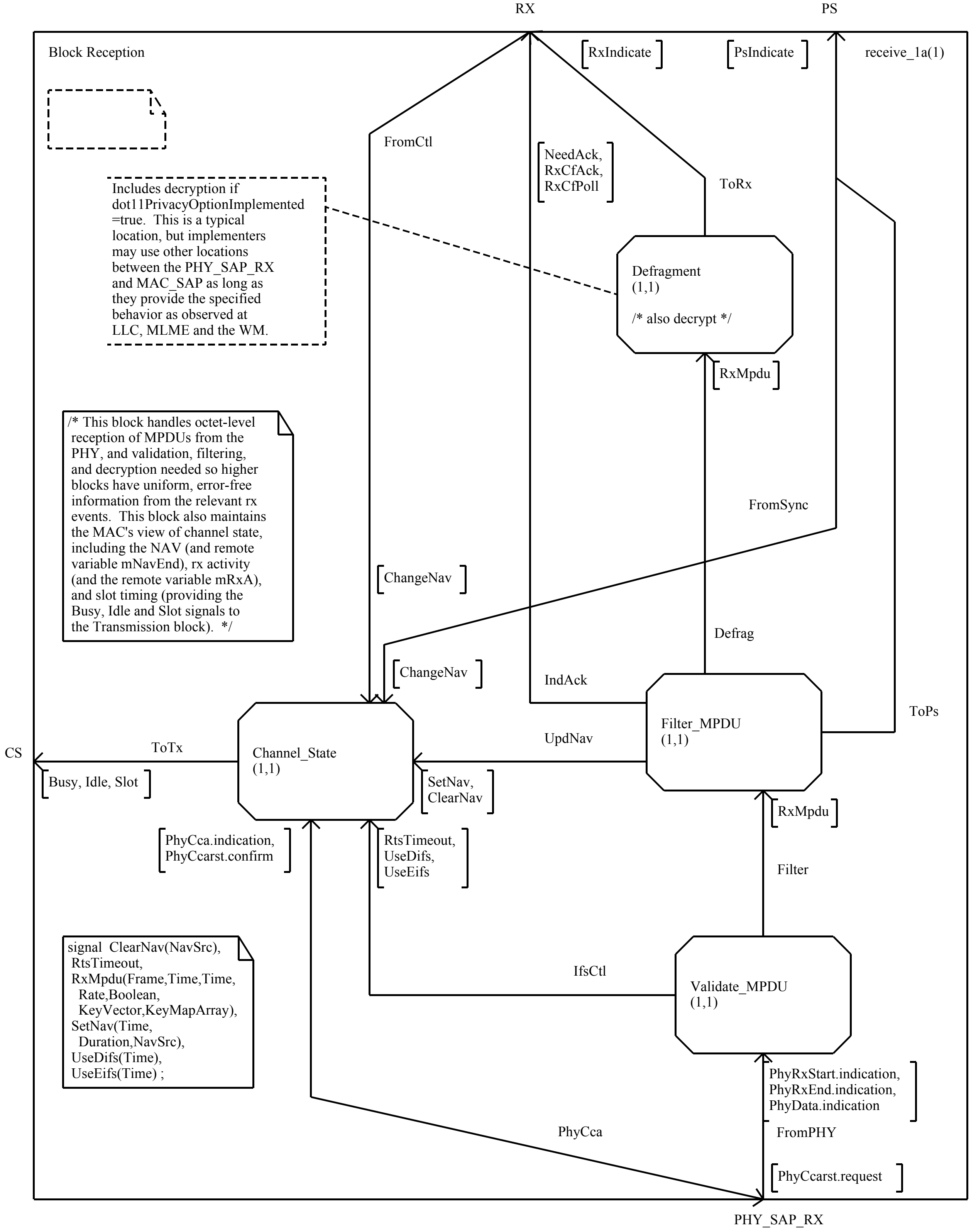

Service Synchronization_Sta
sta_Start_Ibss_3d(8)
No_BSS
Start IBSS on this page, join on next page.
MlmeStart._ request (mSsid, yBtp,
yBcnPeriod, yDtim, /* cfpm */, yPhpm, yIbpm, ypdly, mCap, mBrates, yOrates)
yBtp
(independent)
(infra_structure)
dot11MultiDomain CapabilityImplemented
(false)
(true)
sCanBeAp
(true)
(false)
(false)
dot11MultiDomain_ CapabilityEnabled 'and country information valid'
(true)
'parameters valid'
(false)
(true)
MlmeStart._ confirm (invalid)
No_Bss
mIbss:=true, mPss:=awake, mPdly:=ypdly
mBssId:= 0 // 1 // '46 random bits'
46-bit string needs to be very random, see 11.1.3.
dot11Beacon_ Period:= yBcnPeriod,
dot11DtimPeriod:= yDtim,dot11Oper_ ationalRateSet:=yOrates
export(dot11_ BeaconPeriod, dot11DtimPeriod,
dot11Operational_ RateSet)
AP_Active
Activate AP state machine.
yBcn:= kUsec (yBcnPeriod)
yAtw:=kUsec (atimWin (yIbpm))
'set actual phy parameters from phpm'
Xport
ibss_ init
Wait probe delay before initiating a transmission.
Init_Wait_ _ProbeDelay
Tpdly
*
Ibss_Active
Start out as Ibss probe responder.
ibss_ init
'dot11_ PHYType=
FHphy'
(false)
(true)
yMocp:=kUsec (dwellTime (yFhpm))
mNextBdry:= now+(yMocp - (call TSF
(0,false) mod yMocp))
set (mNextBdry, Tmocp)
Initialize dwell timer.
'yChan:= first (or only) channel'
Set starting channel (FH) or operating channel (DS), null for IR.
SwChnl (yChan,true)
mNextTbtt:= now+(yBcn - (call TSF
(0,false) mod yBcn))
set (mNextTbtt, Tbcn)
Initialize beacon timer.
mIbss:= true, mDisable:= false
Xport
MlmeStart._ confirm (success)
set(now + tUsec(ypdly), Tpdly)


Service Synchronization_Sta
sta_Join_4d(8)
No_Bss
Join on this page, Start on previous page
MlmeJoin._ request (ybd,yJto,
ypdly, yOrates)
'bss dscr valid'
(false)
(true)
MlmeJoin._ confirm (invalid)
No_Bss
mBssId:=ybd!_ bdBssId,mSsId:= ybd!bdSsId,
dot11Beacon_ Period:=ybd!_ bdBcnPer
yBcn:=kUsec(ybd!bdBcnPer), dot11DtimPeriod
:= ybd!bdDtimPer, yphpm:=ybd!_ bdPhyParms
ycfpm:=ybd!bd_ CfParms, yibpm:= ybd!bdIbParms,
mCap:= ybd!_ bdCap,mBrates:= ybd!bdRates, mPdly:=ypdly
dot11Oper_ ationalRateSet:= yOrates
export(dot11_ BeaconPeriod, dot11DtimPeriod,
dot11Oper_ rationalRateSet)
set (now + yBcn, Tbcn)
Xport
'select channel of target Bss'
ytemp:= call TSF((now -
ybd!_ bdStartTs), true)
Join_Wait_ _Beacon
Join_Wait_Bcn
Tbcn
yJto:= yJto - 1
yJto
(=0)
(>0)
set (now + yBcn, Tbcn)
-
MlmeJoin._ confirm (timeout)
mBssId:= nullAddr, mSsId:=null
No_Bss
mCap and cIbss
else
(=cIbss)
mDisable:= false, mIbss:=false
mDisable:= false, mIbss:= true
Xport
Xport
Bss
Ibss_Idle
Beacon (yrdu, yenr, ystt)
A probe response from the bss/ibss can also be used to establish sync.
(mBssId =
addr2(yrdu)) and (mSsId = getElem (eSsId,yrdu))
(true)
(false)
-
dot11MultiDomain CapabilityEnabled
(true)
(false)
'adopt values from yrdu'
'country information present in beacon'
(true)
(false)
-
'dot11_ PHYType=
FHphy'
(false)
(true)
yMocp:=kUsec (dwellTime (yFhpm))
mNextBdry:= now+(yMocp - (call TSF
(0,false) mod yMocp))
set (mNextBdry, Tmocp)
Initialize dwell timer.
'yChan:= first (or only) channel'
Set starting channel (FH) or operating channel (DS), null for IR.
SwChnl (yChan,true)
mNextTbtt:= now+(yBcn - (call TSF
(0,false) mod yBcn))
set (mNextTbtt, Tbcn)
Initialize beacon timer.
MlmeJoin._ confirm (success)


Service Synchronization_Sta
sta_TSF_Ibss_5b(8)
Ibss_Active,
Ibss_Idle
States when joined/started Ibss.
Ibss_Active when sent beacon this
interval so respond to probe requests.
Tbcn
set
(now+yBcn,
Tbcn)
Wake,
TBTT
ytdu:=
mkFrame
(beacon,
bcstAddr,
mBssId, O8
/* timestamp
inserted
during tx */
// mk2octets
(yBcnPeriod)
// mCap
// mkElem
(eSsId,
mSsId)
// mkElem
(eSupRates,
dot11Operational_
RateSet)
// mkElem
(ePhpm,
yPhpm)
// mkElem
(eIbss,
yIbpm))
mAtimW:=true,
yAtimRx:=false,
mPss:=awake
dot11MultiDomain
CapabilityEnabled
(false)
(true)
'add country
information
element'
Xport,
Send
(ytdu,imed)
set
(now+atimWin
(yIbpm),Tatim)
-
Atim
(yrdu)
yAtimRx:=
true
-
Beacon
(yRdu,
yEnr,yStt)
(mBssId =
addr2(yrdu)) and
(mSsId=getElem
(eSsId, yrdu))
(true)
(false)
(bCap(yrdu)
and cIbss)
= cIbss) and
(mSsId=getElem
(eSsId, yrdu))
(true)
(false)
-
tstamp
(yrdu)
> call TSF
(0, false)
(true)
(false)
-
'adopt
values
from yrdu'
MmCancel
ytemp:=
call TSF
(tstamp(yrdu)
+ (now - yStt),
true)
Xport
-
Tatim
mAtimW:=
false
dot11Power_
Management_
Mode
and (not yAtimRx)
and (ytrsl
/= successful)
(false)
(true)
mPss:=
doze
Doze
Xport
-


Service Synchronization_Sta
sta_TSF_bss_6a(8)
Bss
State when joined Bss, receive beacons, ignore probe requests, adjust TSF to track AP's time.
Tbcn
Cfend
mCfp and (now >=
import(mNavEnd))
Beacon (yrdu, yEnr, yStt)
set (now+yBcn, Tbcn)
mCfp:= false
(mBssId =
addr2(yrdu)) and (mSsId=getElem (eSsId, yrdu))
(true)
(false)
mDtim_ Count:=
if mDtimCount=0 then dot11DtimPeriod-1 else mDtimCount-1 fi
'adopt values from yrdu'
-
mDtim_ Count
else
(=0)
ytemp:= call TSF
(tstamp(yrdu) + (now - yStt), true)
bb_ done
Xport
mCfp:= if cfpDurRem (yCfpm) > 0
then true else false fi
yCfpm:= setCfpCount (yCfpm,
if CfpCount(yCfpm)=0 then CfpPeriod(yCfpm)-1 else CfpCount(yCfpm)-1 fi)
-
mCfp
(false)
(true)
CfpCount (yCfpm)
else
(=0)
ChangeNav
(cfpDurRem (yCfpm), cfp_bss)
ChangeNav
(CfpMaxDur (yCfpm), cfp_bss)
Xport
yLicnt:=
if yLicnt=0 then yLint-1 else yLicnt-1 fi
dot11Power_ Management_ Mode
(power_save)
(station_active)
(dot11Power_ Management_ Mode =
power_save) and (yLicnt=0) and (import (mReceiveDtims))
(false)
(true)
yTim:= getElem (eTim, yrdu)
-
mPss:= awake
bufFrame (yTim,
mAId) or (bufBcst(yTim) and (dtCount(yTim) = 0))
(true)
(false)
Wake
mPss:= awake
Doze
bb_ done
Bss

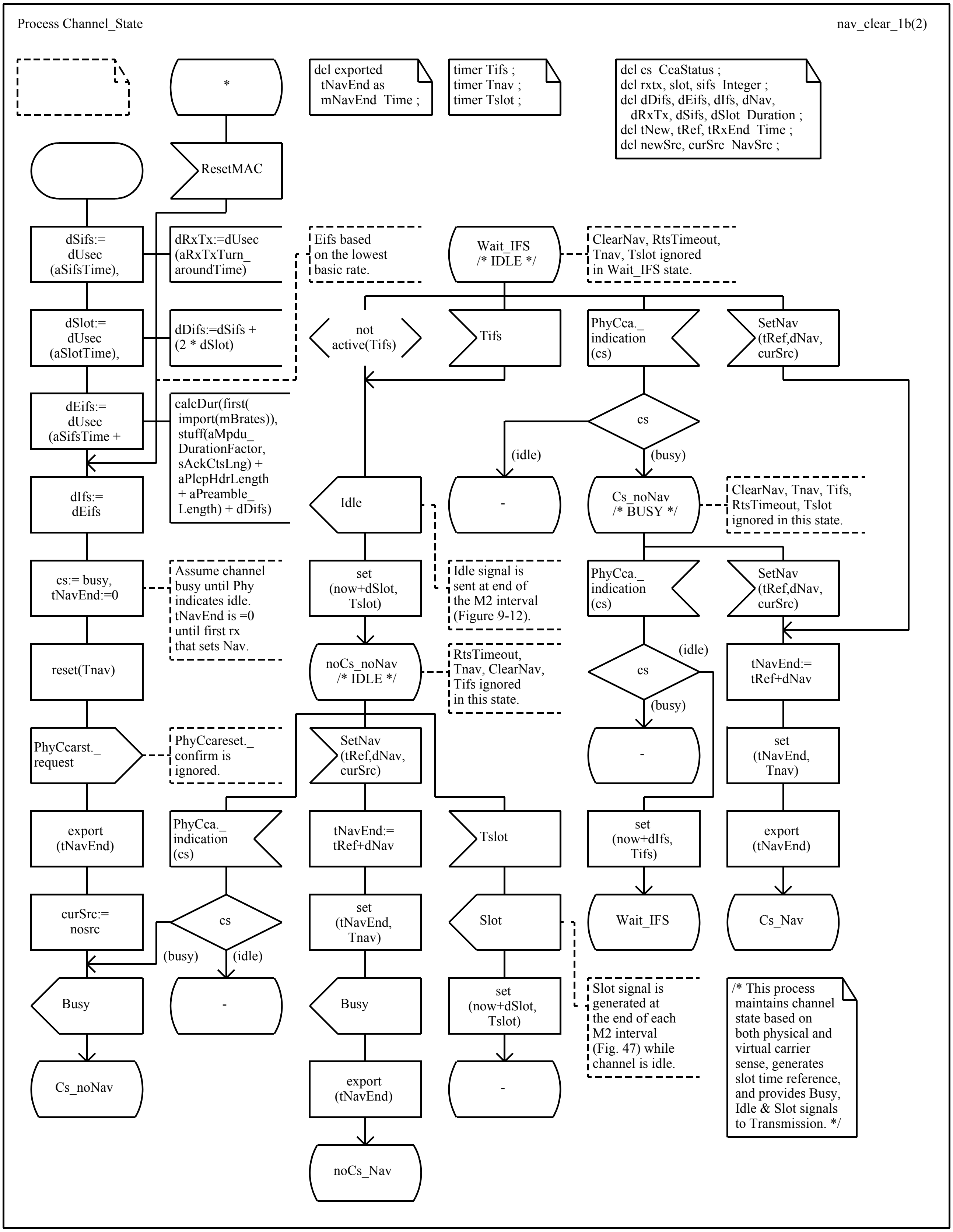

Service Synchronization_Sta
sta_TSF_bss_6.1a(8)
bb_ done
mAsoc or mIbss
(true)
(false)
Xport
mDisable:= true
Bss
Xport
No_Bss
Bss, Ibss_Active, Ibss_Idle
Tmocp
mNextBdry:= mNextBdry + yMocp
set (mNextBdry, Tmocp)
'yChan:= next channel in hop seq'
SwChnl (yChan,true)
Wait_Hop_ Bss
SwDone
mIbss
(true)
(false)
Ibss_Idle
Bss

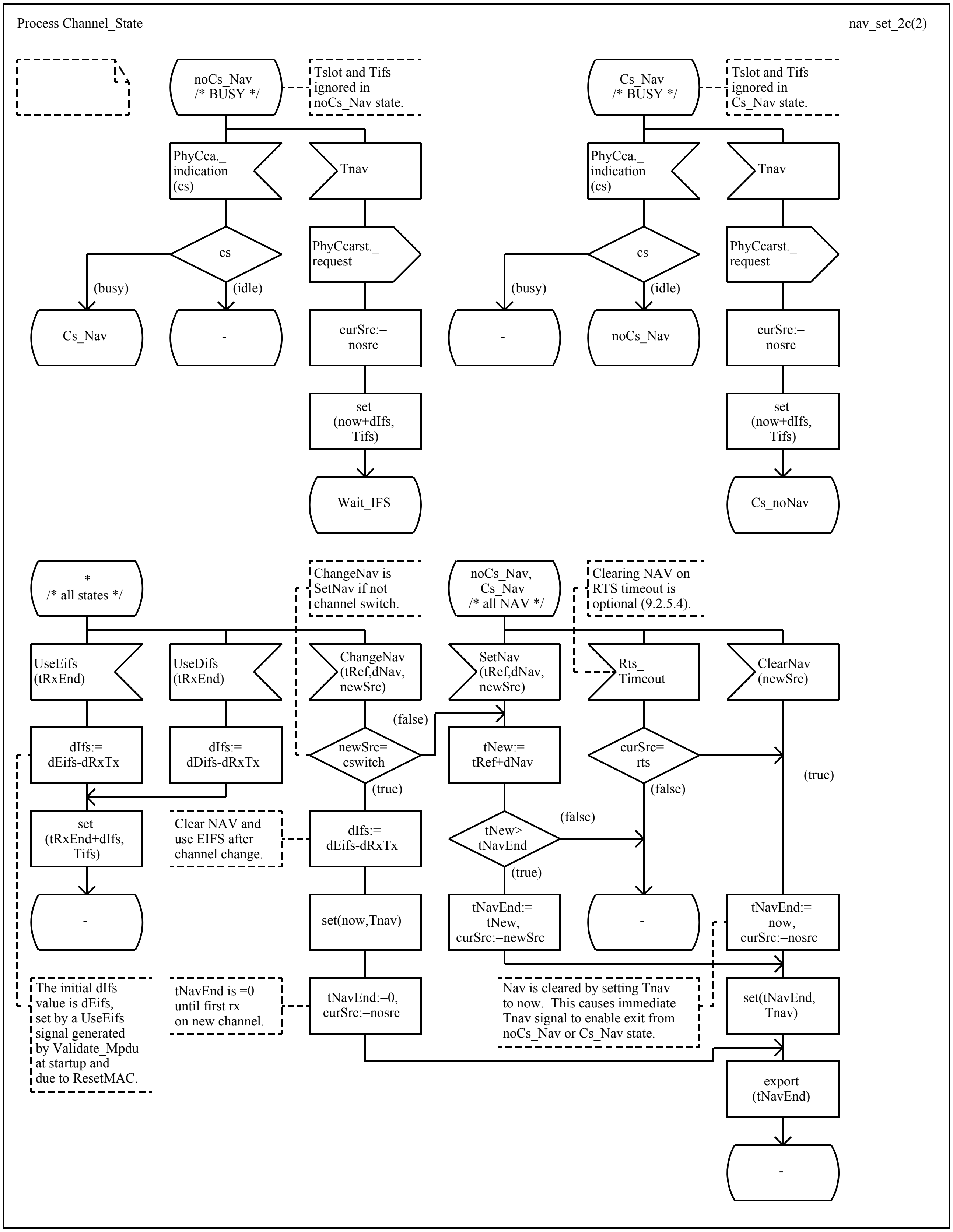


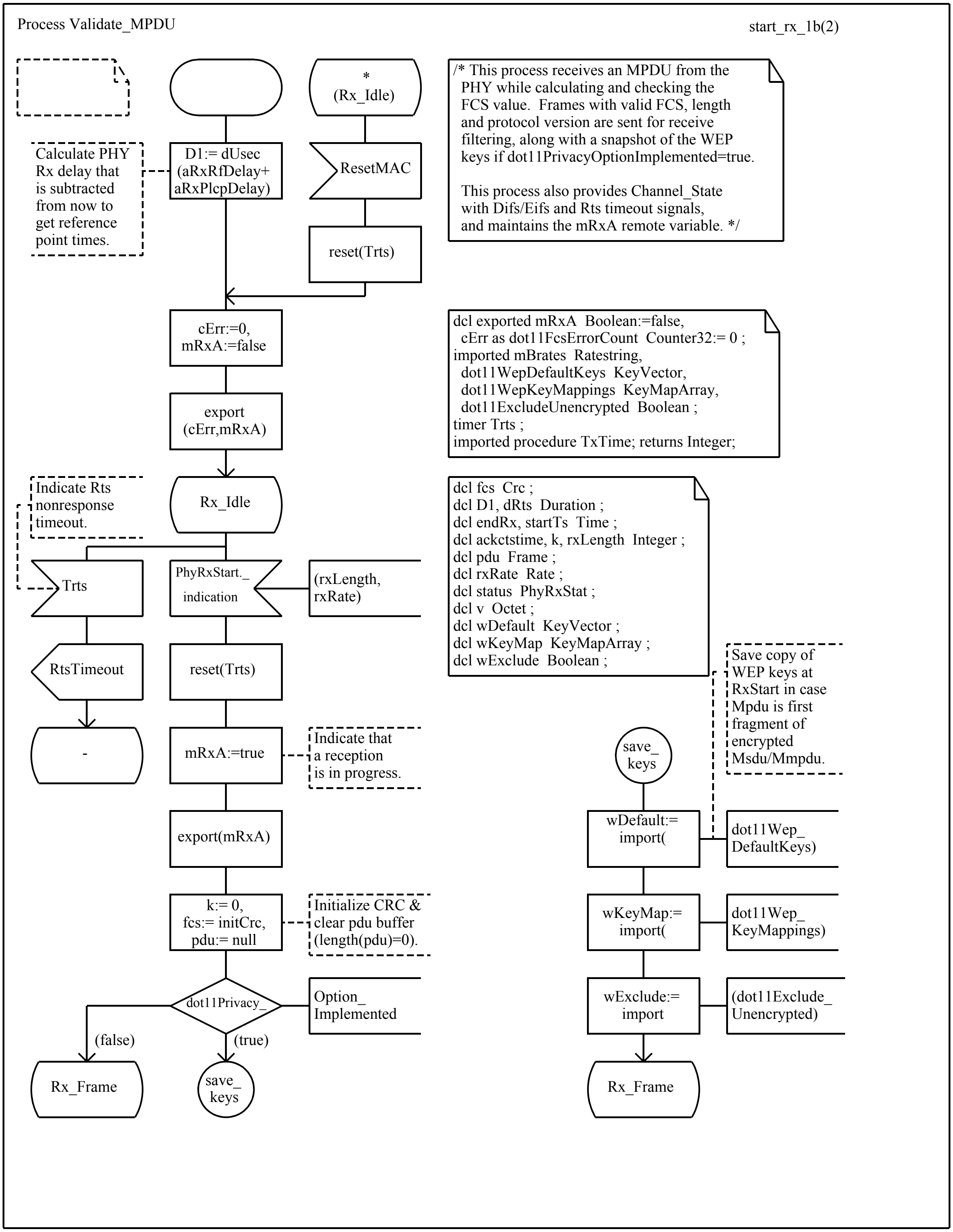
Includes decryption if dot11PrivacyOptionImplemented =true. This is a typical location, but implementers may use other locations between the PHY_SAP_RX and MAC_SAP as long as they provide the specified behavior as observed at LLC, MLME and the WM.

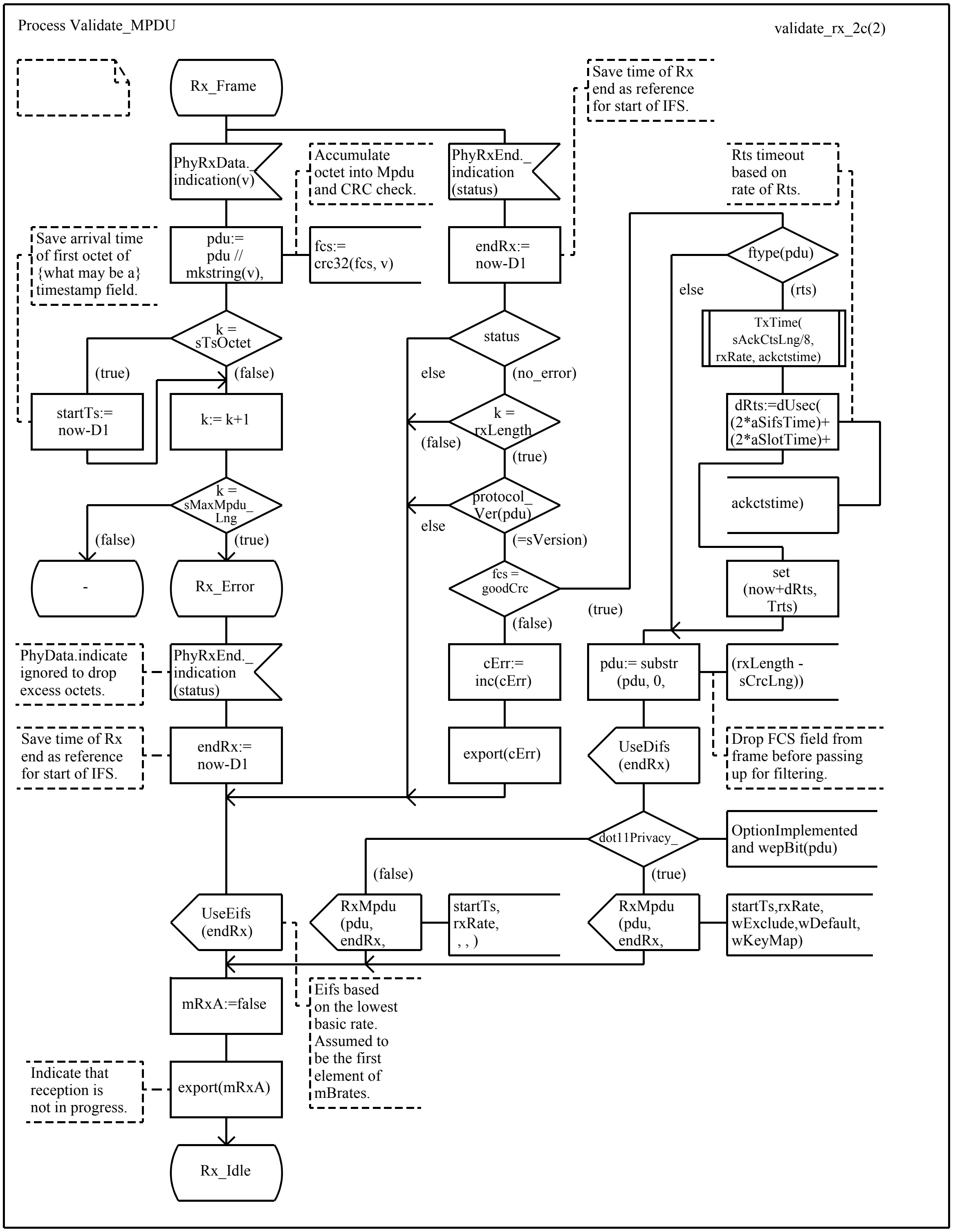
/* This block handles octet-level reception of MPDUs from the PHY, and validation, filtering, and decryption needed so higher blocks have uniform, error-free information from the relevant rx events. This block also maintains the MAC's view of channel state, including the NAV (and remote variable mNavEnd), rx activity (and the remote variable mRxA), and slot timing (providing the Busy, Idle and Slot signals to the Transmission block). */

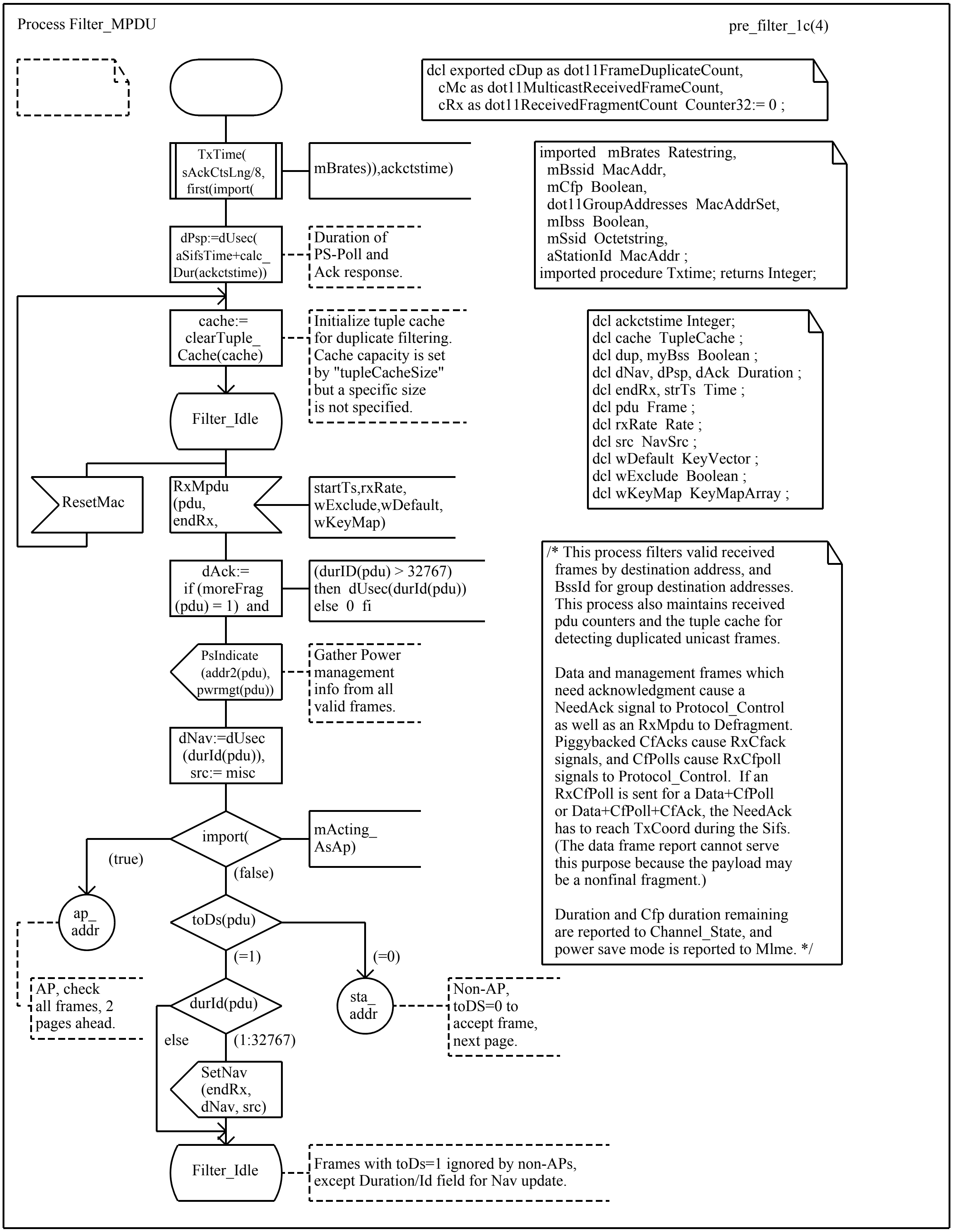
```
signal  ClearNav(NavSrc),
 RtsTimeout,
 RxMpdu(Frame,Time,Time,
   Rate,Boolean,
   KeyVector,KeyMapArray),
 SetNav(Time,
   Duration,NavSrc),
 UseDifs(Time),
 UseEifs(Time) ;
```

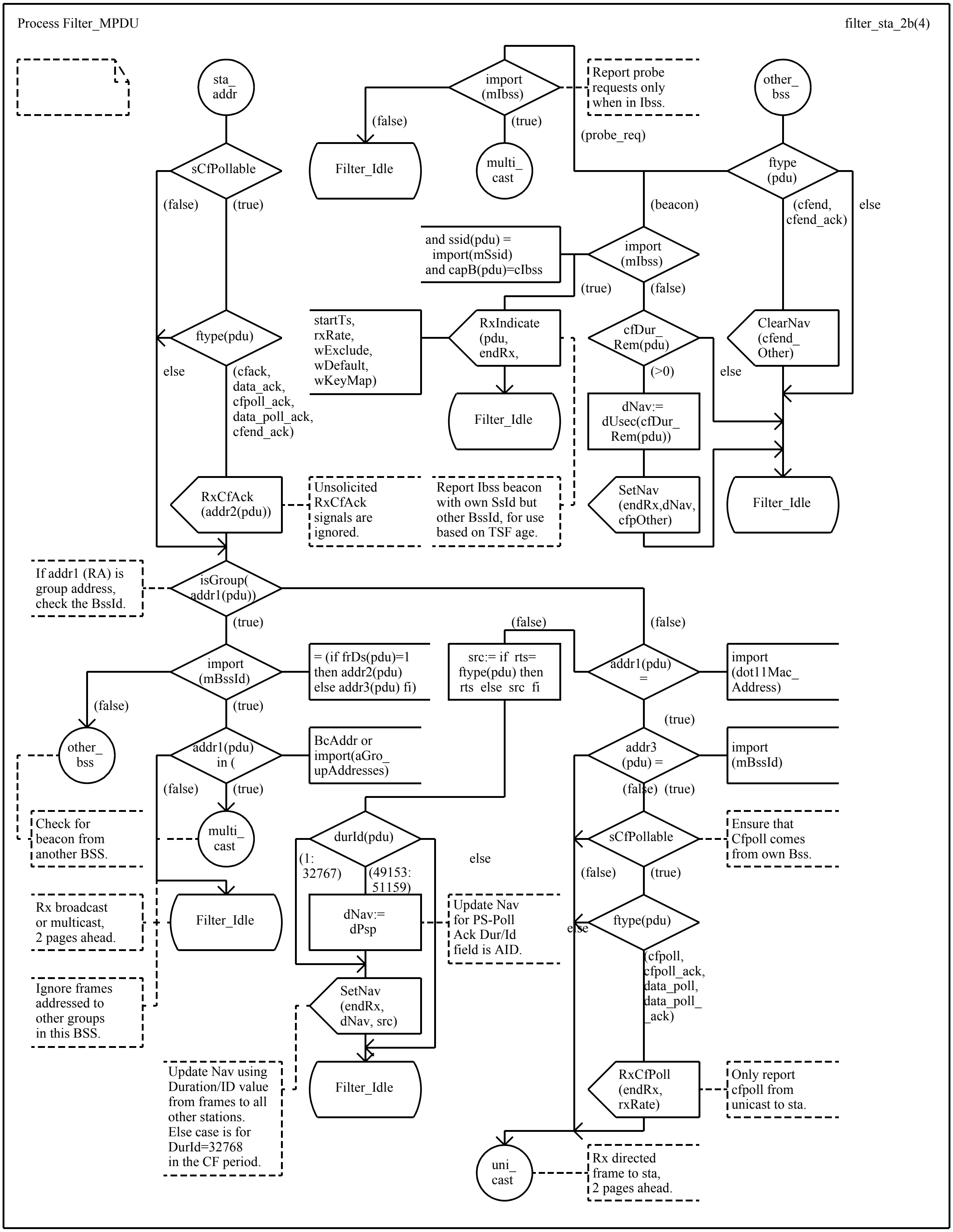

Process Channel_State
nav_clear_1b(2)
*
ResetMAC
dcl exported tNavEnd as mNavEnd Time ;
timer Tifs ; timer Tnav ; timer Tslot ;
dcl cs CcaStatus ; dcl rxtx, slot, sifs Integer ; dcl dDifs, dEifs, dIfs, dNav, dRxTx, dSifs, dSlot Duration ; dcl tNew, tRef, tRxEnd Time ; dcl newSrc, curSrc NavSrc ;
dSifs:= dUsec (aSifsTime),
dRxTx:=dUsec (aRxTxTurn_ aroundTime)
Eifs based on the lowest basic rate.
Wait_IFS /* IDLE */
ClearNav, RtsTimeout, Tnav, Tslot ignored in Wait_IFS state.
dSlot:= dUsec (aSlotTime),
dDifs:=dSifs + (2 * dSlot)
not active(Tifs)
Tifs
PhyCca._ indication (cs)
SetNav (tRef,dNav, curSrc)
dEifs:= dUsec (aSifsTime +
calcDur(first(import(mBrates)), stuff(aMpdu_ DurationFactor, sAckCtsLng) + aPlcpHdrLength + aPreamble_ Length) + dDifs)
cs
(idle)
(busy)
dIfs:= dEifs
Idle
-
Cs_noNav /* BUSY */
ClearNav, Tnav, Tifs, RtsTimeout, Tslot ignored in this state.
cs:= busy, tNavEnd:=0
Assume channel busy until Phy indicates idle. tNavEnd is =0 until first rx that sets Nav.
set (now+dSlot, Tslot)
Idle signal is sent at end of the M2 interval (Figure 9-12).
PhyCca._ indication (cs)
SetNav (tRef,dNav, curSrc)
reset(Tnav)
noCs_noNav /* IDLE */
RtsTimeout, Tnav, ClearNav, Tifs ignored in this state.
cs
(idle)
(busy)
tNavEnd:= tRef+dNav
PhyCcarst._ request
PhyCcareset._ confirm is ignored.
SetNav (tRef,dNav, curSrc)
-
set (tNavEnd, Tnav)
export (tNavEnd)
PhyCca._ indication (cs)
tNavEnd:= tRef+dNav
Tslot
set (now+dIfs, Tifs)
export (tNavEnd)
curSrc:= nosrc
cs
set (tNavEnd, Tnav)
Slot
Wait_IFS
Cs_Nav
(busy)
(idle)
Busy
-
Busy
set (now+dSlot, Tslot)
Slot signal is generated at the end of each M2 interval (Fig. 47) while channel is idle.
/* This process maintains channel state based on both physical and virtual carrier sense, generates slot time reference, and provides Busy, Idle & Slot signals to Transmission. */
Cs_noNav
export (tNavEnd)
-
noCs_Nav

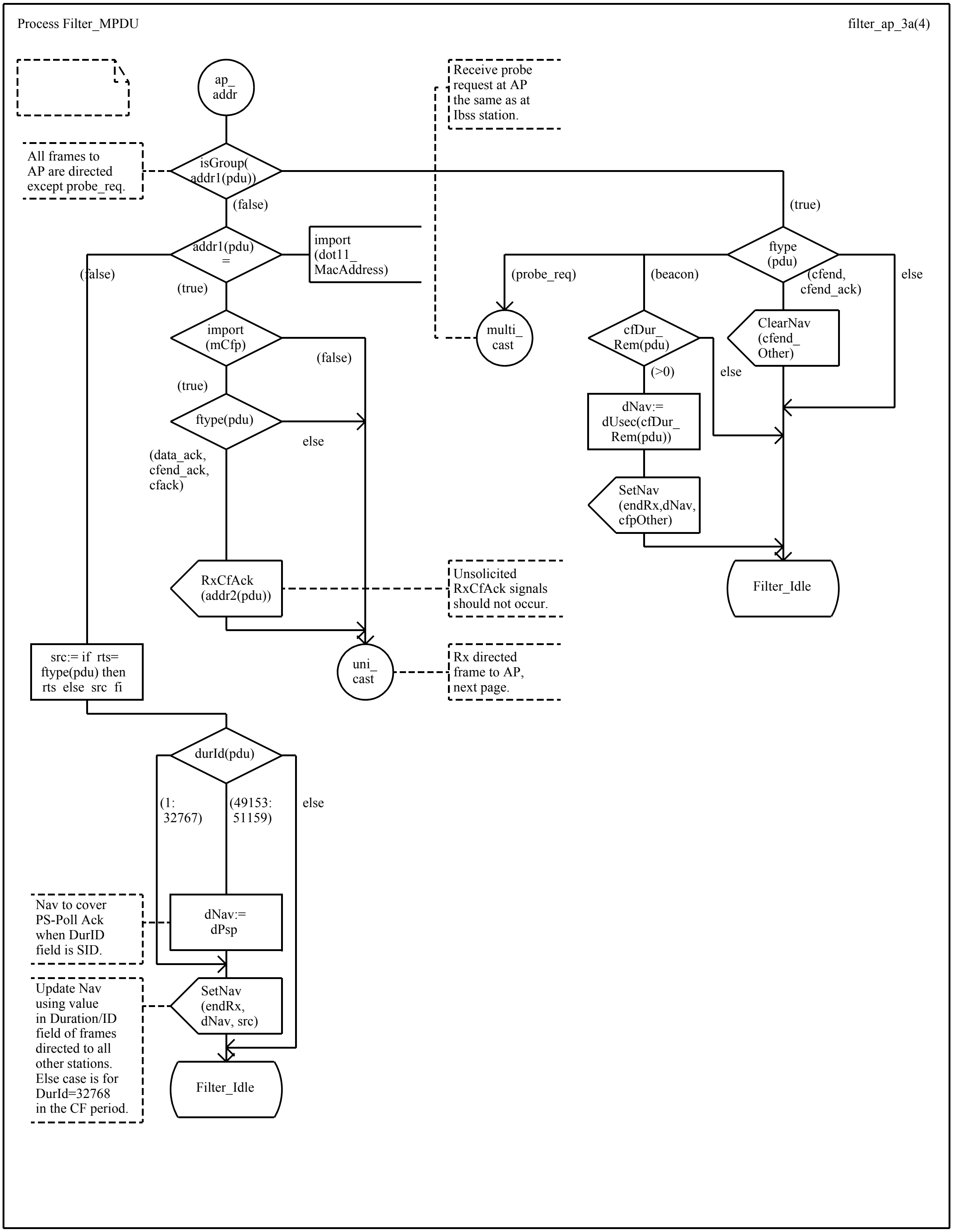

Process Channel_State
nav_set_2c(2)
noCs_Nav
/* BUSY */
Tslot and Tifs
ignored in
noCs_Nav state.
PhyCca._
indication
(cs)
Tnav
cs
(busy)
(idle)
Cs_Nav
-
PhyCcarst._
request
curSrc:=
nosrc
set
(now+dIfs,
Tifs)
Wait_IFS
Cs_Nav
/* BUSY */
Tslot and Tifs
ignored in
Cs_Nav state.
PhyCca._
indication
(cs)
Tnav
cs
(busy)
(idle)
-
noCs_Nav
PhyCcarst._
request
curSrc:=
nosrc
set
(now+dIfs,
Tifs)
Cs_noNav
*
/* all states */
UseEifs
(tRxEnd)
UseDifs
(tRxEnd)
ChangeNav is
SetNav if not
channel switch.
ChangeNav
(tRef,dNav,
newSrc)
noCs_Nav,
Cs_Nav
/* all NAV */
Clearing NAV on
RTS timeout is
optional (9.2.5.4).
SetNav
(tRef,dNav,
newSrc)
Rts_
Timeout
ClearNav
(newSrc)
dIfs:=
dEifs-dRxTx
dIfs:=
dDifs-dRxTx
(false)
newSrc=
cswitch
(true)
tNew:=
tRef+dNav
curSrc=
rts
(false)
(true)
set
(tRxEnd+dIfs,
Tifs)
Clear NAV and
use EIFS after
channel change.
dIfs:=
dEifs-dRxTx
tNew>
tNavEnd
(false)
(true)
-
set(now,Tnav)
tNavEnd:=
tNew,
curSrc:=newSrc
-
tNavEnd:=
now,
curSrc:=nosrc
The initial dIfs
value is dEifs,
set by a UseEifs
signal generated
by Validate_Mpdu
at startup and
due to ResetMAC.
tNavEnd is =0
until first rx
on new channel.
tNavEnd:=0,
curSrc:=nosrc
Nav is cleared by setting Tnav
to now. This causes immediate
Tnav signal to enable exit from
noCs_Nav or Cs_Nav state.
set(tNavEnd,
Tnav)
export
(tNavEnd)
-

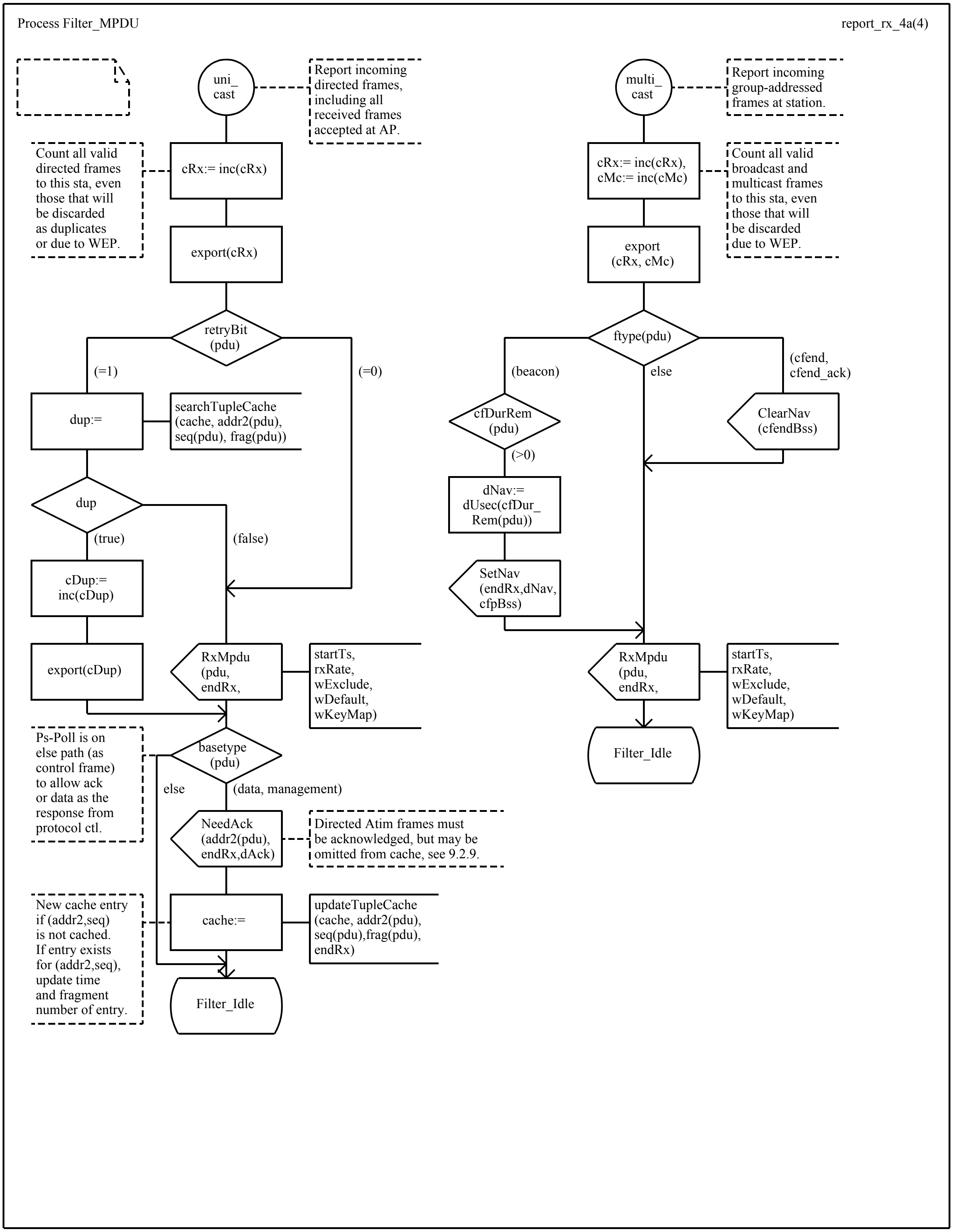

Process Validate_MPDU
start_rx_1b(2)
/* This process receives an MPDU from the PHY while calculating and checking the FCS value. Frames with valid FCS, length and protocol version are sent for receive filtering, along with a snapshot of the WEP keys if dot11PrivacyOptionImplemented=true.
This process also provides Channel_State with Difs/Eifs and Rts timeout signals, and maintains the mRxA remote variable. */
*
(Rx_Idle)
ResetMAC
reset(Trts)
Calculate PHY Rx delay that is subtracted from now to get reference point times.
D1:= dUsec (aRxRfDelay+ aRxPlcpDelay)
cErr:=0, mRxA:=false
export (cErr,mRxA)
dcl exported mRxA Boolean:=false,
cErr as dot11FcsErrorCount Counter32:= 0 ;
imported mBrates Ratestring,
dot11WepDefaultKeys KeyVector,
dot11WepKeyMappings KeyMapArray,
dot11ExcludeUnencrypted Boolean ;
timer Trts ;
imported procedure TxTime; returns Integer;
Indicate Rts nonresponse timeout.
Rx_Idle
dcl fcs Crc ;
dcl D1, dRts Duration ;
dcl endRx, startTs Time ;
dcl ackctstime, k, rxLength Integer ;
dcl pdu Frame ;
dcl rxRate Rate ;
dcl status PhyRxStat ;
dcl v Octet ;
dcl wDefault KeyVector ;
dcl wKeyMap KeyMapArray ;
dcl wExclude Boolean ;
Trts
PhyRxStart._ indication
(rxLength, rxRate)
RtsTimeout
reset(Trts)
-
mRxA:=true
Indicate that a reception is in progress.
export(mRxA)
k:= 0, fcs:= initCrc, pdu:= null
Initialize CRC & clear pdu buffer (length(pdu)=0).
dot11Privacy_
Option_ Implemented
(false)
(true)
Rx_Frame
save_ keys
Save copy of WEP keys at RxStart in case Mpdu is first fragment of encrypted Msdu/Mmpdu.
save_ keys
wDefault:= import(
dot11Wep_ DefaultKeys)
wKeyMap:= import(
dot11Wep_ KeyMappings)
wExclude:= import
(dot11Exclude_ Unencrypted)
Rx_Frame

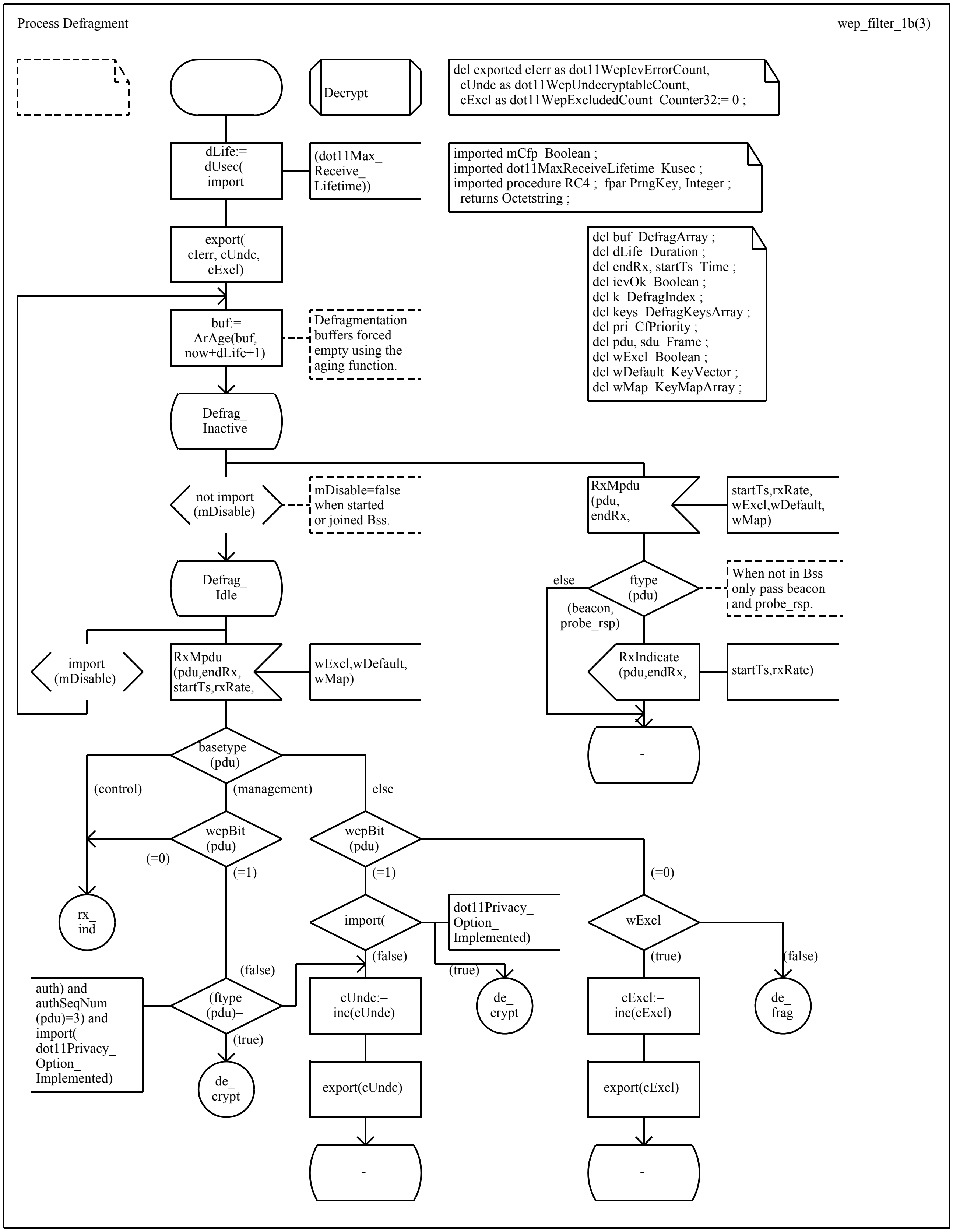

Process Validate_MPDU
validate_rx_2c(2)
Rx_Frame
Save time of Rx
end as reference
for start of IFS.
PhyRxData._
indication(v)
Accumulate
octet into Mpdu
and CRC check.
PhyRxEnd._
indication
(status)
Rts timeout
based on
rate of Rts.
Save arrival time
of first octet of
{what may be a}
timestamp field.
pdu:=
pdu //
mkstring(v),
fcs:=
crc32(fcs, v)
endRx:=
now-D1
ftype(pdu)
else
(rts)
k =
sTsOctet
status
TxTime(
sAckCtsLng/8,
rxRate, ackctstime)
(true)
(false)
else
(no_error)
startTs:=
now-D1
k:= k+1
k =
rxLength
dRts:=dUsec(
(2*aSifsTime)+
(2*aSlotTime)+
(false)
(true)
k =
sMaxMpdu_
Lng
protocol_
Ver(pdu)
ackctstime)
else
(false)
(true)
(=sVersion)
-
Rx_Error
fcs =
goodCrc
set
(now+dRts,
Trts)
(true)
(false)
PhyData.indicate
ignored to drop
excess octets.
PhyRxEnd._
indication
(status)
cErr:=
inc(cErr)
pdu:= substr
(pdu, 0,
(rxLength -
sCrcLng))
Save time of Rx
end as reference
for start of IFS.
endRx:=
now-D1
export(cErr)
UseDifs
(endRx)
Drop FCS field from
frame before passing
up for filtering.
dot11Privacy_
OptionImplemented
and wepBit(pdu)
(false)
(true)
UseEifs
(endRx)
RxMpdu
(pdu,
endRx,
startTs,
rxRate,
, ,)
RxMpdu
(pdu,
endRx,
startTs,rxRate,
wExclude,wDefault,
wKeyMap)
mRxA:=false
Eifs based
on the lowest
basic rate.
Assumed to
be the first
element of
mBrates.
Indicate that
reception is
not in progress.
export(mRxA)
Rx_Idle

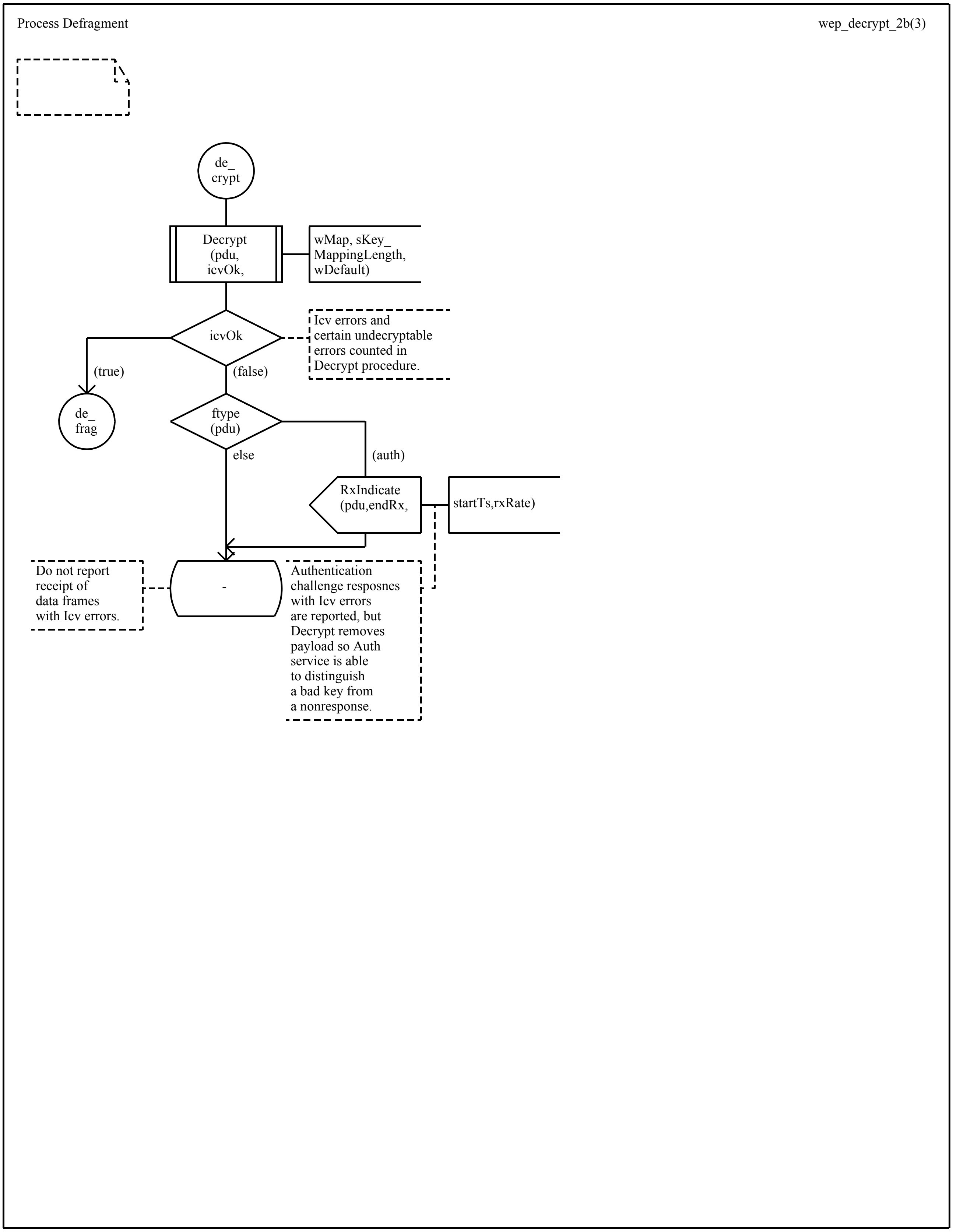

Process Filter_MPDU
pre_filter_1c(4)
dcl exported cDup as dot11FrameDuplicateCount,
cMc as dot11MulticastReceivedFrameCount,
cRx as dot11ReceivedFragmentCount Counter32:= 0 ;
TxTime(
sAckCtsLng/8,
first(import(
mBrates)),ackctstime)
imported mBrates Ratestring,
mBssid MacAddr,
mCfp Boolean,
dot11GroupAddresses MacAddrSet,
mIbss Boolean,
mSsid Octetstring,
aStationId MacAddr ;
imported procedure Txtime; returns Integer;
dPsp:=dUsec(
aSifsTime+calc_
Dur(ackctstime))
Duration of
PS-Poll and
Ack response.
cache:=
clearTuple_
Cache(cache)
Initialize tuple cache
for duplicate filtering.
Cache capacity is set
by "tupleCacheSize"
but a specific size
is not specified.
dcl ackctstime Integer;
dcl cache TupleCache ;
dcl dup, myBss Boolean ;
dcl dNav, dPsp, dAck Duration ;
dcl endRx, strTs Time ;
dcl pdu Frame ;
dcl rxRate Rate ;
dcl src NavSrc ;
dcl wDefault KeyVector ;
dcl wExclude Boolean ;
dcl wKeyMap KeyMapArray ;
Filter_Idle
ResetMac
RxMpdu
(pdu,
endRx,
startTs,rxRate,
wExclude,wDefault,
wKeyMap)
/* This process filters valid received
frames by destination address, and
BssId for group destination addresses.
This process also maintains received
pdu counters and the tuple cache for
detecting duplicated unicast frames.
Data and management frames which
need acknowledgment cause a
NeedAck signal to Protocol_Control
as well as an RxMpdu to Defragment.
Piggybacked CfAcks cause RxCfack
signals, and CfPolls cause RxCfpoll
signals to Protocol_Control. If an
RxCfPoll is sent for a Data+CfPoll
or Data+CfPoll+CfAck, the NeedAck
has to reach TxCoord during the Sifs.
(The data frame report cannot serve
this purpose because the payload may
be a nonfinal fragment.)
Duration and Cfp duration remaining
are reported to Channel_State, and
power save mode is reported to Mlme. */
dAck:=
if (moreFrag
(pdu) = 1) and
(durID(pdu) > 32767)
then dUsec(durId(pdu))
else 0 fi
PsIndicate
(addr2(pdu),
pwrmgt(pdu))
Gather Power
management
info from all
valid frames.
dNav:=dUsec
(durId(pdu)),
src:= misc
import(
mActing_
AsAp)
(true)
(false)
ap_
addr
toDs(pdu)
(=1)
(=0)
AP, check
all frames, 2
pages ahead.
durId(pdu)
else
(1:32767)
sta_
addr
Non-AP,
toDS=0 to
accept frame,
next page.
SetNav
(endRx,
dNav, src)
Filter_Idle
Frames with toDs=1 ignored by non-APs,
except Duration/Id field for Nav update.

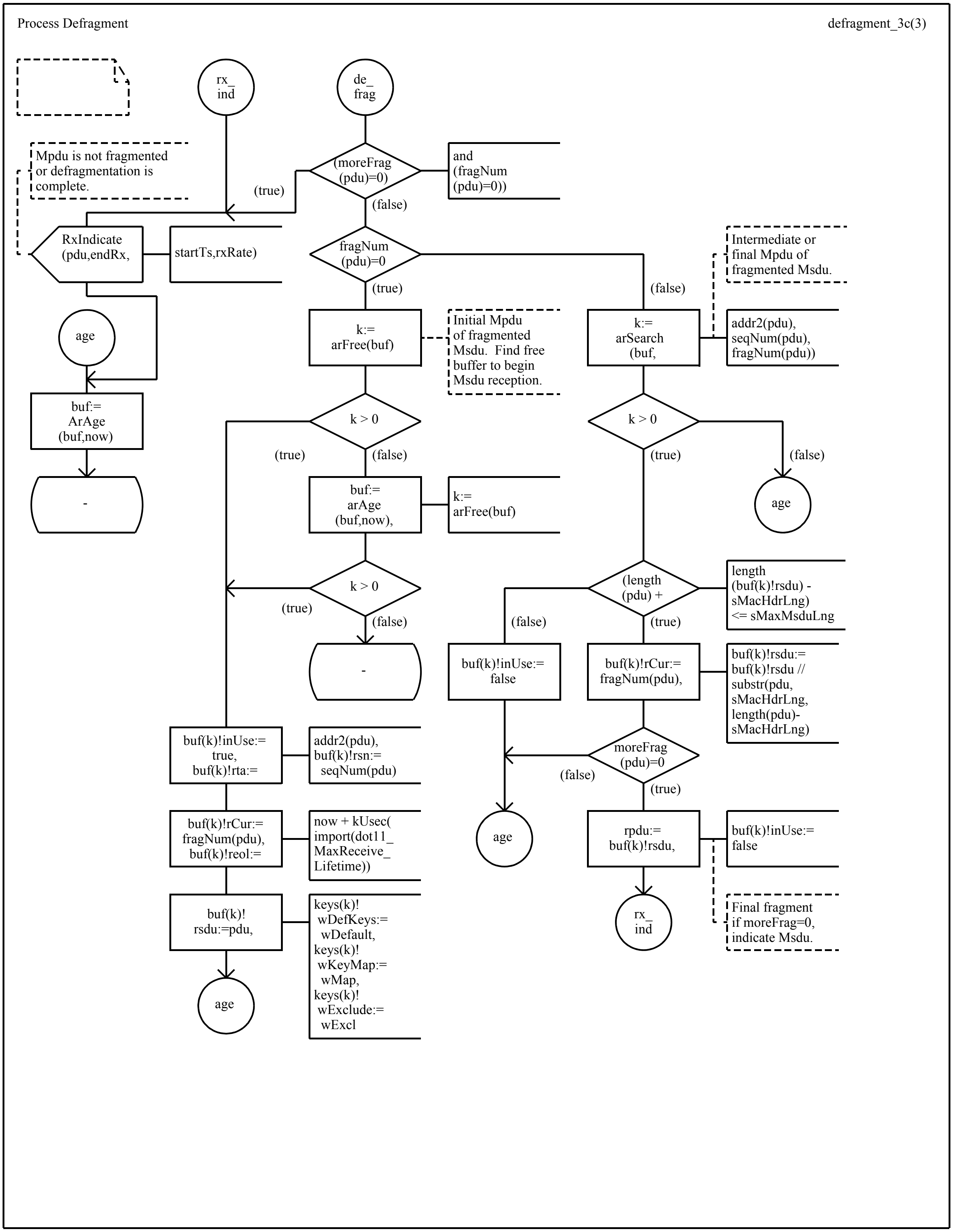

Process Filter_MPDU
filter_sta_2b(4)
sta_
addr
import
(mIbss)
Report probe
requests only
when in Ibss.
other_
bss
(false)
(true)
(probe_req)
sCfPollable
Filter_Idle
multi_
cast
ftype
(pdu)
(false)
(true)
(beacon)
(cfend,
cfend_ack)
else
and ssid(pdu) =
import(mSsid)
and capB(pdu)=cIbss
import
(mIbss)
(true)
(false)
startTs,
rxRate,
wExclude,
wDefault,
wKeyMap)
RxIndicate
(pdu,
endRx,
cfDur_
Rem(pdu)
ClearNav
(cfend_
Other)
ftype(pdu)
else
(cfack,
data_ack,
cfpoll_ack,
data_poll_ack,
cfend_ack)
(>0)
else
Filter_Idle
dNav:=
dUsec(cfDur_
Rem(pdu))
RxCfAck
(addr2(pdu))
Unsolicited
RxCfAck
signals are
ignored.
Report Ibss beacon
with own SsId but
other BssId, for use
based on TSF age.
SetNav
(endRx,dNav,
cfpOther)
Filter_Idle
If addr1 (RA) is
group address,
check the BssId.
isGroup(
addr1(pdu))
(true)
(false)
(false)
import
(mBssId)
= (if frDs(pdu)=1
then addr2(pdu)
else addr3(pdu) fi)
src:= if rts=
ftype(pdu) then
rts else src fi
addr1(pdu)
=
import
(dot11Mac_
Address)
(false)
(true)
(true)
other_
bss
addr1(pdu)
in (
BcAddr or
import(aGro_
upAddresses)
addr3
(pdu) =
import
(mBssId)
(false)
(true)
(false)
(true)
Check for
beacon from
another BSS.
multi_
cast
durId(pdu)
sCfPollable
Ensure that
Cfpoll comes
from own Bss.
(1:
32767)
(49153:
51159)
else
(false)
(true)
Rx broadcast
or multicast,
2 pages ahead.
Filter_Idle
dNav:=
dPsp
Update Nav
for PS-Poll
Ack Dur/Id
field is AID.
else
ftype(pdu)
Ignore frames
addressed to
other groups
in this BSS.
SetNav
(endRx,
dNav, src)
(cfpoll,
cfpoll_ack,
data_poll,
data_poll_
_ack)
Update Nav using
Duration/ID value
from frames to all
other stations.
Else case is for
DurId=32768
in the CF period.
Filter_Idle
RxCfPoll
(endRx,
rxRate)
Only report
cfpoll from
unicast to sta.
uni_
cast
Rx directed
frame to sta,
2 pages ahead.

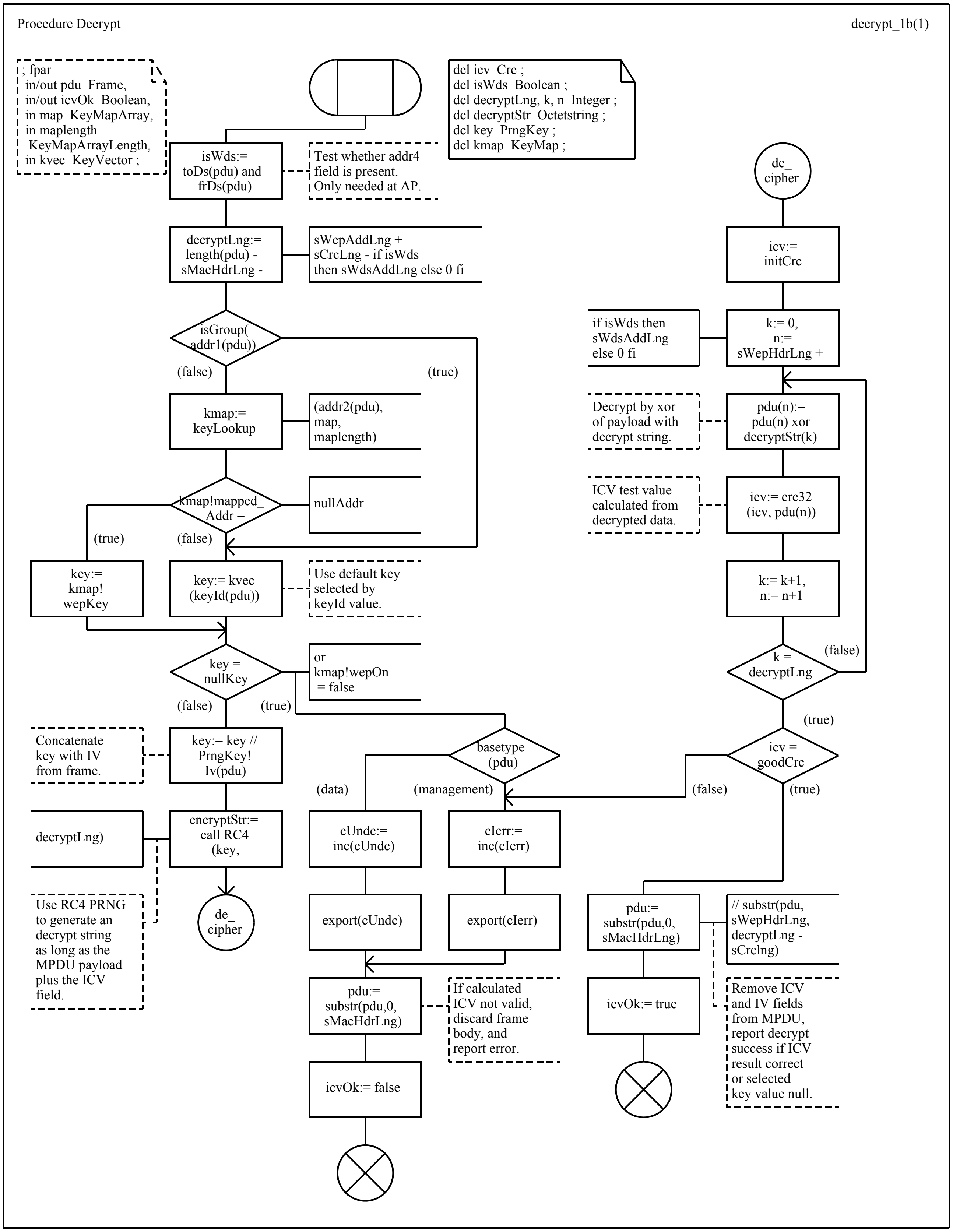

Process Filter_MPDU
filter_ap_3a(4)
ap_ addr
isGroup(addr1(pdu))
All frames to AP are directed except probe_req.
Receive probe request at AP the same as at Ibss station.
(false)
(true)
addr1(pdu) =
import (dot11_ MacAddress)
(false)
(true)
import (mCfp)
(false)
(true)
ftype(pdu)
else
(data_ack, cfend_ack, cfack)
RxCfAck (addr2(pdu))
Unsolicited RxCfAck signals should not occur.
uni_ cast
Rx directed frame to AP, next page.
src:= if rts= ftype(pdu) then rts else src fi
durId(pdu)
(1: 32767)
(49153: 51159)
else
dNav:= dPsp
Nav to cover PS-Poll Ack when DurID field is SID.
SetNav (endRx, dNav, src)
Update Nav using value in Duration/ID field of frames directed to all other stations. Else case is for DurId=32768 in the CF period.
Filter_Idle
ftype (pdu)
(probe_req)
(beacon)
(cfend, cfend_ack)
else
multi_ cast
cfDur_ Rem(pdu)
(>0)
else
dNav:= dUsec(cfDur_ Rem(pdu))
SetNav (endRx,dNav, cfpOther)
ClearNav (cfend_ Other)
Filter_Idle


Process Filter_MPDU
report_rx_4a(4)
uni_
cast
Report incoming directed frames, including all received frames accepted at AP.
Count all valid directed frames to this sta, even those that will be discarded as duplicates or due to WEP.
cRx:= inc(cRx)
export(cRx)
retryBit
(pdu)
(=1)
(=0)
dup:=
searchTupleCache
(cache, addr2(pdu),
seq(pdu), frag(pdu))
dup
(true)
(false)
cDup:=
inc(cDup)
export(cDup)
RxMpdu
(pdu,
endRx,
startTs,
rxRate,
wExclude,
wDefault,
wKeyMap)
Ps-Poll is on else path (as control frame) to allow ack or data as the response from protocol ctl.
basetype
(pdu)
else
(data, management)
NeedAck
(addr2(pdu),
endRx,dAck)
Directed Atim frames must be acknowledged, but may be omitted from cache, see 9.2.9.
New cache entry if (addr2,seq) is not cached. If entry exists for (addr2,seq), update time and fragment number of entry.
cache:=
updateTupleCache
(cache, addr2(pdu),
seq(pdu),frag(pdu),
endRx)
Filter_Idle
multi_
cast
Report incoming group-addressed frames at station.
cRx:= inc(cRx),
cMc:= inc(cMc)
Count all valid broadcast and multicast frames to this sta, even those that will be discarded due to WEP.
export
(cRx, cMc)
ftype(pdu)
(beacon)
else
(cfend,
cfend_ack)
cfDurRem
(pdu)
(>0)
ClearNav
(cfendBss)
dNav:=
dUsec(cfDur_
Rem(pdu))
SetNav
(endRx,dNav,
cfpBss)
RxMpdu
(pdu,
endRx,
startTs,
rxRate,
wExclude,
wDefault,
wKeyMap)
Filter_Idle


Process Defragment
wep_filter_1b(3)
Decrypt
dcl exported cIerr as dot11WepIcvErrorCount,
cUndc as dot11WepUndecryptableCount,
cExcl as dot11WepExcludedCount Counter32:= 0 ;
dLife:= dUsec(import
(dot11Max_ Receive_ Lifetime))
imported mCfp Boolean ;
imported dot11MaxReceiveLifetime Kusec ;
imported procedure RC4 ; fpar PrngKey, Integer ;
returns Octetstring ;
export(cIerr, cUndc, cExcl)
dcl buf DefragArray ;
dcl dLife Duration ;
dcl endRx, startTs Time ;
dcl icvOk Boolean ;
dcl k DefragIndex ;
dcl keys DefragKeysArray ;
dcl pri CfPriority ;
dcl pdu, sdu Frame ;
dcl wExcl Boolean ;
dcl wDefault KeyVector ;
dcl wMap KeyMapArray ;
buf:= ArAge(buf, now+dLife+1)
Defragmentation buffers forced empty using the aging function.
Defrag_ Inactive
not import (mDisable)
mDisable=false when started or joined Bss.
RxMpdu (pdu, endRx,
startTs,rxRate, wExcl,wDefault, wMap)
Defrag_ Idle
else
ftype (pdu)
When not in Bss only pass beacon and probe_rsp.
(beacon, probe_rsp)
RxIndicate (pdu,endRx,
startTs,rxRate)
import (mDisable)
RxMpdu (pdu,endRx, startTs,rxRate,
wExcl,wDefault, wMap)
-
basetype (pdu)
(control)
(management)
else
wepBit (pdu)
(=0)
(=1)
wepBit (pdu)
(=1)
(=0)
rx_ ind
import(
dot11Privacy_ Option_ Implemented)
wExcl
(false)
(true)
(true)
(false)
auth) and authSeqNum (pdu)=3) and import(dot11Privacy_ Option_ Implemented)
(ftype (pdu)=
(false)
cUndc:= inc(cUndc)
de_ crypt
cExcl:= inc(cExcl)
de_ frag
(true)
de_ crypt
export(cUndc)
export(cExcl)
-
-


Process Defragment
wep_decrypt_2b(3)
de_
crypt
Decrypt
(pdu,
icvOk,
wMap, sKey_
MappingLength,
wDefault)
icvOk
Icv errors and
certain undecryptable
errors counted in
Decrypt procedure.
(true)
(false)
de_
frag
ftype
(pdu)
else
(auth)
RxIndicate
(pdu,endRx,
startTs,rxRate)
Do not report
receipt of
data frames
with Icv errors.
-
Authentication
challenge resposnes
with Icv errors
are reported, but
Decrypt removes
payload so Auth
service is able
to distinguish
a bad key from
a nonresponse.


Process Defragment
defragment_3c(3)
rx_ ind
de_ frag
Mpdu is not fragmented or defragmentation is complete.
(moreFrag (pdu)=0)
and (fragNum (pdu)=0))
(true)
(false)
RxIndicate (pdu,endRx,
startTs,rxRate)
fragNum (pdu)=0
(true)
(false)
Intermediate or final Mpdu of fragmented Msdu.
age
k:= arFree(buf)
Initial Mpdu of fragmented Msdu. Find free buffer to begin Msdu reception.
k:= arSearch (buf,
addr2(pdu), seqNum(pdu), fragNum(pdu))
buf:= ArAge (buf,now)
k > 0
(true)
(false)
k > 0
(true)
(false)
-
buf:= arAge (buf,now),
k:= arFree(buf)
age
k > 0
(true)
(false)
(length (pdu) +
length (buf(k)!rsdu) - sMacHdrLng) <= sMaxMsduLng
(false)
(true)
-
buf(k)!inUse:= false
buf(k)!rCur:= fragNum(pdu),
buf(k)!rsdu:= buf(k)!rsdu // substr(pdu, sMacHdrLng, length(pdu)- sMacHdrLng)
buf(k)!inUse:= true, buf(k)!rta:=
addr2(pdu), buf(k)!rsn:= seqNum(pdu)
moreFrag (pdu)=0
(false)
(true)
buf(k)!rCur:= fragNum(pdu), buf(k)!reol:=
now + kUsec(import(dot11_ MaxReceive_ Lifetime))
age
rpdu:= buf(k)!rsdu,
buf(k)!inUse:= false
buf(k)! rsdu:=pdu,
keys(k)! wDefKeys:= wDefault, keys(k)! wKeyMap:= wMap, keys(k)! wExclude:= wExcl
rx_ ind
Final fragment if moreFrag=0, indicate Msdu.
age


Procedure Decrypt
decrypt_1b(1)
; fpar
in/out pdu Frame,
in/out icvOk Boolean,
in map KeyMapArray,
in maplength
KeyMapArrayLength,
in kvec KeyVector ;
dcl icv Crc ;
dcl isWds Boolean ;
dcl decryptLng, k, n Integer ;
dcl decryptStr Octetstring ;
dcl key PrngKey ;
dcl kmap KeyMap ;
isWds:=
toDs(pdu) and
frDs(pdu)
Test whether addr4
field is present.
Only needed at AP.
decryptLng:=
length(pdu) -
sMacHdrLng -
sWepAddLng +
sCrcLng - if isWds
then sWdsAddLng else 0 fi
isGroup(
addr1(pdu))
(false)
(true)
kmap:=
keyLookup
(addr2(pdu),
map,
maplength)
kmap!mapped_
Addr =
nullAddr
(true)
(false)
key:=
kmap!
wepKey
key:= kvec
(keyId(pdu))
Use default key
selected by
keyId value.
key =
nullKey
or
kmap!wepOn
= false
(false)
(true)
Concatenate
key with IV
from frame.
key:= key //
PrngKey!
Iv(pdu)
decryptLng)
encryptStr:=
call RC4
(key,
Use RC4 PRNG
to generate an
decrypt string
as long as the
MPDU payload
plus the ICV
field.
de_
cipher
basetype
(pdu)
(data)
(management)
cUndc:=
inc(cUndc)
cIerr:=
inc(cIerr)
export(cUndc)
export(cIerr)
pdu:=
substr(pdu,0,
sMacHdrLng)
If calculated
ICV not valid,
discard frame
body, and
report error.
icvOk:= false
de_
cipher
icv:=
initCrc
if isWds then
sWdsAddLng
else 0 fi
k:= 0,
n:=
sWepHdrLng +
Decrypt by xor
of payload with
decrypt string.
pdu(n):=
pdu(n) xor
decryptStr(k)
ICV test value
calculated from
decrypted data.
icv:= crc32
(icv, pdu(n))
k:= k+1,
n:= n+1
k =
decryptLng
(false)
(true)
icv =
goodCrc
(false)
(true)
pdu:=
substr(pdu,0,
sMacHdrLng)
// substr(pdu,
sWepHdrLng,
decryptLng -
sCrclng)
icvOk:= true
Remove ICV
and IV fields
from MPDU,
report decrypt
success if ICV
result correct
or selected
key value null.

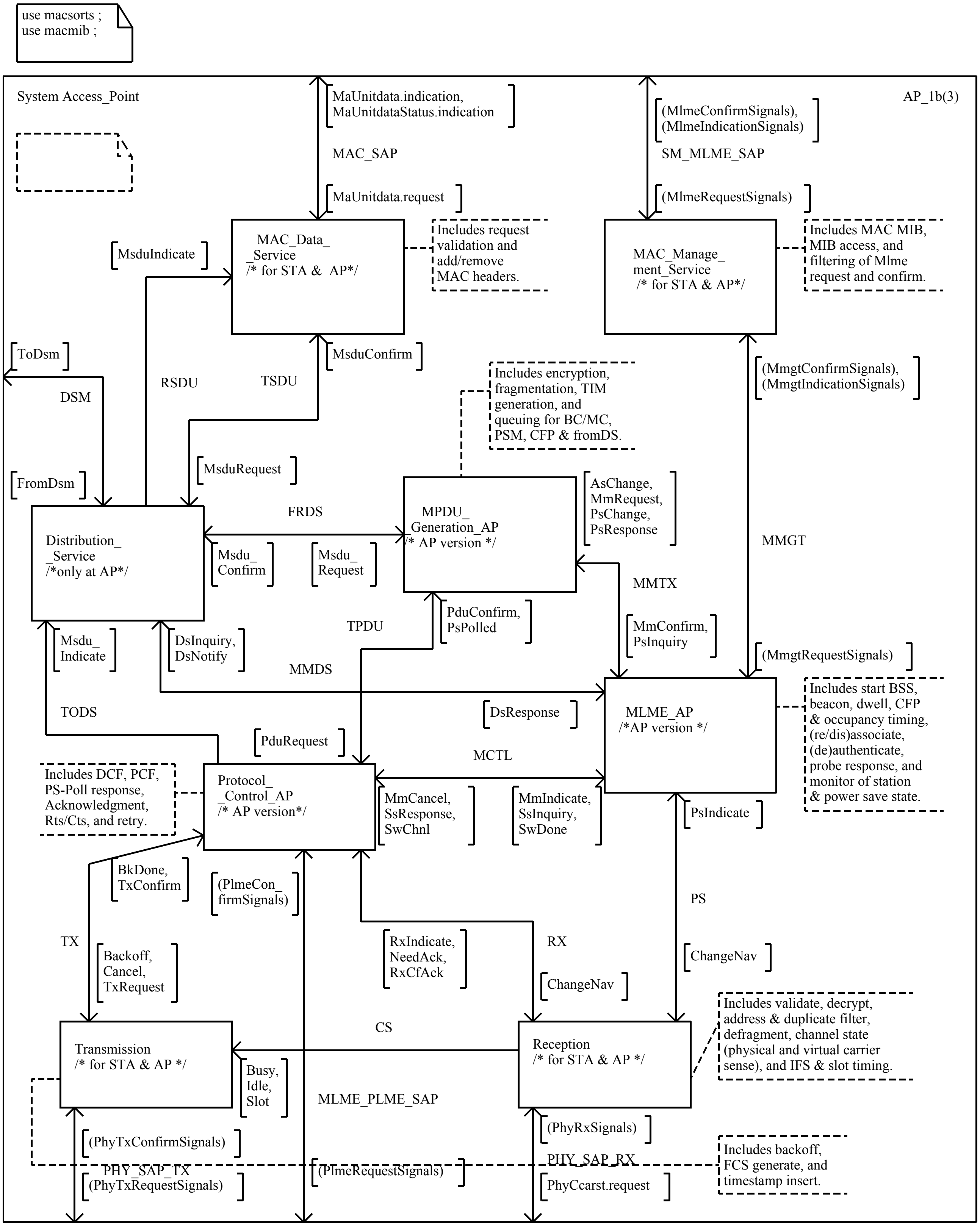
J.6 State machines for MAC AP

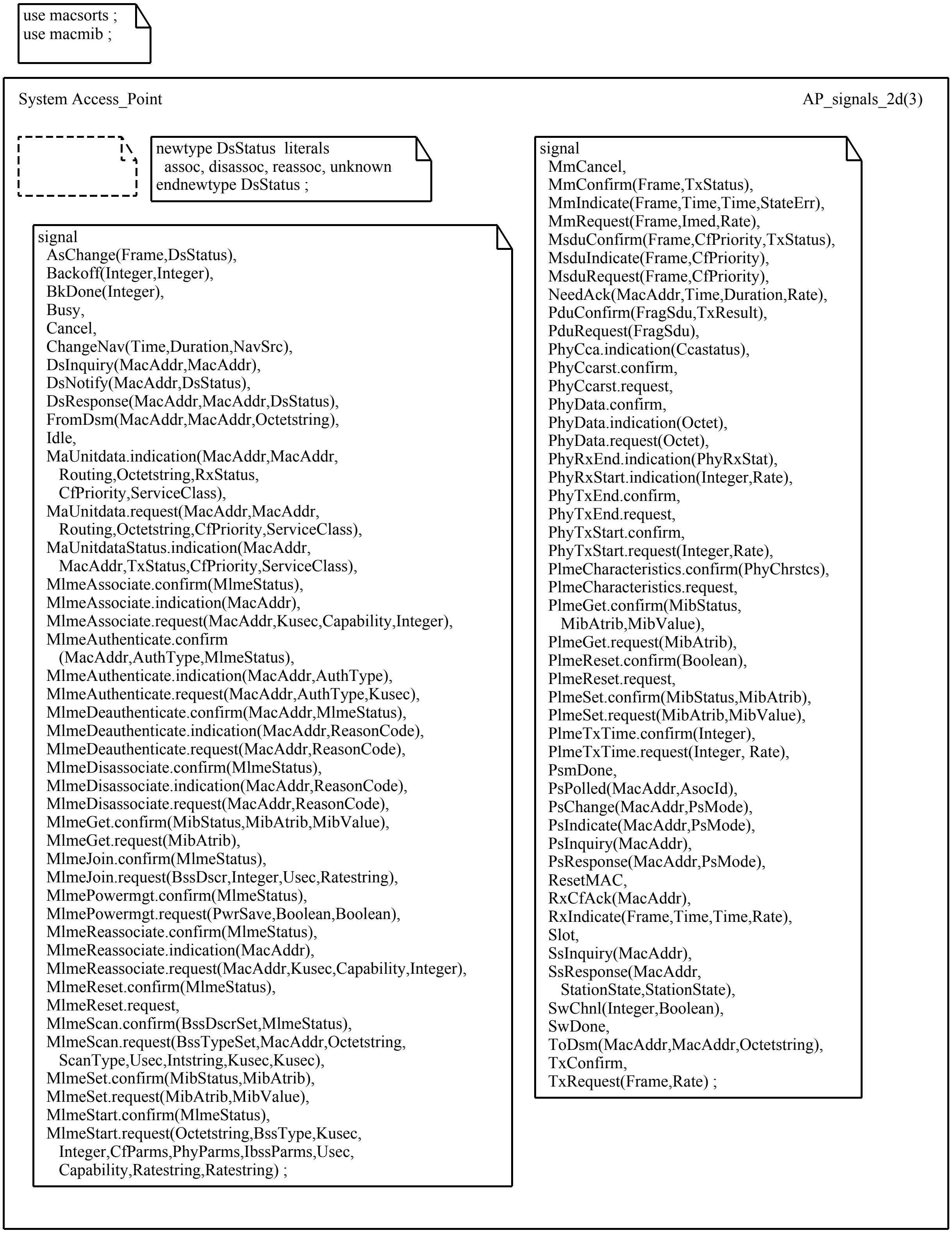
The following SDL-92 system specification defines operation of the MAC protocol at an IEEE 802.11 AP. Many aspects of AP operation are identical to the STA operation. These are defined in blocks and processes referenced from both the STA and AP system specifications. Blocks and processes used in both STA and AP are identifiable by the SDL comment /* for STA & AP */ below the block or process name. Blocks and processes specific to AP operation are identifiable by the SDL comment /* AP version */ below the block or process name. Definitions for the /* AP version */ and the /* STA & AP */ blocks and processes appear in this subclause.

The remainder of this subclause is the formal description, in SDL/GR, of an IEEE 802.11 AP.

This subclause describes the security behavior of only 11.2.2 and 11.2.3.

This subclause does not describe the behavior of a STA with QoS facility.


use macsorts ;
use macmib ;
System Access_Point
AP_1b(3)
MaUnitdata.indication,
MaUnitdataStatus.indication
MAC_SAP
MaUnitdata.request
(MlmeConfirmSignals),
(MlmeIndicationSignals)
SM_MLME_SAP
(MlmeRequestSignals)
MsduIndicate
MAC_Data_
_Service
/* for STA & AP*/
Includes request validation and add/remove MAC headers.
MAC_Manage_
ment_Service
/* for STA & AP*/
Includes MAC MIB, MIB access, and filtering of Mlme request and confirm.
ToDsm
DSM
RSDU
TSDU
MsduConfirm
(MmgtConfirmSignals),
(MmgtIndicationSignals)
Includes encryption, fragmentation, TIM generation, and queuing for BC/MC, PSM, CFP & fromDS.
MsduRequest
FromDsm
FRDS
MPDU_
Generation_AP
/* AP version */
AsChange,
MmRequest,
PsChange,
PsResponse
MMGT
Distribution_
_Service
/*only at AP*/
Msdu_
Confirm
Msdu_
Request
MMTX
PduConfirm,
PsPolled
TPDU
MmConfirm,
PsInquiry
Msdu_
Indicate
DsInquiry,
DsNotify
MMDS
(MmgtRequestSignals)
TODS
DsResponse
MLME_AP
/*AP version */
Includes start BSS, beacon, dwell, CFP & occupancy timing, (re/dis)associate, (de)authenticate, probe response, and monitor of station & power save state.
PduRequest
MCTL
Includes DCF, PCF, PS-Poll response, Acknowledgment, Rts/Cts, and retry.
Protocol_
_Control_AP
/* AP version*/
MmCancel,
SsResponse,
SwChnl
MmIndicate,
SsInquiry,
SwDone
PsIndicate
BkDone,
TxConfirm
(PlmeCon_
firmSignals)
PS
TX
Backoff,
Cancel,
TxRequest
RxIndicate,
NeedAck,
RxCfAck
RX
ChangeNav
ChangeNav
CS
Includes validate, decrypt, address & duplicate filter, defragment, channel state (physical and virtual carrier sense), and IFS & slot timing.
Transmission
/* for STA & AP */
Busy,
Idle,
Slot
MLME_PLME_SAP
Reception
/* for STA & AP */
(PhyTxConfirmSignals)
(PhyRxSignals)
PHY_SAP_TX
(PhyTxRequestSignals)
(PlmeRequestSignals)
PHY_SAP_RX
PhyCcarst.request
Includes backoff, FCS generate, and timestamp insert.



```
use macsorts ;
use macmib ;
```

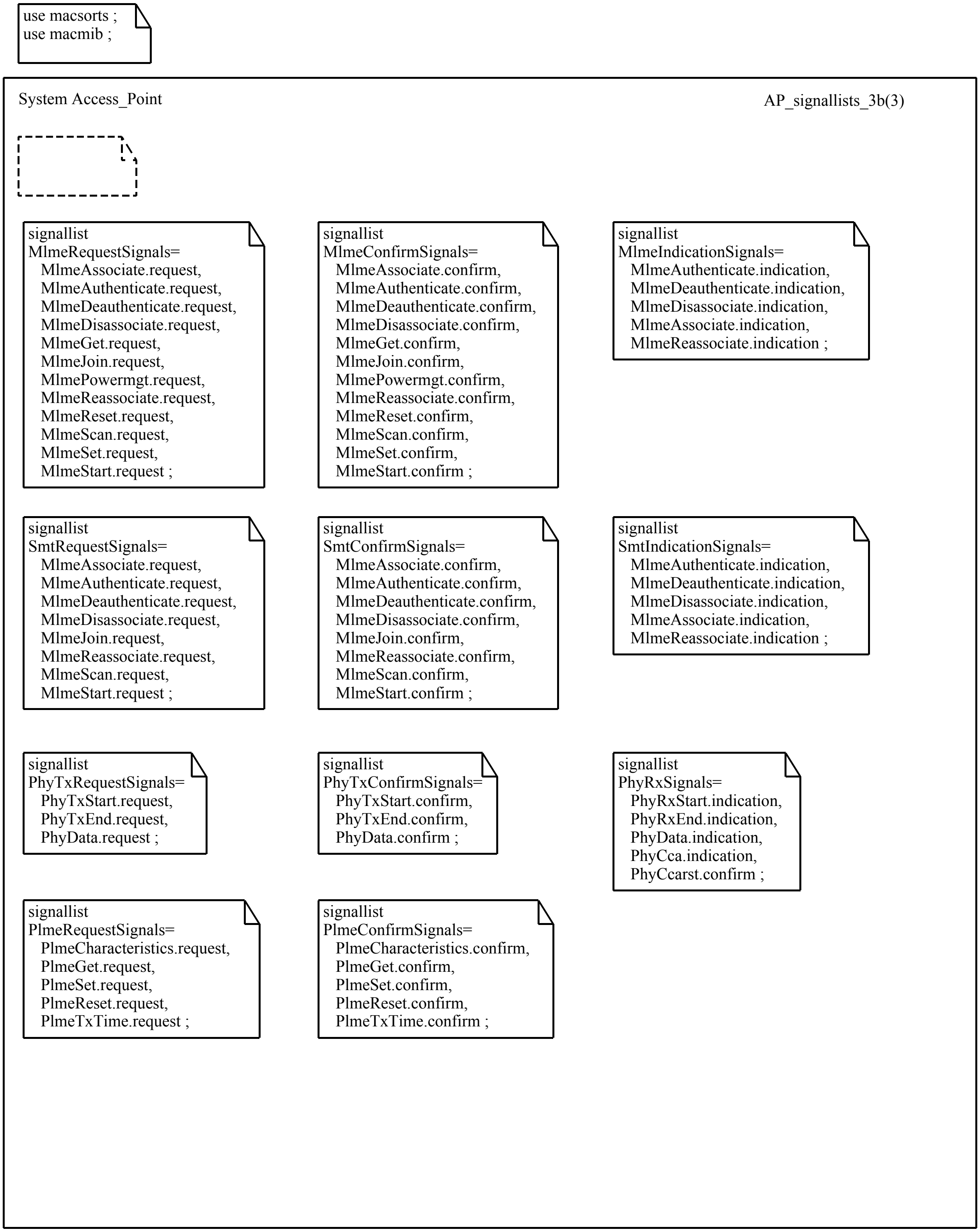
System Access_Point

AP_signals_2d(3)

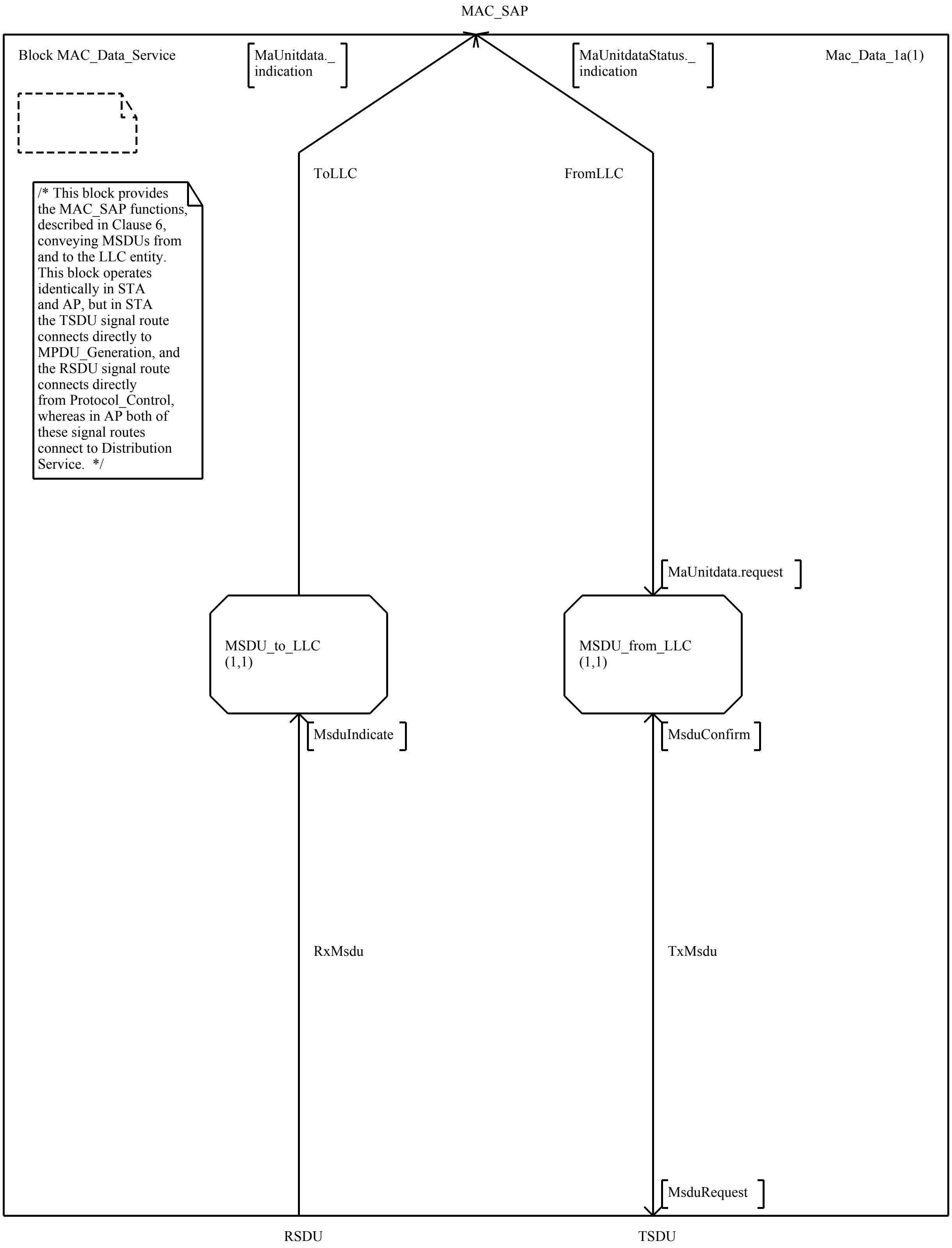
```
newtype DsStatus  literals
  assoc, disassoc, reassoc, unknown
endnewtype DsStatus ;
```

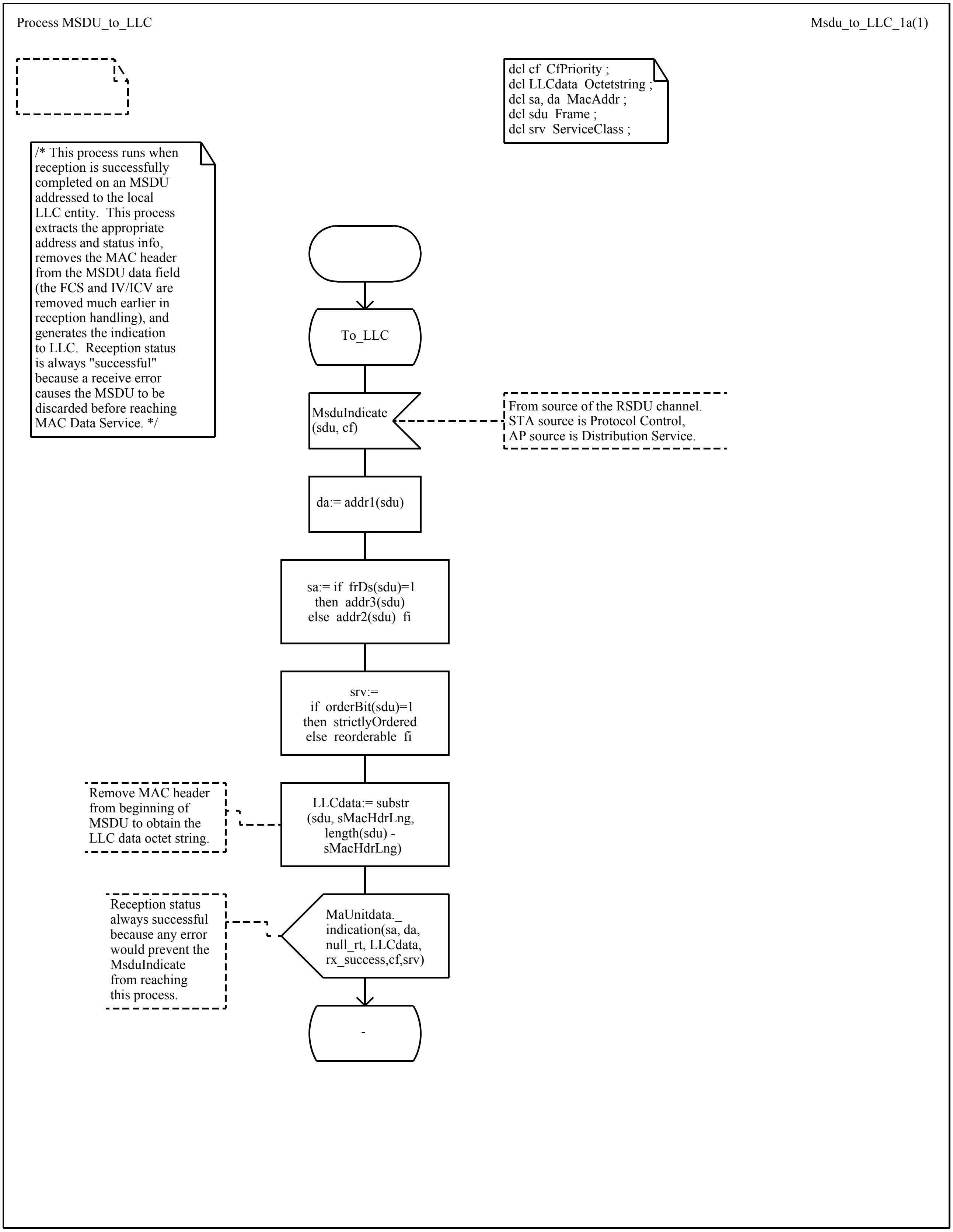
```
signal
  AsChange(Frame,DsStatus),
  Backoff(Integer,Integer),
  BkDone(Integer),
  Busy,
  Cancel,
  ChangeNav(Time,Duration,NavSrc),
  DsInquiry(MacAddr,MacAddr),
  DsNotify(MacAddr,DsStatus),
  DsResponse(MacAddr,MacAddr,DsStatus),
  FromDsm(MacAddr,MacAddr,Octetstring),
  Idle,
  MaUnitdata.indication(MacAddr,MacAddr,
    Routing,Octetstring,RxStatus,
    CfPriority,ServiceClass),
  MaUnitdata.request(MacAddr,MacAddr,
    Routing,Octetstring,CfPriority,ServiceClass),
  MaUnitdataStatus.indication(MacAddr,
    MacAddr,TxStatus,CfPriority,ServiceClass),
  MlmeAssociate.confirm(MlmeStatus),
  MlmeAssociate.indication(MacAddr),
  MlmeAssociate.request(MacAddr,Kusec,Capability,Integer),
  MlmeAuthenticate.confirm
    (MacAddr,AuthType,MlmeStatus),
  MlmeAuthenticate.indication(MacAddr,AuthType),
  MlmeAuthenticate.request(MacAddr,AuthType,Kusec),
  MlmeDeauthenticate.confirm(MacAddr,MlmeStatus),
  MlmeDeauthenticate.indication(MacAddr,ReasonCode),
  MlmeDeauthenticate.request(MacAddr,ReasonCode),
  MlmeDisassociate.confirm(MlmeStatus),
  MlmeDisassociate.indication(MacAddr,ReasonCode),
  MlmeDisassociate.request(MacAddr,ReasonCode),
  MlmeGet.confirm(MibStatus,MibAtrib,MibValue),
  MlmeGet.request(MibAtrib),
  MlmeJoin.confirm(MlmeStatus),
  MlmeJoin.request(BssDscr,Integer,Usec,Ratestring),
  MlmePowermgt.confirm(MlmeStatus),
  MlmePowermgt.request(PwrSave,Boolean,Boolean),
  MlmeReassociate.confirm(MlmeStatus),
  MlmeReassociate.indication(MacAddr),
  MlmeReassociate.request(MacAddr,Kusec,Capability,Integer),
  MlmeReset.confirm(MlmeStatus),
  MlmeReset.request,
  MlmeScan.confirm(BssDscrSet,MlmeStatus),
  MlmeScan.request(BssTypeSet,MacAddr,Octetstring,
    ScanType,Usec,Intstring,Kusec,Kusec),
  MlmeSet.confirm(MibStatus,MibAtrib),
  MlmeSet.request(MibAtrib,MibValue),
  MlmeStart.confirm(MlmeStatus),
  MlmeStart.request(Octetstring,BssType,Kusec,
    Integer,CfParms,PhyParms,IbssParms,Usec,
    Capability,Ratestring,Ratestring) ;
```

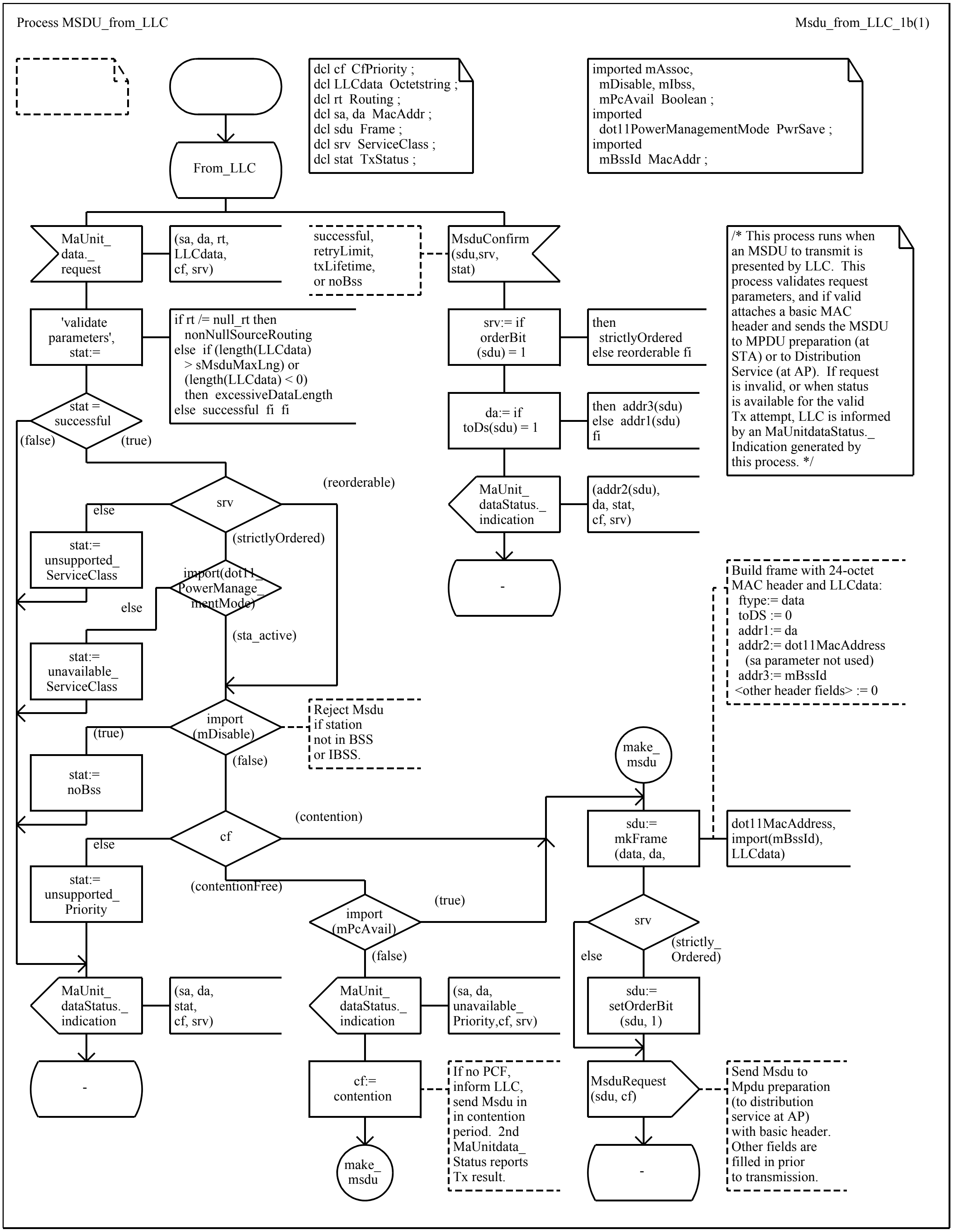
```
signal
  MmCancel,
  MmConfirm(Frame,TxStatus),
  MmIndicate(Frame,Time,Time,StateErr),
  MmRequest(Frame,Imed,Rate),
  MsduConfirm(Frame,CfPriority,TxStatus),
  MsduIndicate(Frame,CfPriority),
  MsduRequest(Frame,CfPriority),
  NeedAck(MacAddr,Time,Duration,Rate),
  PduConfirm(FragSdu,TxResult),
  PduRequest(FragSdu),
  PhyCca.indication(Ccastatus),
  PhyCcarst.confirm,
  PhyCcarst.request,
  PhyData.confirm,
  PhyData.indication(Octet),
  PhyData.request(Octet),
  PhyRxEnd.indication(PhyRxStat),
  PhyRxStart.indication(Integer,Rate),
  PhyTxEnd.confirm,
  PhyTxEnd.request,
  PhyTxStart.confirm,
  PhyTxStart.request(Integer,Rate),
  PlmeCharacteristics.confirm(PhyChrstcs),
  PlmeCharacteristics.request,
  PlmeGet.confirm(MibStatus,
    MibAtrib,MibValue),
  PlmeGet.request(MibAtrib),
  PlmeReset.confirm(Boolean),
  PlmeReset.request,
  PlmeSet.confirm(MibStatus,MibAtrib),
  PlmeSet.request(MibAtrib,MibValue),
  PlmeTxTime.confirm(Integer),
  PlmeTxTime.request(Integer, Rate),
  PsmDone,
  PsPolled(MacAddr,AsocId),
  PsChange(MacAddr,PsMode),
  PsIndicate(MacAddr,PsMode),
  PsInquiry(MacAddr),
  PsResponse(MacAddr,PsMode),
  ResetMAC,
  RxCfAck(MacAddr),
  RxIndicate(Frame,Time,Time,Rate),
  Slot,
  SsInquiry(MacAddr),
  SsResponse(MacAddr,
    StationState,StationState),
  SwChnl(Integer,Boolean),
  SwDone,
  ToDsm(MacAddr,MacAddr,Octetstring),
  TxConfirm,
  TxRequest(Frame,Rate) ;
```

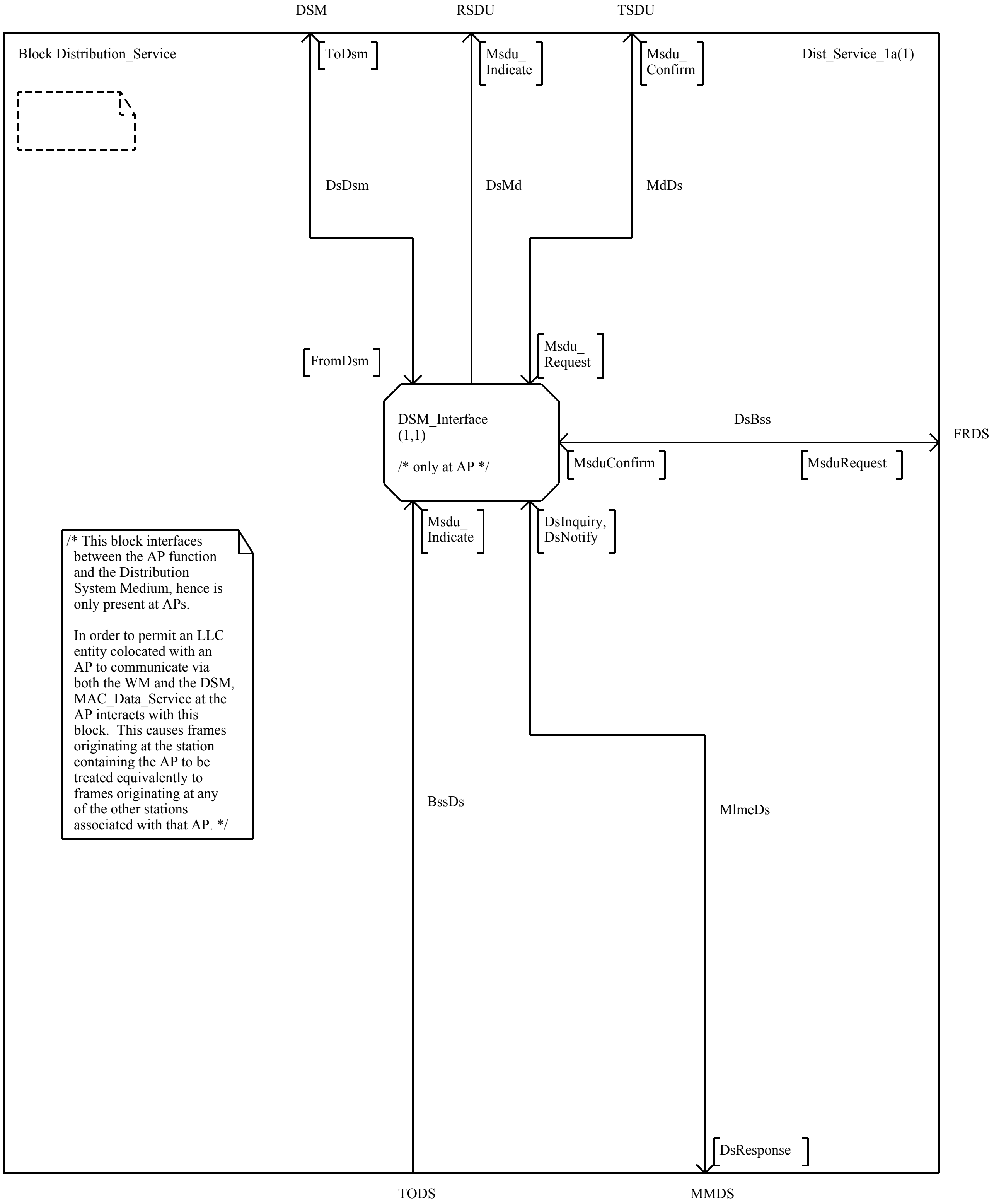


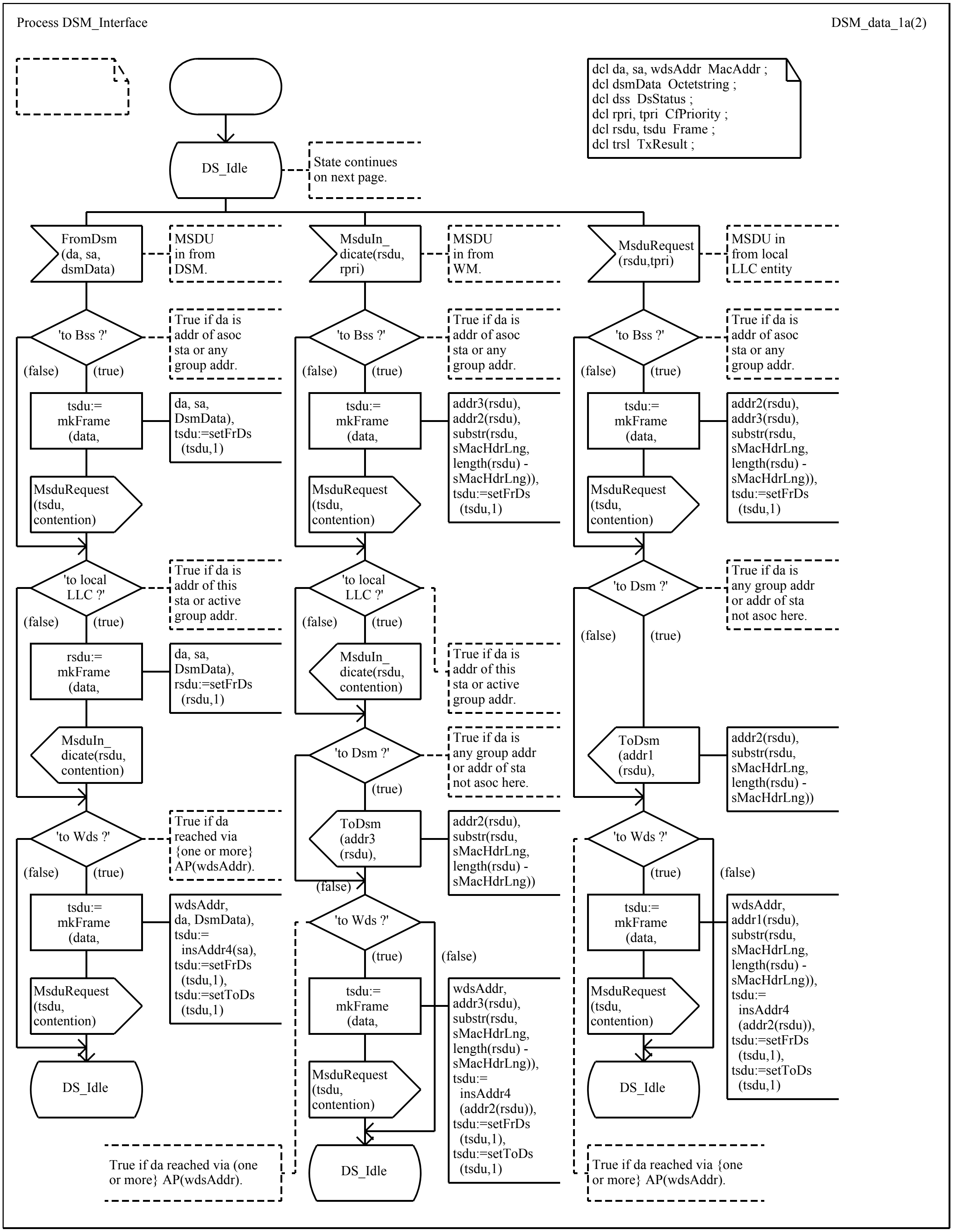
```
use macsorts ;
use macmib ;

System Access_Point                                    AP_signallists_3b(3)

signallist
MlmeRequestSignals=
  MlmeAssociate.request,
  MlmeAuthenticate.request,
  MlmeDeauthenticate.request,
  MlmeDisassociate.request,
  MlmeGet.request,
  MlmeJoin.request,
  MlmePowermgt.request,
  MlmeReassociate.request,
  MlmeReset.request,
  MlmeScan.request,
  MlmeSet.request,
  MlmeStart.request ;

signallist
MlmeConfirmSignals=
  MlmeAssociate.confirm,
  MlmeAuthenticate.confirm,
  MlmeDeauthenticate.confirm,
  MlmeDisassociate.confirm,
  MlmeGet.confirm,
  MlmeJoin.confirm,
  MlmePowermgt.confirm,
  MlmeReassociate.confirm,
  MlmeReset.confirm,
  MlmeScan.confirm,
  MlmeSet.confirm,
  MlmeStart.confirm ;

signallist
MlmeIndicationSignals=
  MlmeAuthenticate.indication,
  MlmeDeauthenticate.indication,
  MlmeDisassociate.indication,
  MlmeAssociate.indication,
  MlmeReassociate.indication ;

signallist
SmtRequestSignals=
  MlmeAssociate.request,
  MlmeAuthenticate.request,
  MlmeDeauthenticate.request,
  MlmeDisassociate.request,
  MlmeJoin.request,
  MlmeReassociate.request,
  MlmeScan.request,
  MlmeStart.request ;

signallist
SmtConfirmSignals=
  MlmeAssociate.confirm,
  MlmeAuthenticate.confirm,
  MlmeDeauthenticate.confirm,
  MlmeDisassociate.confirm,
  MlmeJoin.confirm,
  MlmeReassociate.confirm,
  MlmeScan.confirm,
  MlmeStart.confirm ;

signallist
SmtIndicationSignals=
  MlmeAuthenticate.indication,
  MlmeDeauthenticate.indication,
  MlmeDisassociate.indication,
  MlmeAssociate.indication,
  MlmeReassociate.indication ;

signallist
PhyTxRequestSignals=
  PhyTxStart.request,
  PhyTxEnd.request,
  PhyData.request ;

signallist
PhyTxConfirmSignals=
  PhyTxStart.confirm,
  PhyTxEnd.confirm,
  PhyData.confirm ;

signallist
PhyRxSignals=
  PhyRxStart.indication,
  PhyRxEnd.indication,
  PhyData.indication,
  PhyCca.indication,
  PhyCcarst.confirm ;

signallist
PlmeRequestSignals=
  PlmeCharacteristics.request,
  PlmeGet.request,
  PlmeSet.request,
  PlmeReset.request,
  PlmeTxTime.request ;

signallist
PlmeConfirmSignals=
  PlmeCharacteristics.confirm,
  PlmeGet.confirm,
  PlmeSet.confirm,
  PlmeReset.confirm,
  PlmeTxTime.confirm ;
```

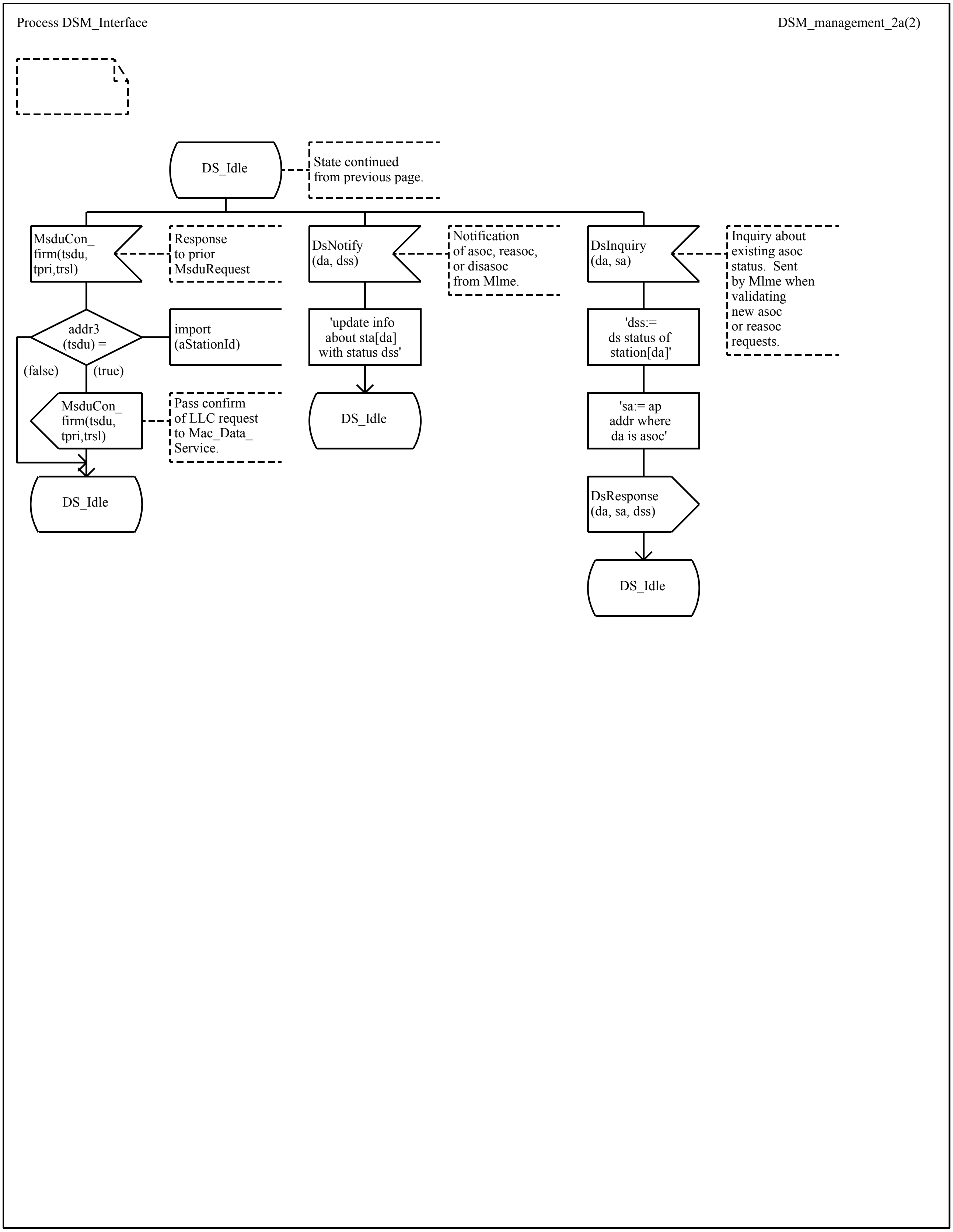

MAC_SAP
Block MAC_Data_Service
MaUnitdata._
indication
MaUnitdataStatus._
indication
Mac_Data_1a(1)
ToLLC
FromLLC
/* This block provides the MAC_SAP functions, described in Clause 6, conveying MSDUs from and to the LLC entity. This block operates identically in STA and AP, but in STA the TSDU signal route connects directly to MPDU_Generation, and the RSDU signal route connects directly from Protocol_Control, whereas in AP both of these signal routes connect to Distribution Service. */
MaUnitdata.request
MSDU_to_LLC
(1,1)
MSDU_from_LLC
(1,1)
MsduIndicate
MsduConfirm
RxMsdu
TxMsdu
MsduRequest
RSDU
TSDU

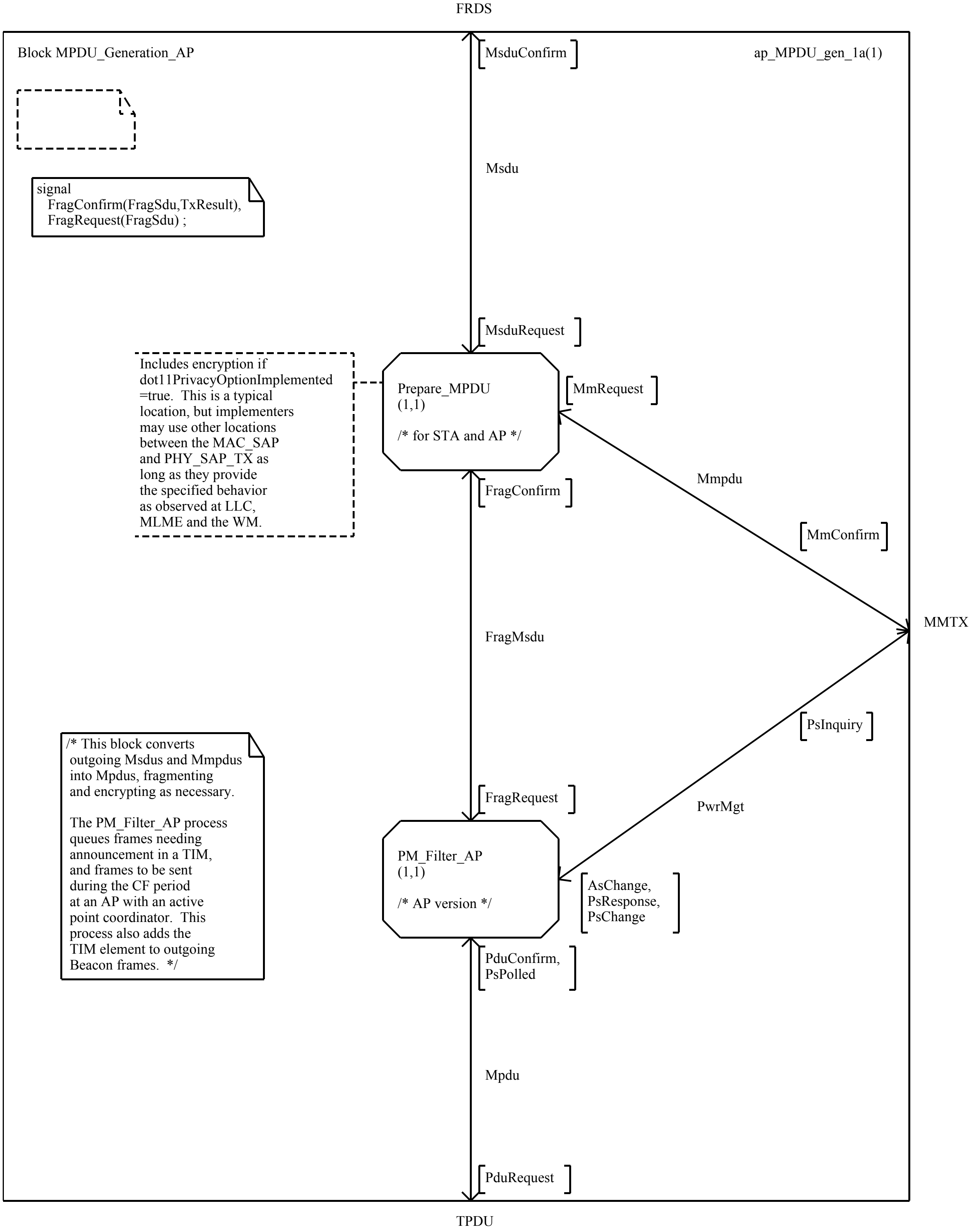

Process MSDU_to_LLC
Msdu_to_LLC_1a(1)
dcl cf CfPriority ;
dcl LLCdata Octetstring ;
dcl sa, da MacAddr ;
dcl sdu Frame ;
dcl srv ServiceClass ;
/* This process runs when reception is successfully completed on an MSDU addressed to the local LLC entity. This process extracts the appropriate address and status info, removes the MAC header from the MSDU data field (the FCS and IV/ICV are removed much earlier in reception handling), and generates the indication to LLC. Reception status is always "successful" because a receive error causes the MSDU to be discarded before reaching MAC Data Service. */
To_LLC
MsduIndicate (sdu, cf)
From source of the RSDU channel. STA source is Protocol Control, AP source is Distribution Service.
da:= addr1(sdu)
sa:= if frDs(sdu)=1 then addr3(sdu) else addr2(sdu) fi
srv:= if orderBit(sdu)=1 then strictlyOrdered else reorderable fi
Remove MAC header from beginning of MSDU to obtain the LLC data octet string.
LLCdata:= substr (sdu, sMacHdrLng, length(sdu) - sMacHdrLng)
Reception status always successful because any error would prevent the MsduIndicate from reaching this process.
MaUnitdata._ indication(sa, da, null_rt, LLCdata, rx_success,cf,srv)
-

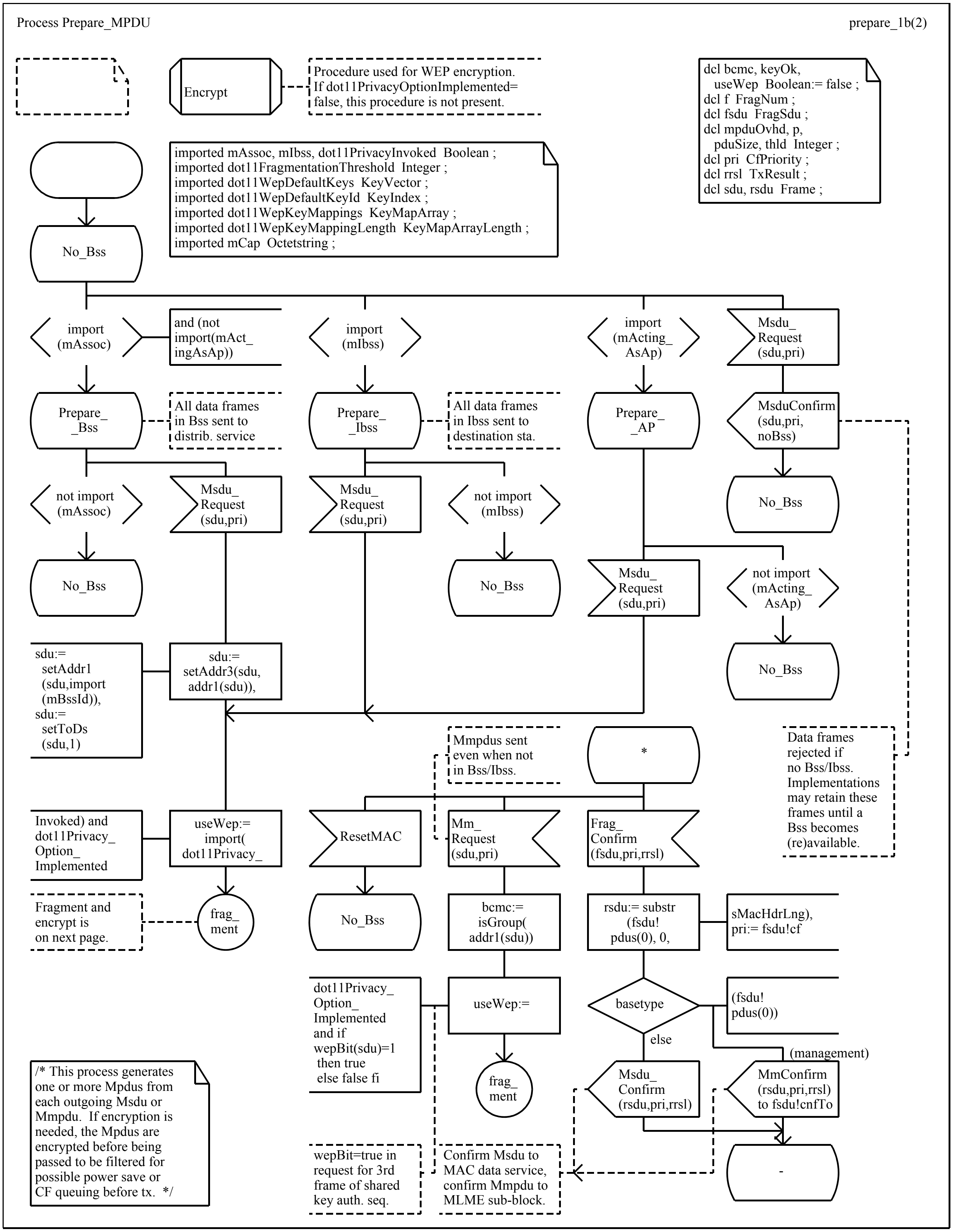

Process MSDU_from_LLC
Msdu_from_LLC_1b(1)
dcl cf CfPriority ;
dcl LLCdata Octetstring ;
dcl rt Routing ;
dcl sa, da MacAddr ;
dcl sdu Frame ;
dcl srv ServiceClass ;
dcl stat TxStatus ;
imported mAssoc,
mDisable, mIbss,
mPcAvail Boolean ;
imported
dot11PowerManagementMode PwrSave ;
imported
mBssId MacAddr ;
From_LLC
MaUnit_ data._ request
(sa, da, rt, LLCdata, cf, srv)
successful, retryLimit, txLifetime, or noBss
MsduConfirm (sdu,srv, stat)
/* This process runs when an MSDU to transmit is presented by LLC. This process validates request parameters, and if valid attaches a basic MAC header and sends the MSDU to MPDU preparation (at STA) or to Distribution Service (at AP). If request is invalid, or when status is available for the valid Tx attempt, LLC is informed by an MaUnitdataStatus._ Indication generated by this process. */
'validate parameters', stat:=
if rt /= null_rt then
nonNullSourceRouting
else if (length(LLCdata)
> sMsduMaxLng) or
(length(LLCdata) < 0)
then excessiveDataLength
else successful fi fi
srv:= if orderBit (sdu) = 1
then strictlyOrdered else reorderable fi
da:= if toDs(sdu) = 1
then addr3(sdu) else addr1(sdu) fi
stat = successful
(false)
(true)
MaUnit_ dataStatus._ indication
(addr2(sdu), da, stat, cf, srv)
(reorderable)
srv
else
(strictlyOrdered)
stat:= unsupported_ ServiceClass
import(dot11_ PowerManage_ mentMode)
else
-
Build frame with 24-octet MAC header and LLCdata:
ftype:= data
toDS := 0
addr1:= da
addr2:= dot11MacAddress
(sa parameter not used)
addr3:= mBssId
<other header fields> := 0
(sta_active)
stat:= unavailable_ ServiceClass
import (mDisable)
(true)
(false)
Reject Msdu if station not in BSS or IBSS.
make_ msdu
stat:= noBss
(contention)
cf
else
sdu:= mkFrame (data, da,
dot11MacAddress, import(mBssId), LLCdata)
stat:= unsupported_ Priority
(contentionFree)
import (mPcAvail)
(true)
srv
else
(strictly_ Ordered)
(false)
MaUnit_ dataStatus._ indication
(sa, da, stat, cf, srv)
MaUnit_ dataStatus._ indication
(sa, da, unavailable_ Priority,cf, srv)
sdu:= setOrderBit (sdu, 1)
-
cf:= contention
If no PCF, inform LLC, send Msdu in in contention period. 2nd MaUnitdata_ Status reports Tx result.
MsduRequest (sdu, cf)
Send Msdu to Mpdu preparation (to distribution service at AP) with basic header. Other fields are filled in prior to transmission.
make_ msdu
-

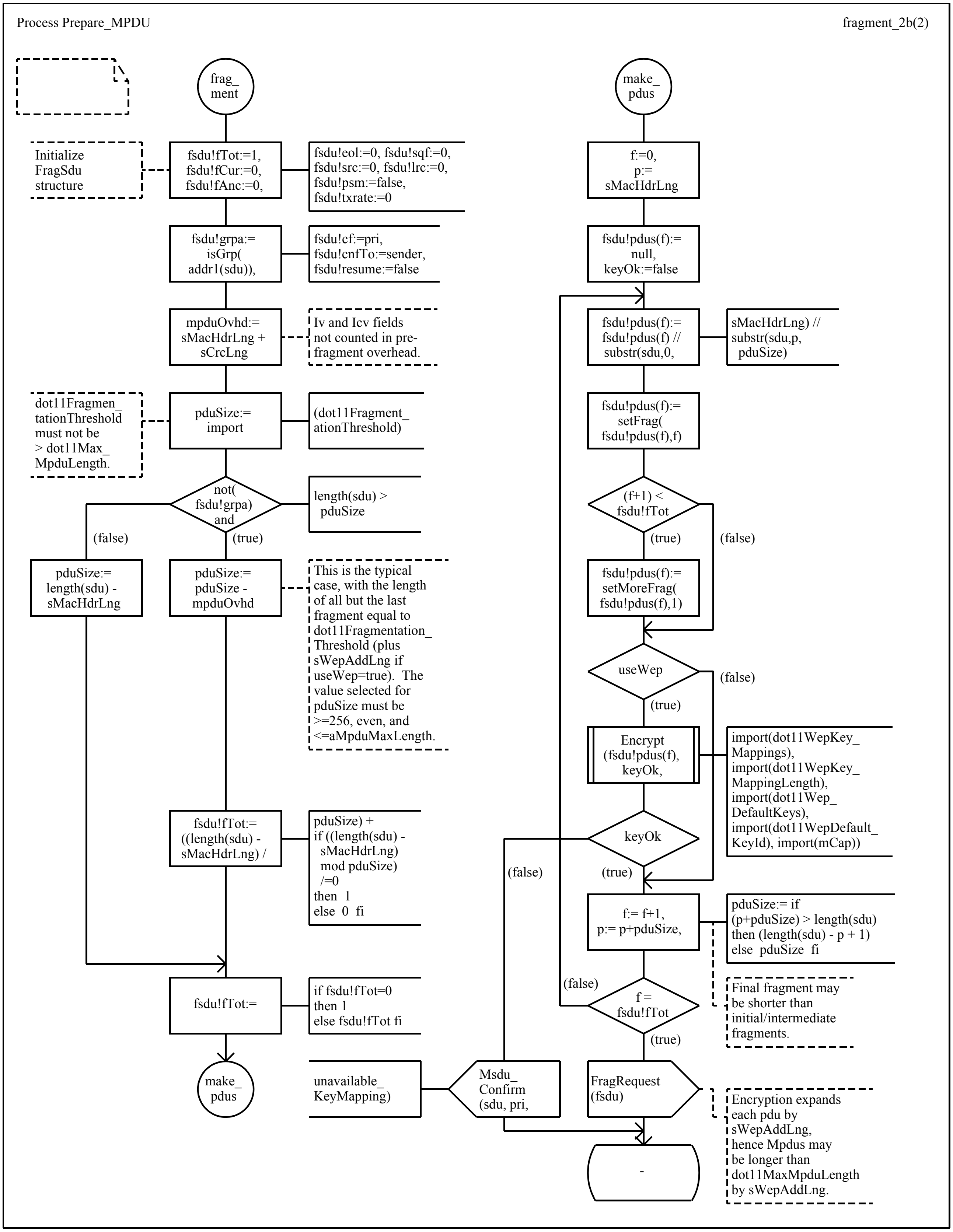

DSM
RSDU
TSDU
Block Distribution_Service
ToDsm
Msdu_ Indicate
Msdu_ Confirm
Dist_Service_1a(1)
DsDsm
DsMd
MdDs
FromDsm
Msdu_ Request
DSM_Interface (1,1)
/* only at AP */
DsBss
FRDS
MsduConfirm
MsduRequest
Msdu_ Indicate
DsInquiry, DsNotify
/* This block interfaces between the AP function and the Distribution System Medium, hence is only present at APs.
In order to permit an LLC entity colocated with an AP to communicate via both the WM and the DSM, MAC_Data_Service at the AP interacts with this block. This causes frames originating at the station containing the AP to be treated equivalently to frames originating at any of the other stations associated with that AP. */
BssDs
MlmeDs
DsResponse
TODS
MMDS

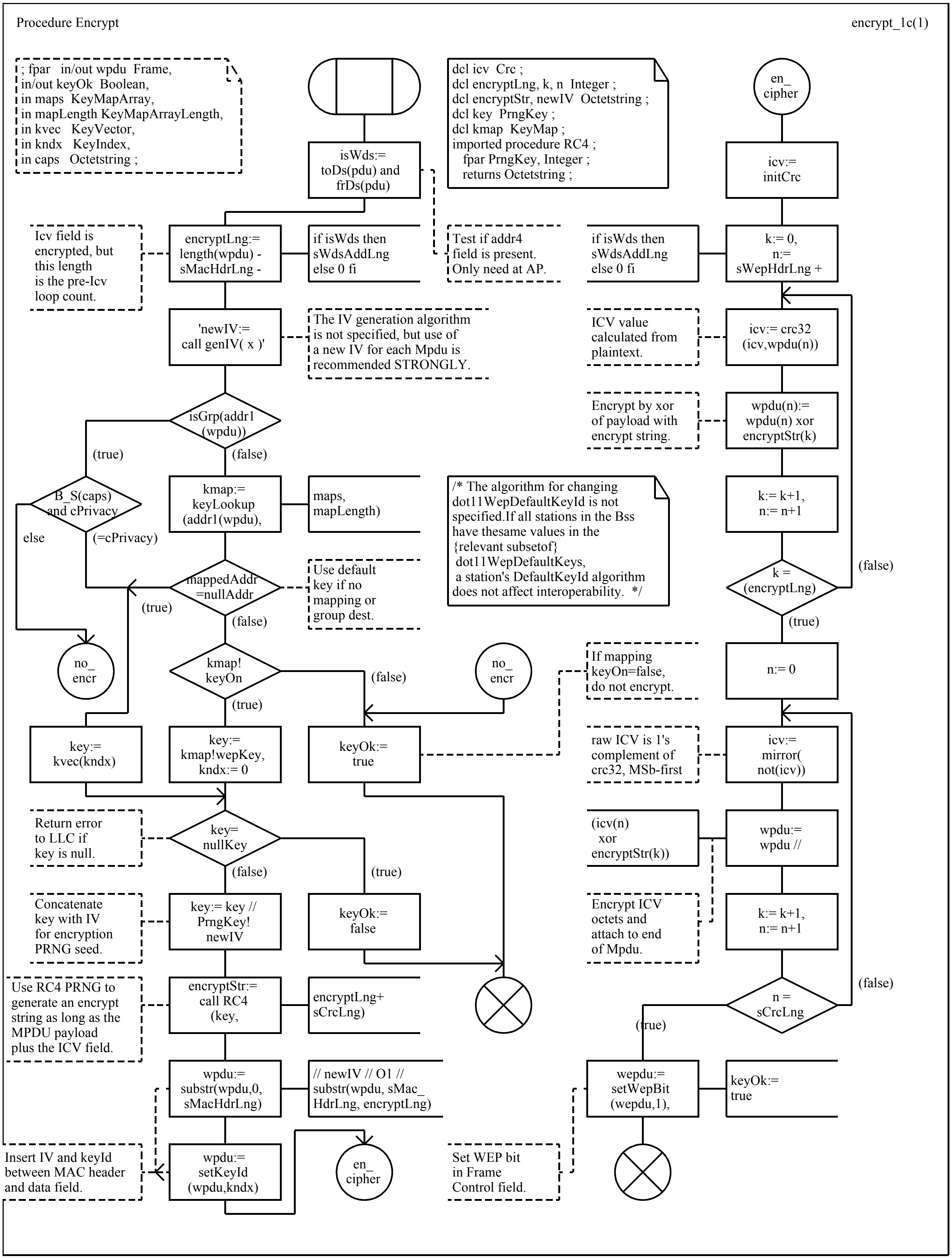

Process DSM_Interface
DSM_data_1a(2)
dcl da, sa, wdsAddr MacAddr ;
dcl dsmData Octetstring ;
dcl dss DsStatus ;
dcl rpri, tpri CfPriority ;
dcl rsdu, tsdu Frame ;
dcl trsl TxResult ;
DS_Idle
State continues on next page.
FromDsm (da, sa, dsmData)
MSDU in from DSM.
MsduIn_ dicate(rsdu, rpri)
MSDU in from WM.
MsduRequest (rsdu,tpri)
MSDU in from local LLC entity
'to Bss ?'
True if da is addr of asoc sta or any group addr.
(false)
(true)
tsdu:= mkFrame (data,
da, sa, DsmData), tsdu:=setFrDs (tsdu,1)
addr3(rsdu), addr2(rsdu), substr(rsdu, sMacHdrLng, length(rsdu) - sMacHdrLng)), tsdu:=setFrDs (tsdu,1)
addr2(rsdu), addr3(rsdu), substr(rsdu, sMacHdrLng, length(rsdu) - sMacHdrLng)), tsdu:=setFrDs (tsdu,1)
MsduRequest (tsdu, contention)
'to local LLC ?'
True if da is addr of this sta or active group addr.
'to Dsm ?'
True if da is any group addr or addr of sta not asoc here.
rsdu:= mkFrame (data,
da, sa, DsmData), rsdu:=setFrDs (rsdu,1)
MsduIn_ dicate(rsdu, contention)
ToDsm (addr1 (rsdu),
addr2(rsdu), substr(rsdu, sMacHdrLng, length(rsdu) - sMacHdrLng))
ToDsm (addr3 (rsdu),
'to Wds ?'
True if da reached via {one or more} AP(wdsAddr).
tsdu:= mkFrame (data,
wdsAddr, da, DsmData), tsdu:= insAddr4(sa), tsdu:=setFrDs (tsdu,1), tsdu:=setToDs (tsdu,1)
wdsAddr, addr3(rsdu), substr(rsdu, sMacHdrLng, length(rsdu) - sMacHdrLng)), tsdu:= insAddr4 (addr2(rsdu)), tsdu:=setFrDs (tsdu,1), tsdu:=setToDs (tsdu,1)
wdsAddr, addr1(rsdu), substr(rsdu, sMacHdrLng, length(rsdu) - sMacHdrLng)), tsdu:= insAddr4 (addr2(rsdu)), tsdu:=setFrDs (tsdu,1), tsdu:=setToDs (tsdu,1)
True if da reached via (one or more} AP(wdsAddr).
True if da reached via {one or more} AP(wdsAddr).

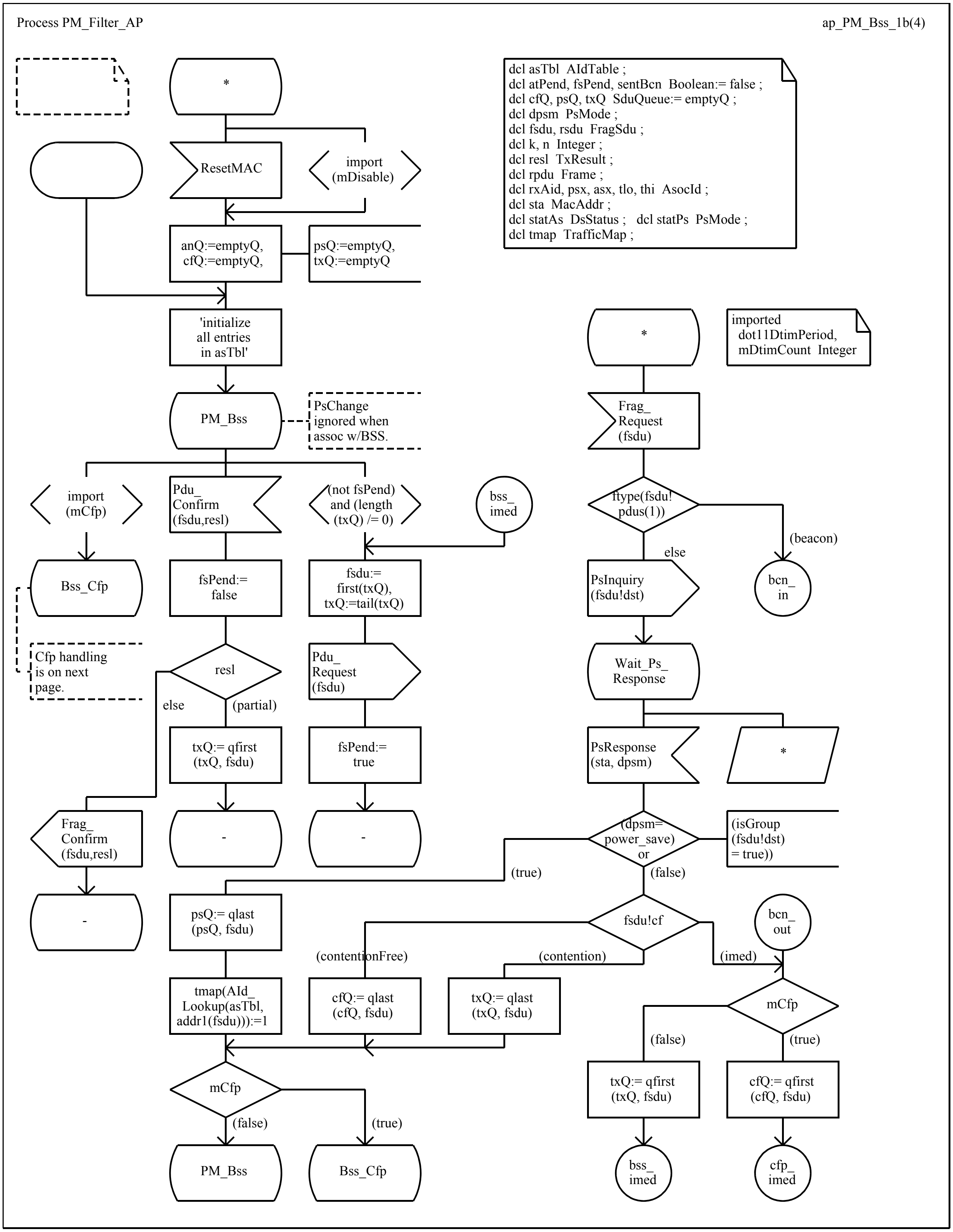

Process DSM_Interface
DSM_management_2a(2)
DS_Idle
State continued from previous page.
MsduCon_ firm(tsdu, tpri,trsl)
Response to prior MsduRequest
addr3 (tsdu) =
import (aStationId)
(false)
(true)
MsduCon_ firm(tsdu, tpri,trsl)
Pass confirm of LLC request to Mac_Data_ Service.
DS_Idle
DsNotify (da, dss)
Notification of asoc, reasoc, or disasoc from Mlme.
'update info about sta[da] with status dss'
DS_Idle
DsInquiry (da, sa)
Inquiry about existing asoc status. Sent by Mlme when validating new asoc or reasoc requests.
'dss:= ds status of station[da]'
'sa:= ap addr where da is asoc'
DsResponse (da, sa, dss)
DS_Idle

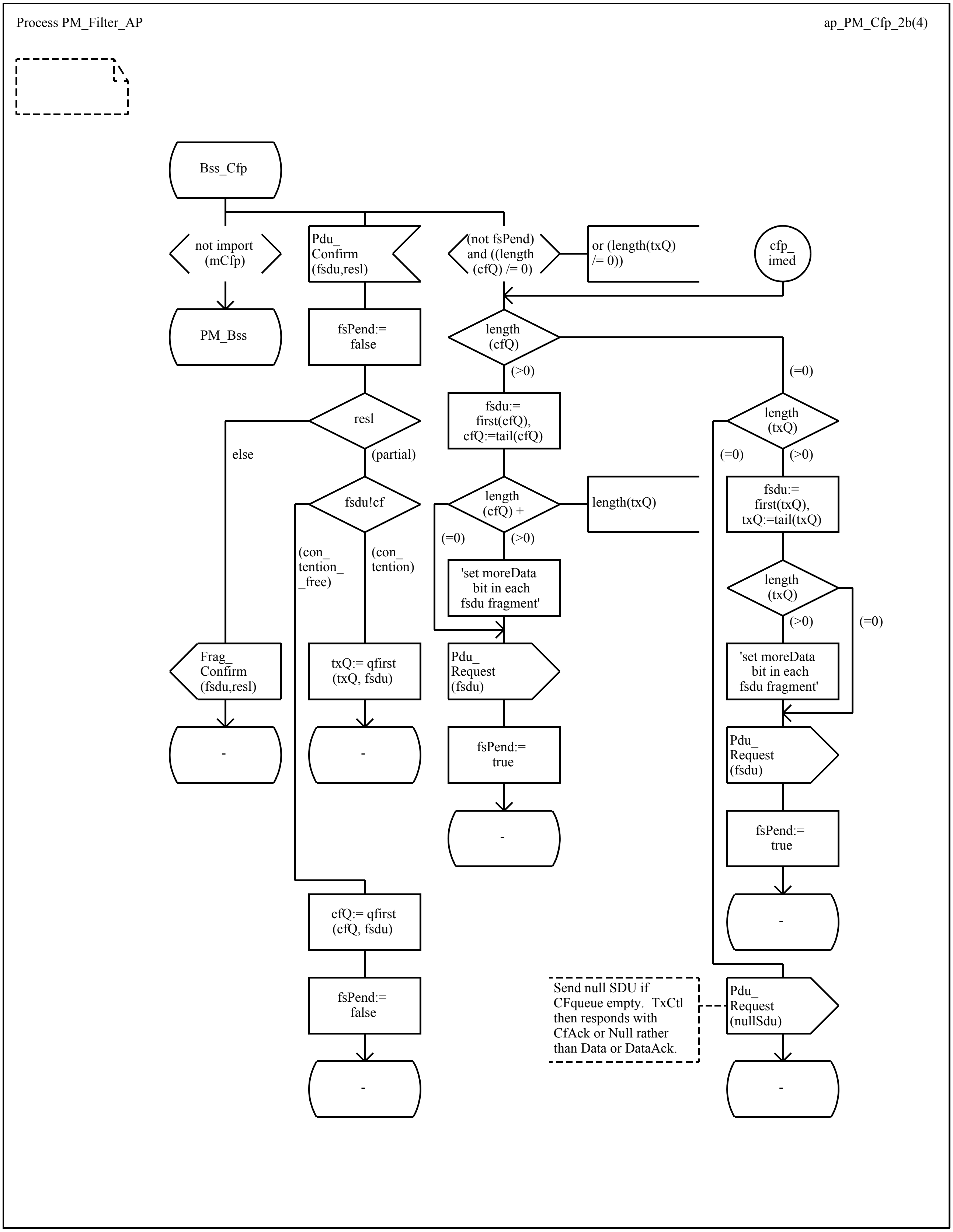

FRDS
Block MPDU_Generation_AP
MsduConfirm
ap_MPDU_gen_1a(1)
Msdu
signal
FragConfirm(FragSdu,TxResult),
FragRequest(FragSdu) ;
MsduRequest
Includes encryption if dot11PrivacyOptionImplemented =true. This is a typical location, but implementers may use other locations between the MAC_SAP and PHY_SAP_TX as long as they provide the specified behavior as observed at LLC, MLME and the WM.
Prepare_MPDU
(1,1)
/* for STA and AP */
MmRequest
Mmpdu
FragConfirm
MmConfirm
MMTX
FragMsdu
/* This block converts outgoing Msdus and Mmpdus into Mpdus, fragmenting and encrypting as necessary.
The PM_Filter_AP process queues frames needing announcement in a TIM, and frames to be sent during the CF period at an AP with an active point coordinator. This process also adds the TIM element to outgoing Beacon frames. */
PsInquiry
FragRequest
PwrMgt
PM_Filter_AP
(1,1)
/* AP version */
AsChange,
PsResponse,
PsChange
PduConfirm,
PsPolled
Mpdu
PduRequest
TPDU

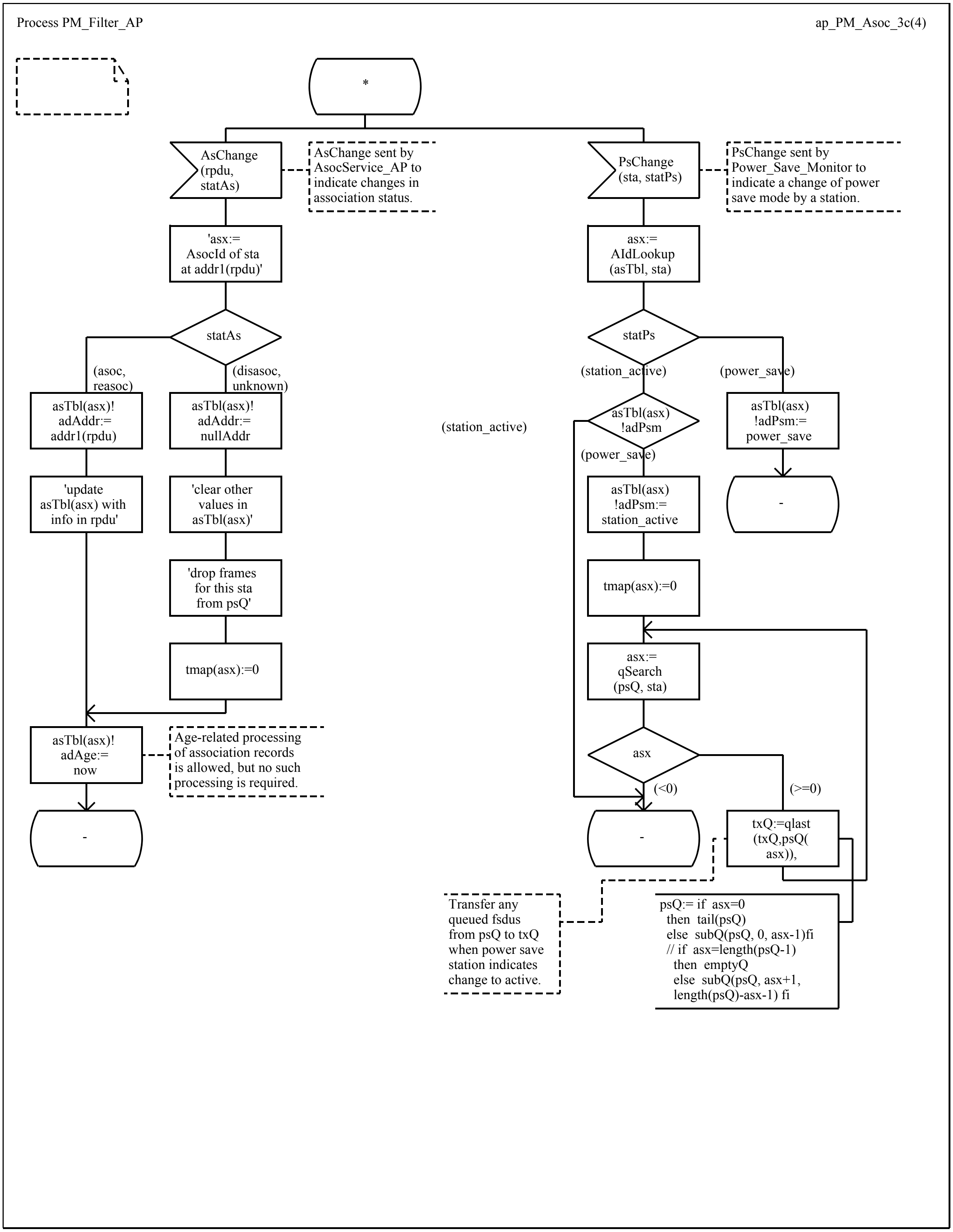

Process Prepare_MPDU
prepare_1b(2)
Encrypt
Procedure used for WEP encryption.
If dot11PrivacyOptionImplemented=
false, this procedure is not present.
dcl bcmc, keyOk,
useWep Boolean:= false ;
dcl f FragNum ;
dcl fsdu FragSdu ;
dcl mpduOvhd, p,
pduSize, thld Integer ;
dcl pri CfPriority ;
dcl rrsl TxResult ;
dcl sdu, rsdu Frame ;
imported mAssoc, mIbss, dot11PrivacyInvoked Boolean ;
imported dot11FragmentationThreshold Integer ;
imported dot11WepDefaultKeys KeyVector ;
imported dot11WepDefaultKeyId KeyIndex ;
imported dot11WepKeyMappings KeyMapArray ;
imported dot11WepKeyMappingLength KeyMapArrayLength ;
imported mCap Octetstring ;
No_Bss
import (mAssoc)
and (not import(mAct_ ingAsAp))
import (mIbss)
import (mActing_ AsAp)
Msdu_ Request (sdu,pri)
Prepare_ _Bss
All data frames in Bss sent to distrib. service
Prepare_ _Ibss
All data frames in Ibss sent to destination sta.
Prepare_ _AP
MsduConfirm (sdu,pri, noBss)
not import (mAssoc)
Msdu_ Request (sdu,pri)
Msdu_ Request (sdu,pri)
not import (mIbss)
No_Bss
No_Bss
No_Bss
Msdu_ Request (sdu,pri)
not import (mActing_ AsAp)
No_Bss
sdu:= setAddr1 (sdu,import (mBssId)), sdu:= setToDs (sdu,1)
sdu:= setAddr3(sdu, addr1(sdu)),
Mmpdus sent even when not in Bss/Ibss.
*
Data frames rejected if no Bss/Ibss. Implementations may retain these frames until a Bss becomes (re)available.
Invoked) and dot11Privacy_ Option_ Implemented
useWep:= import(dot11Privacy_
ResetMAC
Mm_ Request (sdu,pri)
Frag_ Confirm (fsdu,pri,rrsl)
Fragment and encrypt is on next page.
frag_ ment
No_Bss
bcmc:= isGroup(addr1(sdu))
rsdu:= substr (fsdu! pdus(0), 0,
sMacHdrLng), pri:= fsdu!cf
dot11Privacy_ Option_ Implemented and if wepBit(sdu)=1 then true else false fi
useWep:=
basetype
(fsdu! pdus(0))
else
(management)
frag_ ment
Msdu_ Confirm (rsdu,pri,rrsl)
MmConfirm (rsdu,pri,rrsl) to fsdu!cnfTo
/* This process generates one or more Mpdus from each outgoing Msdu or Mmpdu. If encryption is needed, the Mpdus are encrypted before being passed to be filtered for possible power save or CF queuing before tx. */
wepBit=true in request for 3rd frame of shared key auth. seq.
Confirm Msdu to MAC data service, confirm Mmpdu to MLME sub-block.
-

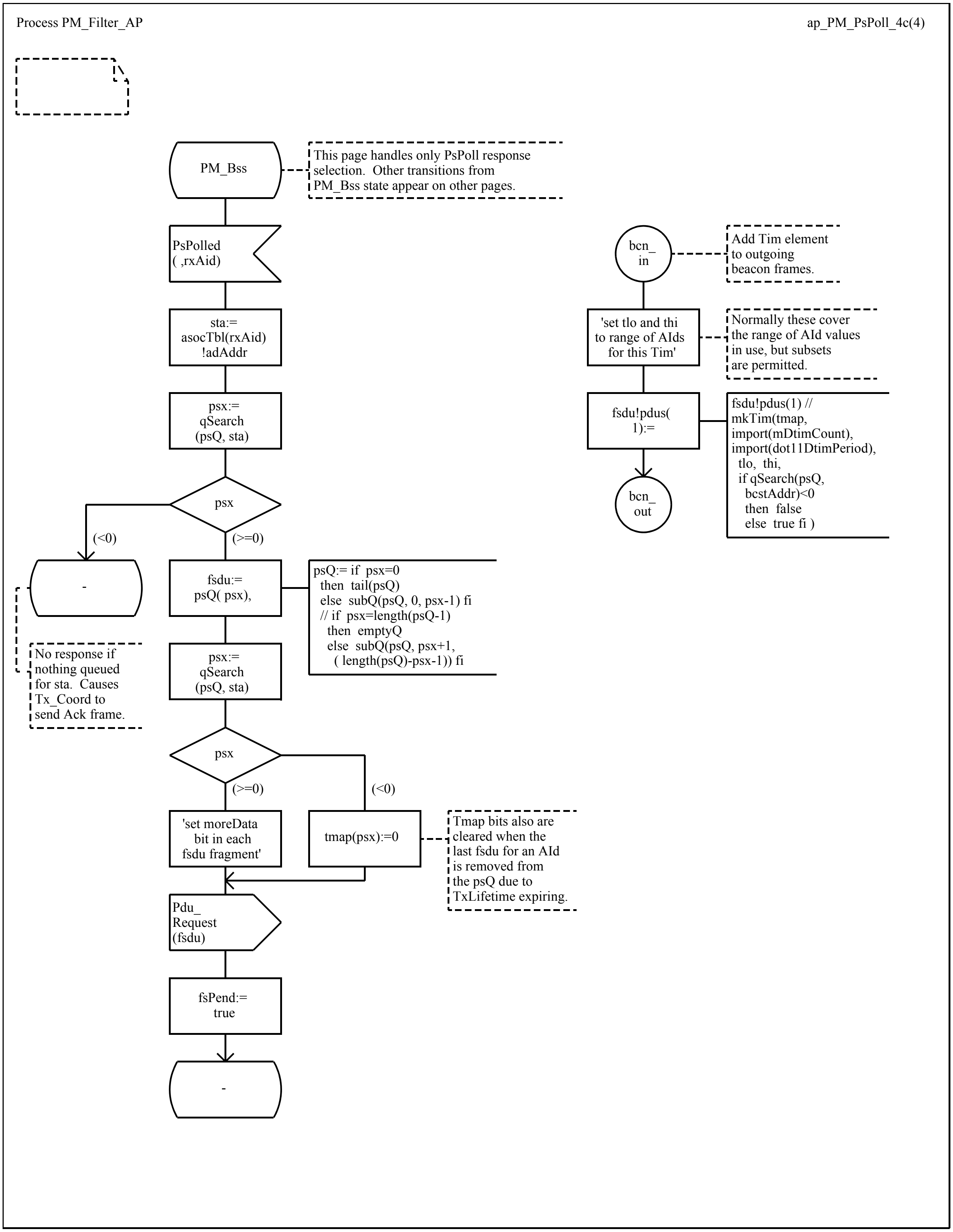

Process Prepare_MPDU
fragment_2b(2)
frag_ ment
Initialize FragSdu structure
fsdu!fTot:=1, fsdu!fCur:=0, fsdu!fAnc:=0,
fsdu!eol:=0, fsdu!sqf:=0, fsdu!src:=0, fsdu!lrc:=0, fsdu!psm:=false, fsdu!txrate:=0
fsdu!grpa:= isGrp(addr1(sdu)),
fsdu!cf:=pri, fsdu!cnfTo:=sender, fsdu!resume:=false
mpduOvhd:= sMacHdrLng + sCrcLng
Iv and Icv fields not counted in pre-fragment overhead.
dot11Fragmen_ tationThreshold must not be > dot11Max_ MpduLength.
pduSize:= import
(dot11Fragment_ ationThreshold)
not(fsdu!grpa) and
length(sdu) > pduSize
(false)
(true)
pduSize:= length(sdu) - sMacHdrLng
pduSize:= pduSize - mpduOvhd
This is the typical case, with the length of all but the last fragment equal to dot11Fragmentation_ Threshold (plus sWepAddLng if useWep=true). The value selected for pduSize must be >=256, even, and <=aMpduMaxLength.
fsdu!fTot:= ((length(sdu) - sMacHdrLng) /
pduSize) + if ((length(sdu) - sMacHdrLng) mod pduSize) /=0 then 1 else 0 fi
fsdu!fTot:=
if fsdu!fTot=0 then 1 else fsdu!fTot fi
make_ pdus
make_ pdus
f:=0, p:= sMacHdrLng
fsdu!pdus(f):= null, keyOk:=false
fsdu!pdus(f):= fsdu!pdus(f) // substr(sdu,0,
sMacHdrLng) // substr(sdu,p, pduSize)
fsdu!pdus(f):= setFrag(fsdu!pdus(f),f)
(f+1) < fsdu!fTot
(true)
(false)
fsdu!pdus(f):= setMoreFrag(fsdu!pdus(f),1)
useWep
(true)
(false)
Encrypt (fsdu!pdus(f), keyOk,
import(dot11WepKey_ Mappings), import(dot11WepKey_ MappingLength), import(dot11Wep_ DefaultKeys), import(dot11WepDefault_ KeyId), import(mCap))
keyOk
(false)
(true)
f:= f+1, p:= p+pduSize,
pduSize:= if (p+pduSize) > length(sdu) then (length(sdu) - p + 1) else pduSize fi
Final fragment may be shorter than initial/intermediate fragments.
(false)
f = fsdu!fTot
(true)
unavailable_ KeyMapping)
Msdu_ Confirm (sdu, pri,
FragRequest (fsdu)
Encryption expands each pdu by sWepAddLng, hence Mpdus may be longer than dot11MaxMpduLength by sWepAddLng.
-

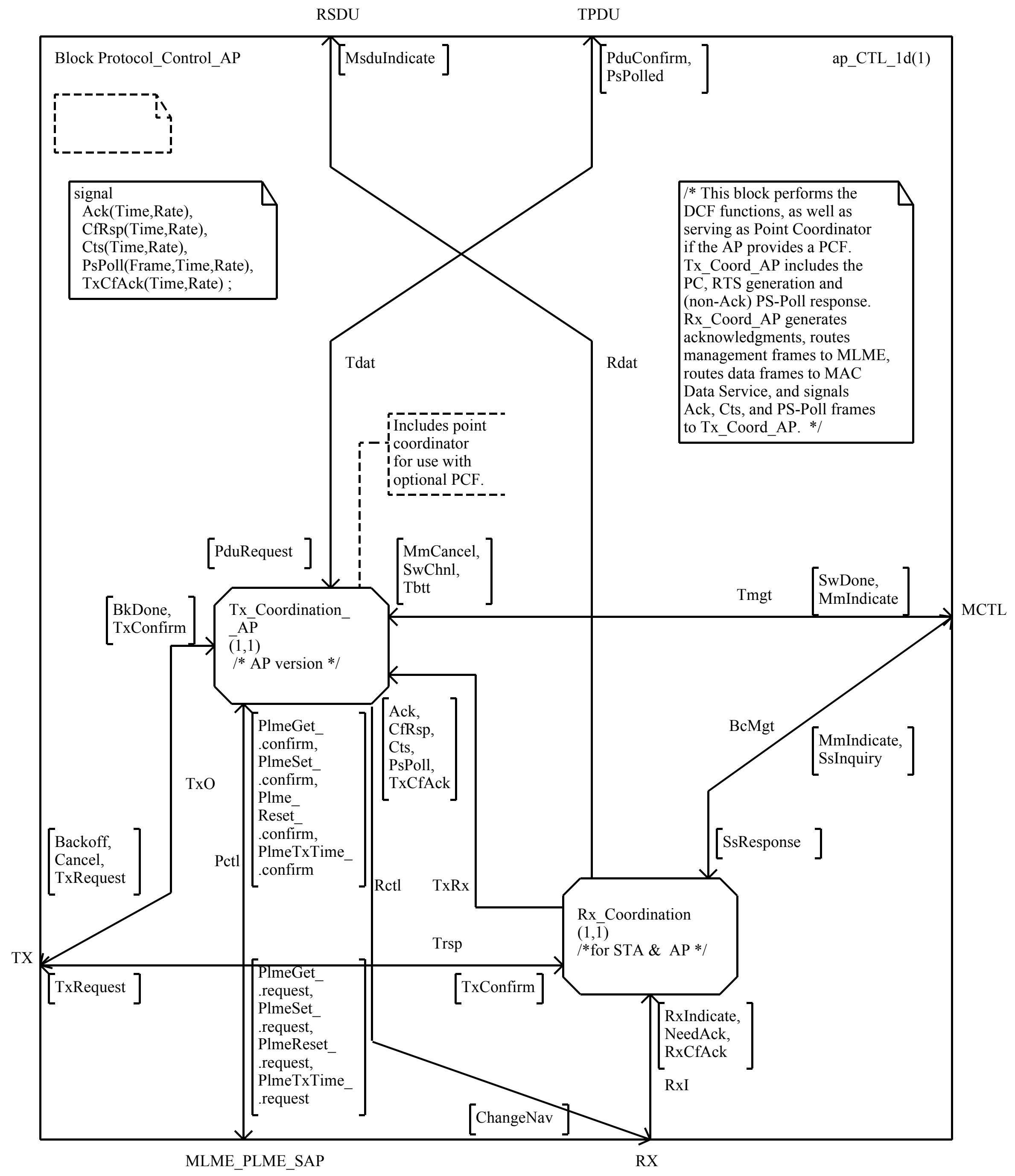

Procedure Encrypt
encrypt_1c(1)
; fpar in/out wpdu Frame,
in/out keyOk Boolean,
in maps KeyMapArray,
in mapLength KeyMapArrayLength,
in kvec KeyVector,
in kndx KeyIndex,
in caps Octetstring ;
dcl icv Crc ;
dcl encryptLng, k, n Integer ;
dcl encryptStr, newIV Octetstring ;
dcl key PrngKey ;
dcl kmap KeyMap ;
imported procedure RC4 ;
fpar PrngKey, Integer ;
returns Octetstring ;
isWds:= toDs(pdu) and frDs(pdu)
Icv field is encrypted, but this length is the pre-Icv loop count.
encryptLng:= length(wpdu) - sMacHdrLng -
if isWds then sWdsAddLng else 0 fi
Test if addr4 field is present. Only need at AP.
'newIV:= call genIV(x)'
The IV generation algorithm is not specified, but use of a new IV for each Mpdu is recommended STRONGLY.
isGrp(addr1 (wpdu))
(true)
(false)
B_S(caps) and cPrivacy
else
(=cPrivacy)
kmap:= keyLookup (addr1(wpdu),
maps, mapLength)
/* The algorithm for changing dot11WepDefaultKeyId is not specified.If all stations in the Bss have thesame values in the {relevant subsetof} dot11WepDefaultKeys, a station's DefaultKeyId algorithm does not affect interoperability. */
mappedAddr =nullAddr
(true)
(false)
Use default key if no mapping or group dest.
no_ encr
kmap! keyOn
(false)
(true)
no_ encr
key:= kvec(kndx)
key:= kmap!wepKey, kndx:= 0
keyOk:= true
Return error to LLC if key is null.
key= nullKey
(false)
(true)
Concatenate key with IV for encryption PRNG seed.
key:= key // PrngKey! newIV
keyOk:= false
Use RC4 PRNG to generate an encrypt string as long as the MPDU payload plus the ICV field.
encryptStr:= call RC4 (key,
encryptLng+ sCrcLng)
wpdu:= substr(wpdu,0, sMacHdrLng)
// newIV // O1 // substr(wpdu, sMac_ HdrLng, encryptLng)
Insert IV and keyId between MAC header and data field.
wpdu:= setKeyId (wpdu,kndx)
en_ cipher
en_ cipher
icv:= initCrc
if isWds then sWdsAddLng else 0 fi
k:= 0, n:= sWepHdrLng +
ICV value calculated from plaintext.
icv:= crc32 (icv,wpdu(n))
Encrypt by xor of payload with encrypt string.
wpdu(n):= wpdu(n) xor encryptStr(k)
k:= k+1, n:= n+1
k = (encryptLng)
(false)
(true)
n:= 0
If mapping keyOn=false, do not encrypt.
raw ICV is 1's complement of crc32, MSb-first
icv:= mirror(not(icv))
(icv(n) xor encryptStr(k))
wpdu:= wpdu //
Encrypt ICV octets and attach to end of Mpdu.
k:= k+1, n:= n+1
n = sCrcLng
(false)
(true)
wepdu:= setWepBit (wepdu,1),
keyOk:= true
Set WEP bit in Frame Control field.


Process PM_Filter_AP
ap_PM_Bss_1b(4)
dcl asTbl AIdTable ;
dcl atPend, fsPend, sentBcn Boolean:= false ;
dcl cfQ, psQ, txQ SduQueue:= emptyQ ;
dcl dpsm PsMode ;
dcl fsdu, rsdu FragSdu ;
dcl k, n Integer ;
dcl resl TxResult ;
dcl rpdu Frame ;
dcl rxAid, psx, asx, tlo, thi AsocId ;
dcl sta MacAddr ;
dcl statAs DsStatus ; dcl statPs PsMode ;
dcl tmap TrafficMap ;
*
ResetMAC
import (mDisable)
anQ:=emptyQ, cfQ:=emptyQ,
psQ:=emptyQ, txQ:=emptyQ
'initialize all entries in asTbl'
PM_Bss
PsChange ignored when assoc w/BSS.
import (mCfp)
Pdu_ Confirm (fsdu,resl)
(not fsPend) and (length (txQ) /= 0)
bss_ imed
Bss_Cfp
fsPend:= false
fsdu:= first(txQ), txQ:=tail(txQ)
Cfp handling is on next page.
resl
else
(partial)
Pdu_ Request (fsdu)
txQ:= qfirst (txQ, fsdu)
fsPend:= true
Frag_ Confirm (fsdu,resl)
-
-
-
psQ:= qlast (psQ, fsdu)
tmap(AId_ Lookup(asTbl, addr1(fsdu))):=1
mCfp
(false)
(true)
PM_Bss
Bss_Cfp
*
imported dot11DtimPeriod, mDtimCount Integer
Frag_ Request (fsdu)
ftype(fsdu! pdus(1))
(beacon)
else
bcn_ in
PsInquiry (fsdu!dst)
Wait_Ps_ Response
PsResponse (sta, dpsm)
*
(dpsm= power_save) or
(isGroup (fsdu!dst) = true))
(true)
(false)
fsdu!cf
(contentionFree)
(contention)
(imed)
bcn_ out
cfQ:= qlast (cfQ, fsdu)
txQ:= qlast (txQ, fsdu)
mCfp
(false)
(true)
txQ:= qfirst (txQ, fsdu)
cfQ:= qfirst (cfQ, fsdu)
bss_ imed
cfp_ imed


Process PM_Filter_AP
ap_PM_Cfp_2b(4)
Bss_Cfp
not import (mCfp)
PM_Bss
Pdu_ Confirm (fsdu,resl)
fsPend:= false
resl
else
(partial)
fsdu!cf
(con_ tention_ _free)
(con_ tention)
Frag_ Confirm (fsdu,resl)
-
txQ:= qfirst (txQ, fsdu)
-
cfQ:= qfirst (cfQ, fsdu)
fsPend:= false
-
(not fsPend) and ((length (cfQ) /= 0)
or (length(txQ) /= 0))
cfp_ imed
length (cfQ)
(>0)
(=0)
fsdu:= first(cfQ), cfQ:=tail(cfQ)
length (cfQ) +
length(txQ)
(=0)
(>0)
'set moreData bit in each fsdu fragment'
Pdu_ Request (fsdu)
fsPend:= true
-
length (txQ)
(=0)
(>0)
fsdu:= first(txQ), txQ:=tail(txQ)
length (txQ)
(>0)
(=0)
'set moreData bit in each fsdu fragment'
Pdu_ Request (fsdu)
fsPend:= true
-
Send null SDU if CFqueue empty. TxCtl then responds with CfAck or Null rather than Data or DataAck.
Pdu_ Request (nullSdu)
-

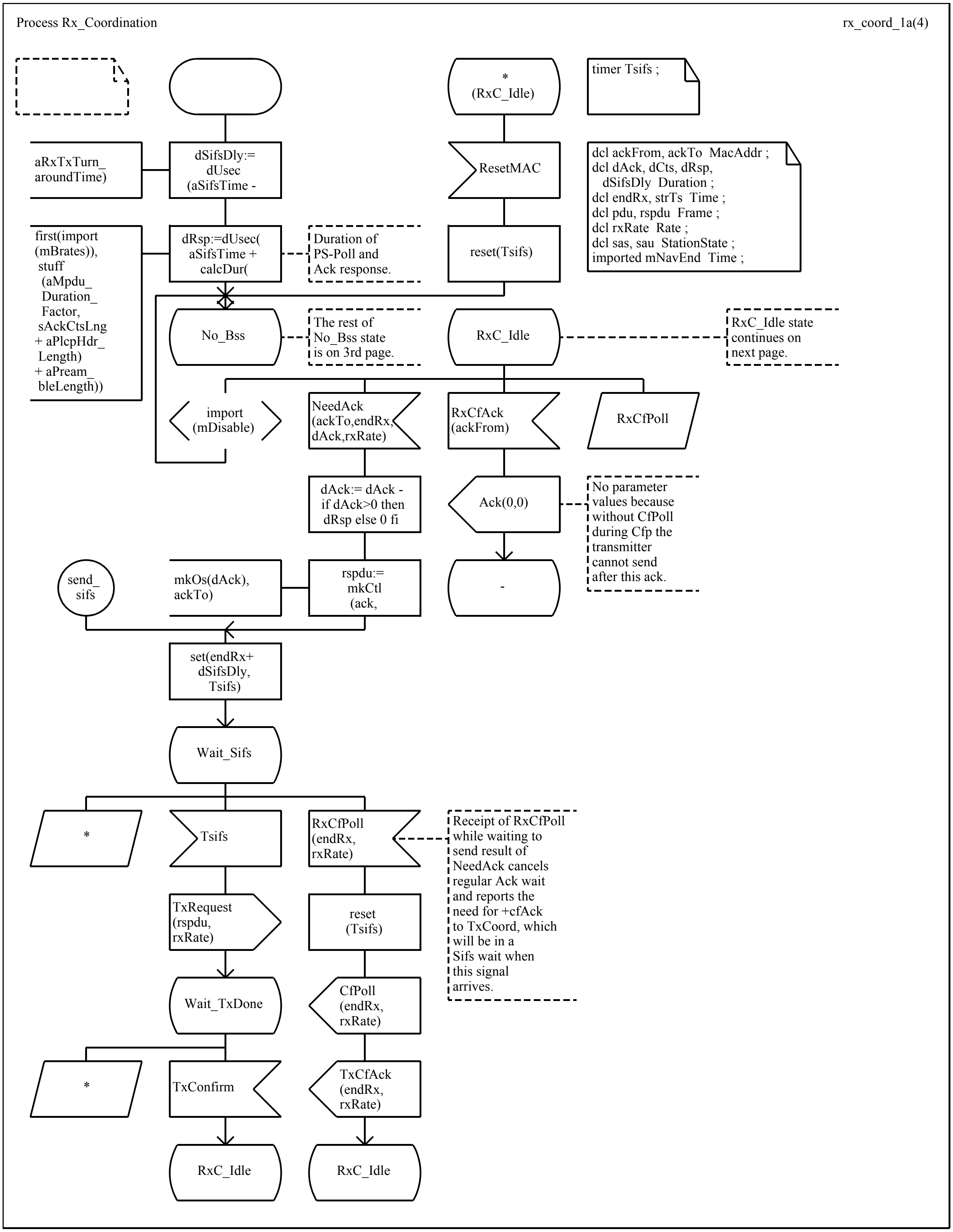

Process PM_Filter_AP
ap_PM_Asoc_3c(4)
*
AsChange (rpdu, statAs)
AsChange sent by AsocService_AP to indicate changes in association status.
PsChange (sta, statPs)
PsChange sent by Power_Save_Monitor to indicate a change of power save mode by a station.
'asx:= AsocId of sta at addr1(rpdu)'
asx:= AIdLookup (asTbl, sta)
statAs
statPs
(asoc, reasoc)
(disasoc, unknown)
(station_active)
(power_save)
asTbl(asx)! adAddr:= addr1(rpdu)
asTbl(asx)! adAddr:= nullAddr
(station_active)
asTbl(asx) !adPsm
asTbl(asx) !adPsm:= power_save
(power_save)
'update asTbl(asx) with info in rpdu'
'clear other values in asTbl(asx)'
asTbl(asx) !adPsm:= station_active
-
'drop frames for this sta from psQ'
tmap(asx):=0
tmap(asx):=0
asx:= qSearch (psQ, sta)
asTbl(asx)! adAge:= now
Age-related processing of association records is allowed, but no such processing is required.
asx
(<0)
(>=0)
-
-
txQ:=qlast (txQ,psQ(asx)),
Transfer any queued fsdus from psQ to txQ when power save station indicates change to active.
psQ:= if asx=0
then tail(psQ)
else subQ(psQ, 0, asx-1)fi
// if asx=length(psQ-1)
then emptyQ
else subQ(psQ, asx+1,
length(psQ)-asx-1) fi

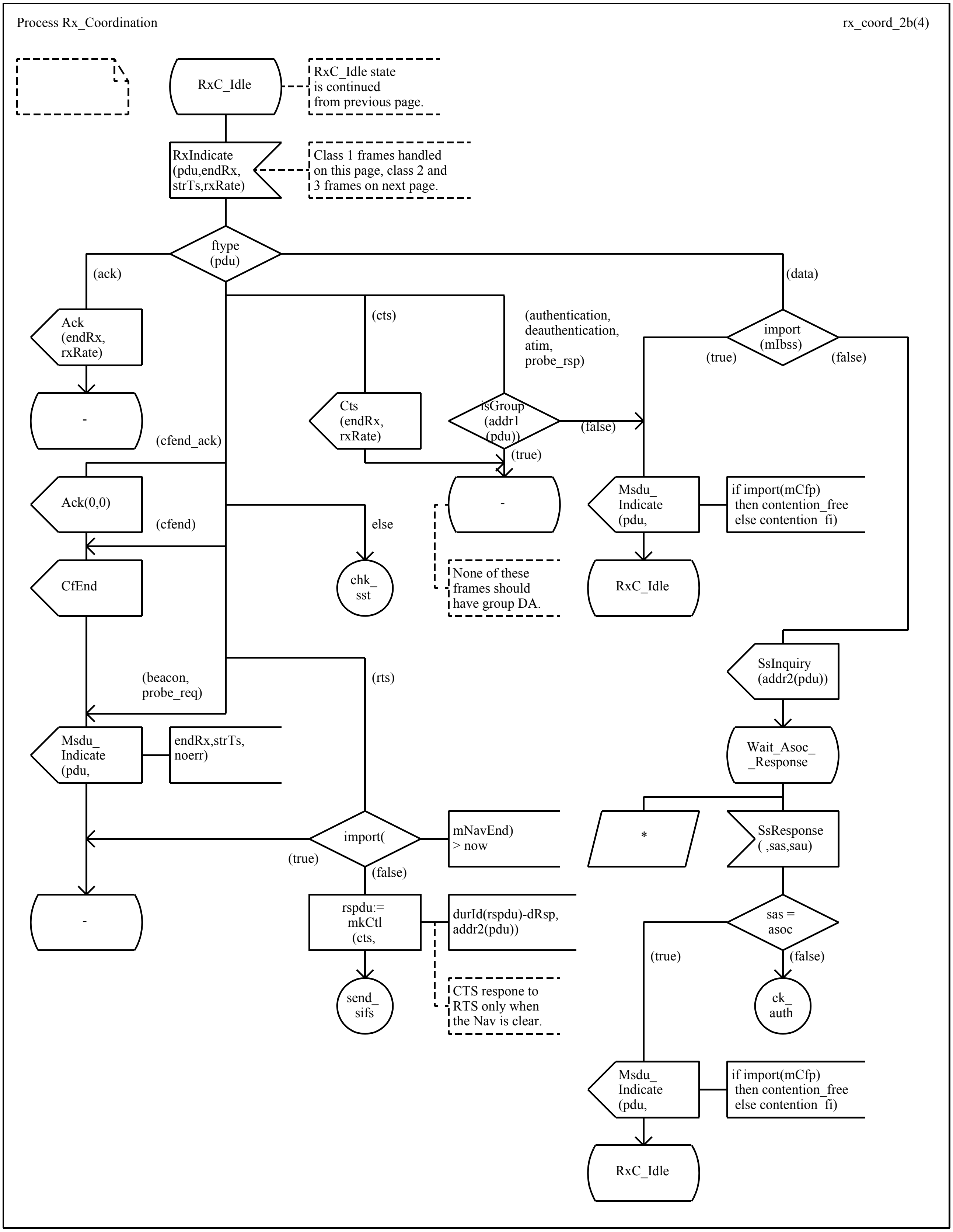

Process PM_Filter_AP
ap_PM_PsPoll_4c(4)
PM_Bss
This page handles only PsPoll response selection. Other transitions from PM_Bss state appear on other pages.
PsPolled (,rxAid)
sta:= asocTbl(rxAid) !adAddr
psx:= qSearch (psQ, sta)
psx
(<0)
(>=0)
-
No response if nothing queued for sta. Causes Tx_Coord to send Ack frame.
fsdu:= psQ(psx),
psQ:= if psx=0
  then tail(psQ)
  else subQ(psQ, 0, psx-1) fi
  // if psx=length(psQ-1)
    then emptyQ
    else subQ(psQ, psx+1,
      (length(psQ)-psx-1)) fi
psx:= qSearch (psQ, sta)
psx
(>=0)
(<0)
'set moreData bit in each fsdu fragment'
tmap(psx):=0
Tmap bits also are cleared when the last fsdu for an AId is removed from the psQ due to TxLifetime expiring.
Pdu_ Request (fsdu)
fsPend:= true
-
bcn_ in
Add Tim element to outgoing beacon frames.
'set tlo and thi to range of AIds for this Tim'
Normally these cover the range of AId values in use, but subsets are permitted.
fsdu!pdus(1):=
fsdu!pdus(1) //
mkTim(tmap,
import(mDtimCount),
import(dot11DtimPeriod),
  tlo, thi,
  if qSearch(psQ,
    bcstAddr)<0
    then false
    else true fi)
bcn_ out

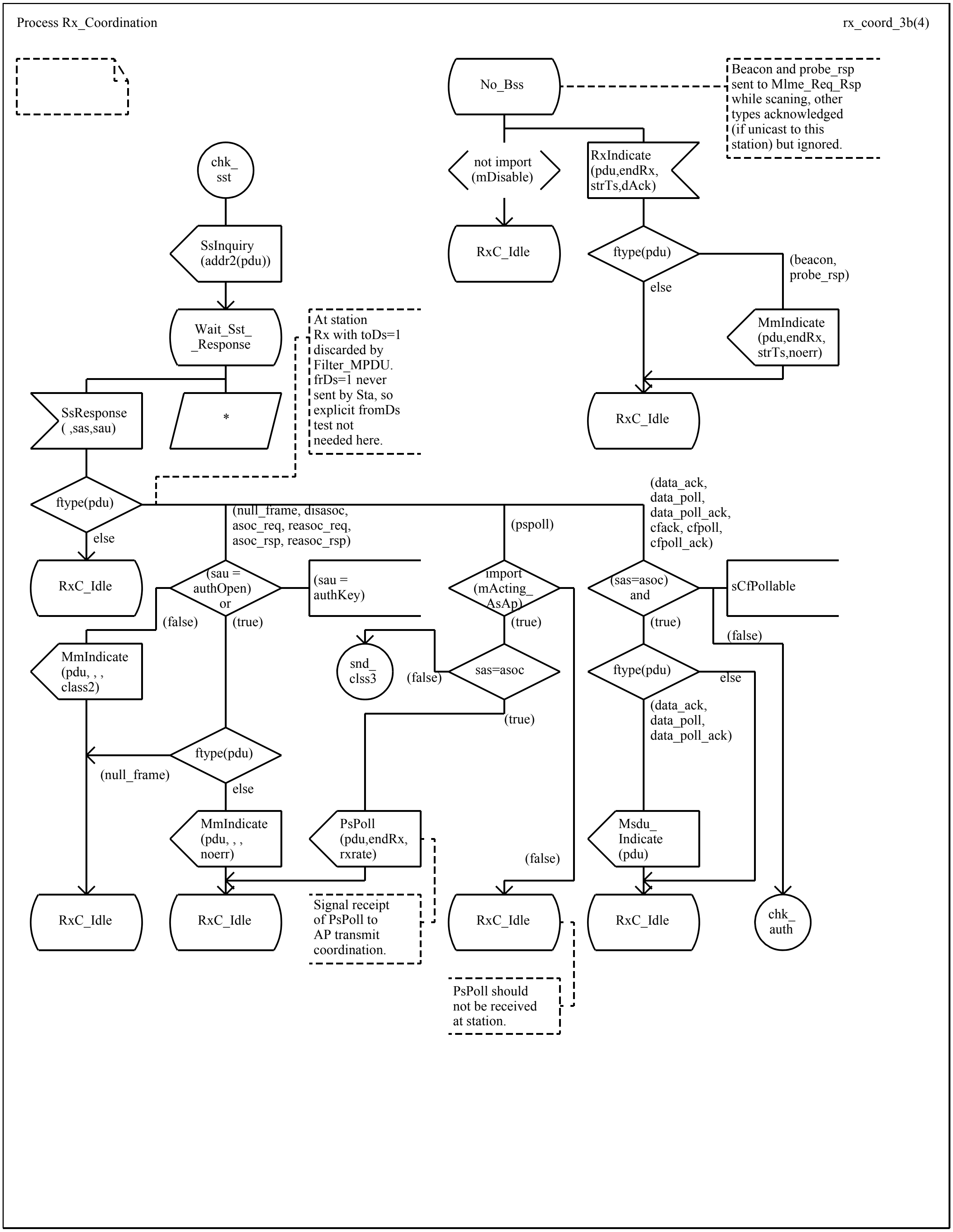


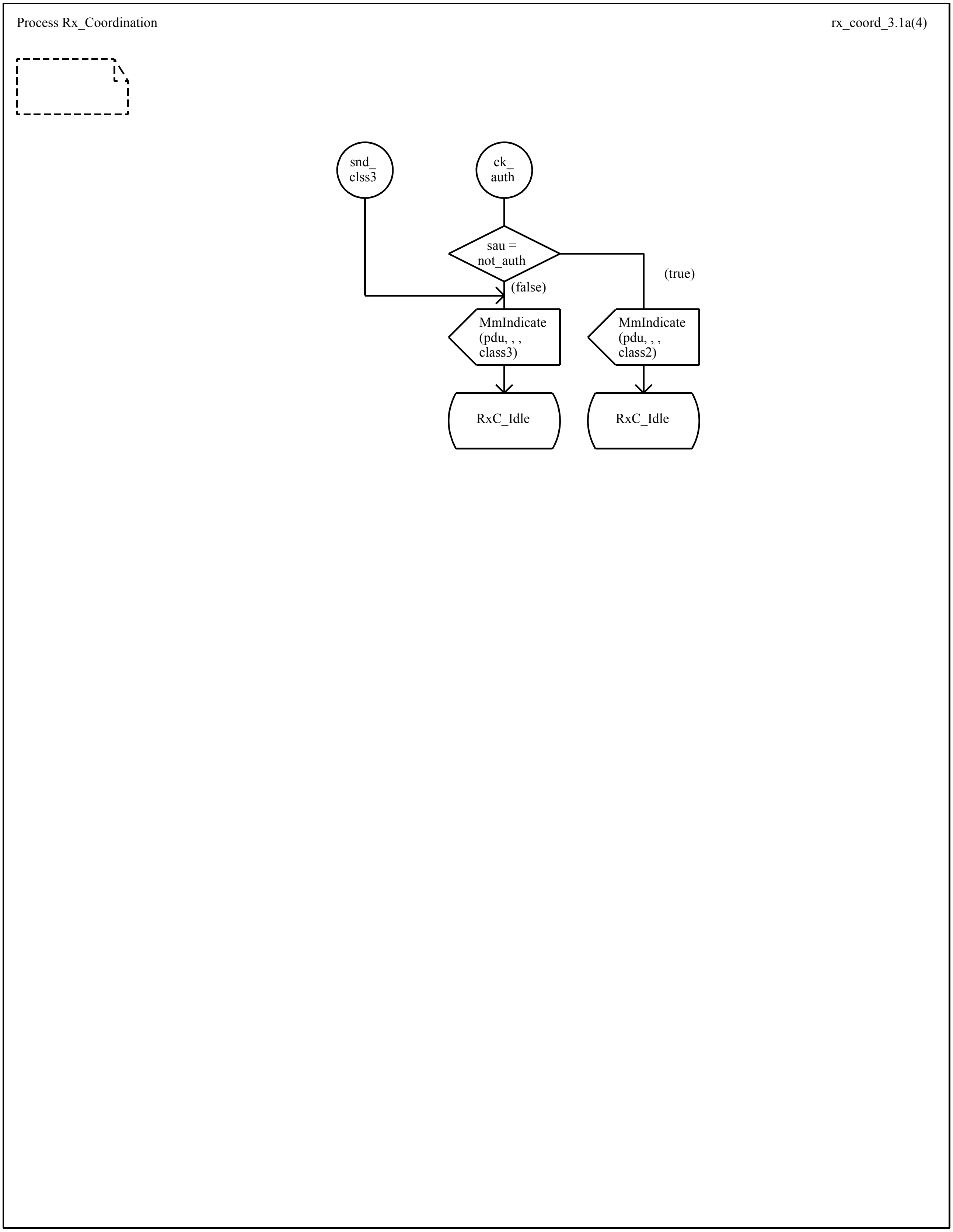
signal
  Ack(Time,Rate),
  CfRsp(Time,Rate),
  Cts(Time,Rate),
  PsPoll(Frame,Time,Rate),
  TxCfAck(Time,Rate) ;

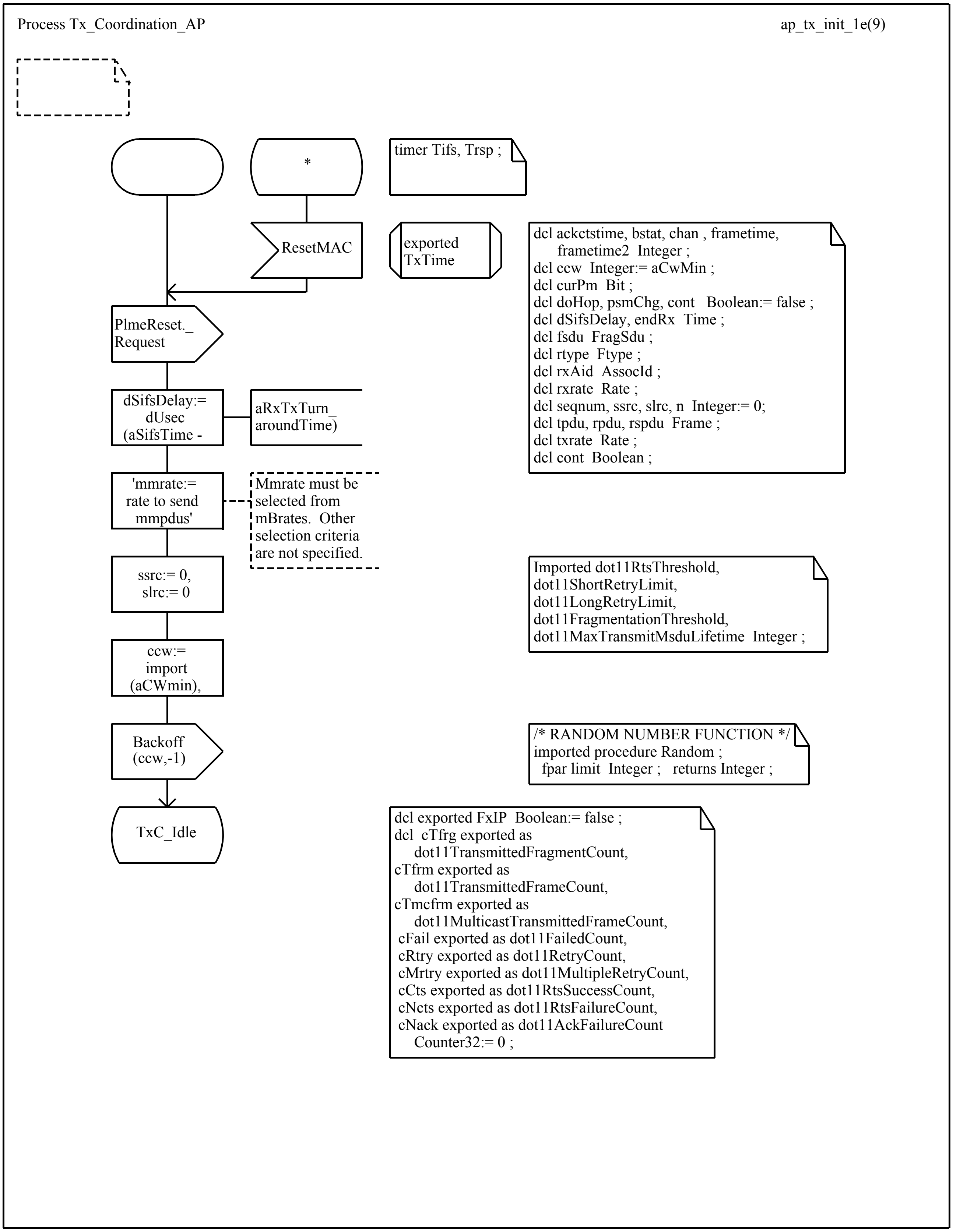
/* This block performs the DCF functions, as well as serving as Point Coordinator if the AP provides a PCF. Tx_Coord_AP includes the PC, RTS generation and (non-Ack) PS-Poll response. Rx_Coord_AP generates acknowledgments, routes management frames to MLME, routes data frames to MAC Data Service, and signals Ack, Cts, and PS-Poll frames to Tx_Coord_AP. */


Process Rx_Coordination
rx_coord_1a(4)
*
(RxC_Idle)
timer Tsifs ;
aRxTxTurn_
aroundTime)
dSifsDly:=
dUsec
(aSifsTime -
ResetMAC
dcl ackFrom, ackTo MacAddr ;
dcl dAck, dCts, dRsp,
dSifsDly Duration ;
dcl endRx, strTs Time ;
dcl pdu, rspdu Frame ;
dcl rxRate Rate ;
dcl sas, sau StationState ;
imported mNavEnd Time ;
first(import
(mBrates)),
stuff
(aMpdu_
Duration_
Factor,
sAckCtsLng
+ aPlcpHdr_
Length)
+ aPream_
bleLength))
dRsp:=dUsec(
aSifsTime +
calcDur(
Duration of
PS-Poll and
Ack response.
reset(Tsifs)
No_Bss
The rest of
No_Bss state
is on 3rd page.
RxC_Idle
RxC_Idle state
continues on
next page.
import
(mDisable)
NeedAck
(ackTo,endRx,
dAck,rxRate)
RxCfAck
(ackFrom)
RxCfPoll
dAck:= dAck -
if dAck>0 then
dRsp else 0 fi
Ack(0,0)
No parameter
values because
without CfPoll
during Cfp the
transmitter
cannot send
after this ack.
send_
sifs
mkOs(dAck),
ackTo)
rspdu:=
mkCtl
(ack,
-
set(endRx+
dSifsDly,
Tsifs)
Wait_Sifs
*
Tsifs
RxCfPoll
(endRx,
rxRate)
Receipt of RxCfPoll
while waiting to
send result of
NeedAck cancels
regular Ack wait
and reports the
need for +cfAck
to TxCoord, which
will be in a
Sifs wait when
this signal
arrives.
TxRequest
(rspdu,
rxRate)
reset
(Tsifs)
Wait_TxDone
CfPoll
(endRx,
rxRate)
*
TxConfirm
TxCfAck
(endRx,
rxRate)
RxC_Idle
RxC_Idle


Process Rx_Coordination
rx_coord_2b(4)
RxC_Idle
RxC_Idle state is continued from previous page.
RxIndicate (pdu,endRx, strTs,rxRate)
Class 1 frames handled on this page, class 2 and 3 frames on next page.
ftype (pdu)
(ack)
(data)
Ack (endRx, rxRate)
-
(cts)
(authentication, deauthentication, atim, probe_rsp)
import (mIbss)
(true)
(false)
Cts (endRx, rxRate)
isGroup (addr1 (pdu))
(false)
(true)
(cfend_ack)
Ack(0,0)
-
Msdu_ Indicate (pdu,
if import(mCfp) then contention_free else contention fi)
(cfend)
else
CfEnd
chk_ sst
None of these frames should have group DA.
RxC_Idle
SsInquiry (addr2(pdu))
(beacon, probe_req)
(rts)
Msdu_ Indicate (pdu,
endRx,strTs, noerr)
Wait_Asoc_ _Response
import(
mNavEnd) > now
*
SsResponse (,sas,sau)
(true)
(false)
-
rspdu:= mkCtl (cts,
durId(rspdu)-dRsp, addr2(pdu))
sas = asoc
(true)
(false)
send_ sifs
CTS respone to RTS only when the Nav is clear.
ck_ auth
Msdu_ Indicate (pdu,
if import(mCfp) then contention_free else contention fi)
RxC_Idle


Process Rx_Coordination
rx_coord_3b(4)
No_Bss
Beacon and probe_rsp sent to Mlme_Req_Rsp while scaning, other types acknowledged (if unicast to this station) but ignored.
not import (mDisable)
RxIndicate (pdu,endRx, strTs,dAck)
RxC_Idle
ftype(pdu)
(beacon, probe_rsp)
else
MmIndicate (pdu,endRx, strTs,noerr)
RxC_Idle
chk_ sst
SsInquiry (addr2(pdu))
Wait_Sst_ _Response
At station Rx with toDs=1 discarded by Filter_MPDU. frDs=1 never sent by Sta, so explicit fromDs test not needed here.
SsResponse (,sas,sau)
*
ftype(pdu)
else
RxC_Idle
(null_frame, disasoc, asoc_req, reasoc_req, asoc_rsp, reasoc_rsp)
(pspoll)
(data_ack, data_poll, data_poll_ack, cfack, cfpoll, cfpoll_ack)
(sau = authOpen) or
(sau = authKey)
(false)
(true)
MmIndicate (pdu, , , class2)
import (mActing_ AsAp)
(true)
snd_ clss3
(false)
sas=asoc
(true)
(sas=asoc) and
sCfPollable
(true)
(false)
ftype(pdu)
else
(data_ack, data_poll, data_poll_ack)
ftype(pdu)
(null_frame)
else
MmIndicate (pdu, , , noerr)
PsPoll (pdu,endRx, rxrate)
(false)
Msdu_ Indicate (pdu)
RxC_Idle
RxC_Idle
Signal receipt of PsPoll to AP transmit coordination.
RxC_Idle
RxC_Idle
chk_ auth
PsPoll should not be received at station.


Process Rx_Coordination
rx_coord_3.1a(4)
snd_
clss3
ck_
auth
sau =
not_auth
(false)
(true)
MmIndicate
(pdu, , ,
class3)
MmIndicate
(pdu, , ,
class2)
RxC_Idle
RxC_Idle



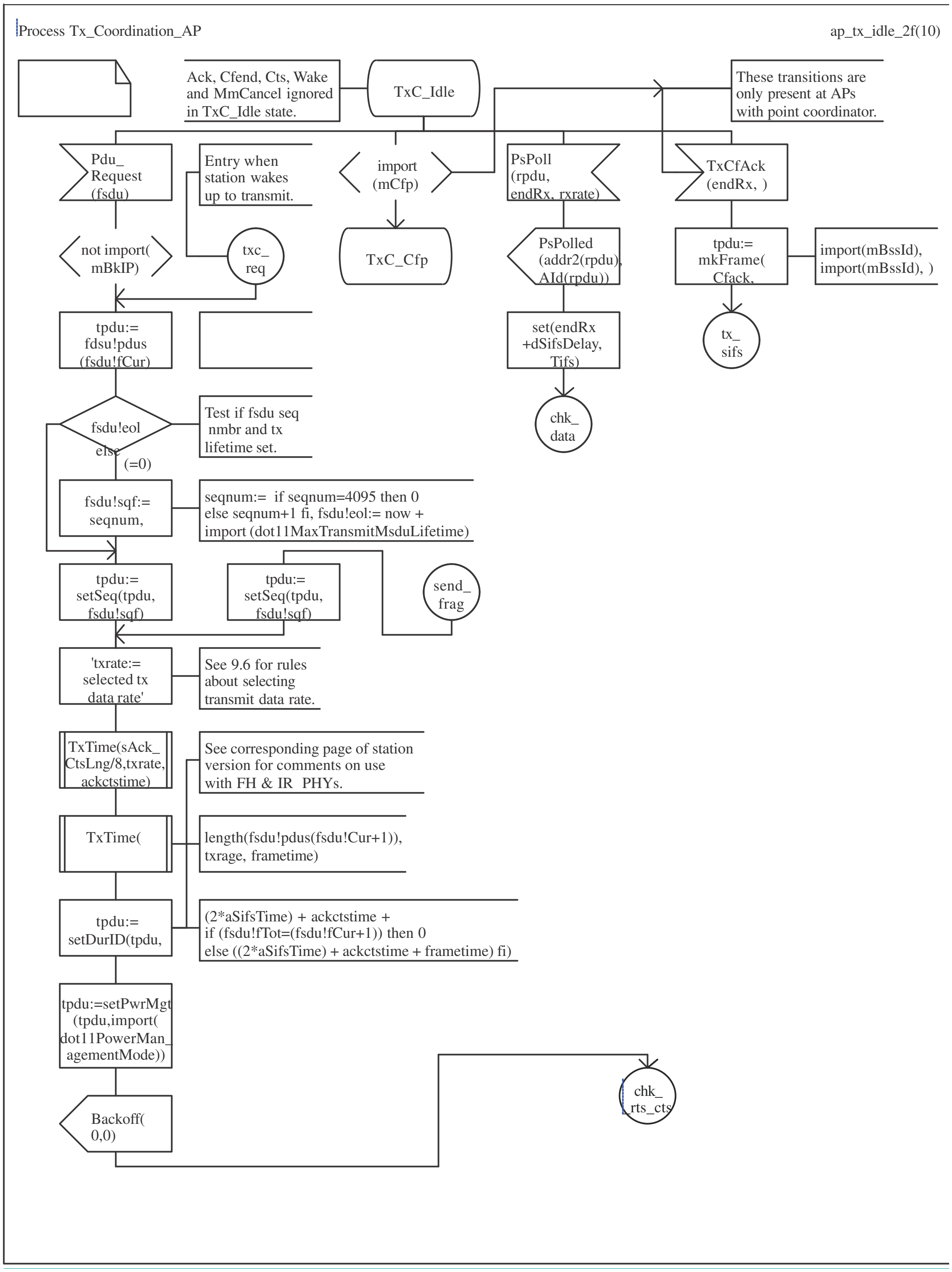
```
timer Tifs, Trsp ;
```

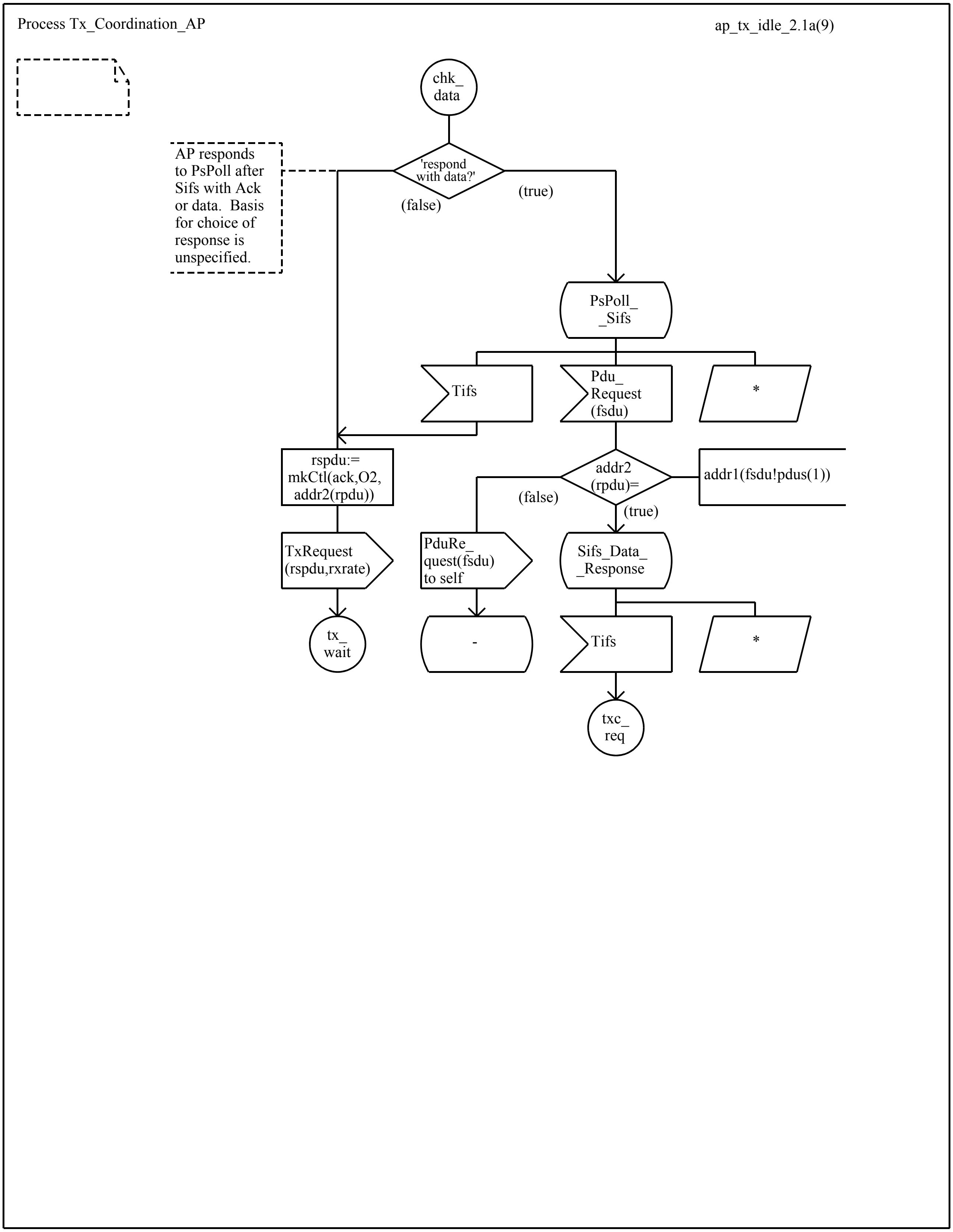
```
dcl ackctstime, bstat, chan , frametime,
      frametime2  Integer ;
dcl ccw  Integer:= aCwMin ;
dcl curPm  Bit ;
dcl doHop, psmChg, cont   Boolean:= false ;
dcl dSifsDelay, endRx  Time ;
dcl fsdu  FragSdu ;
dcl rtype  Ftype ;
dcl rxAid  AssocId ;
dcl rxrate  Rate ;
dcl seqnum, ssrc, slrc, n  Integer:= 0;
dcl tpdu, rpdu, rspdu  Frame ;
dcl txrate  Rate ;
dcl cont  Boolean ;
```

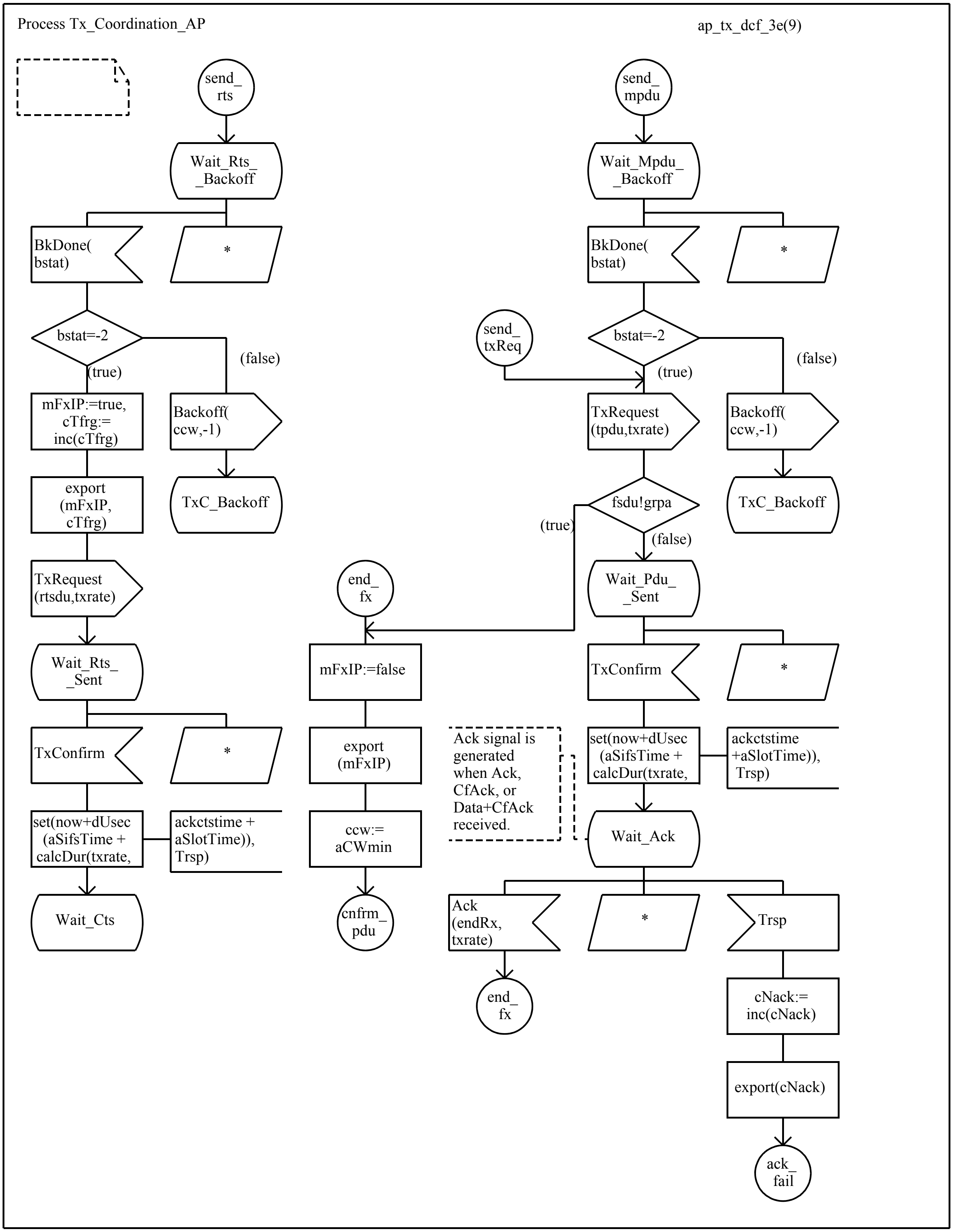
```
Imported dot11RtsThreshold,
dot11ShortRetryLimit,
dot11LongRetryLimit,
dot11FragmentationThreshold,
dot11MaxTransmitMsduLifetime  Integer ;
```

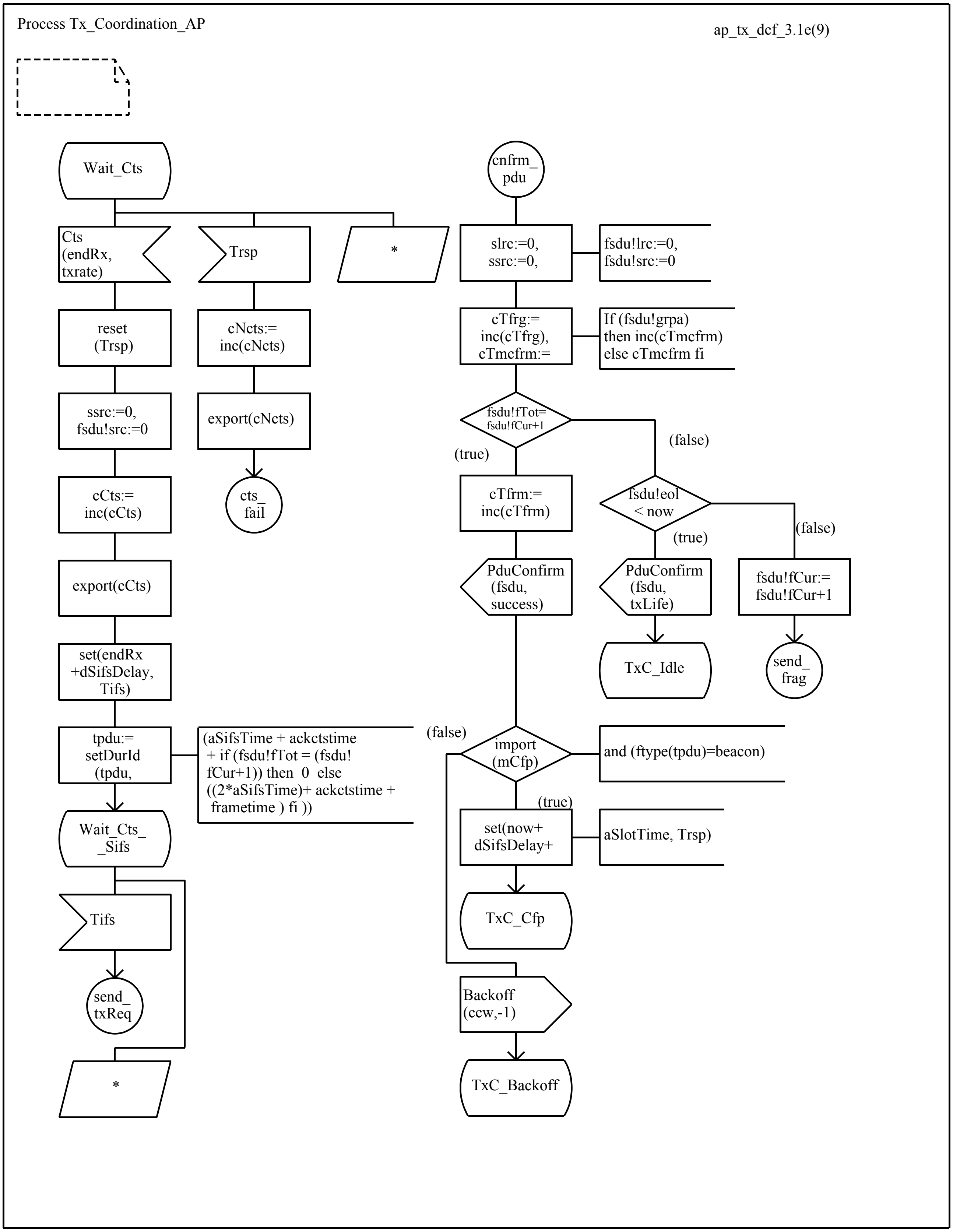
```
/* RANDOM NUMBER FUNCTION */
imported procedure Random ;
  fpar limit  Integer ;   returns Integer ;
```

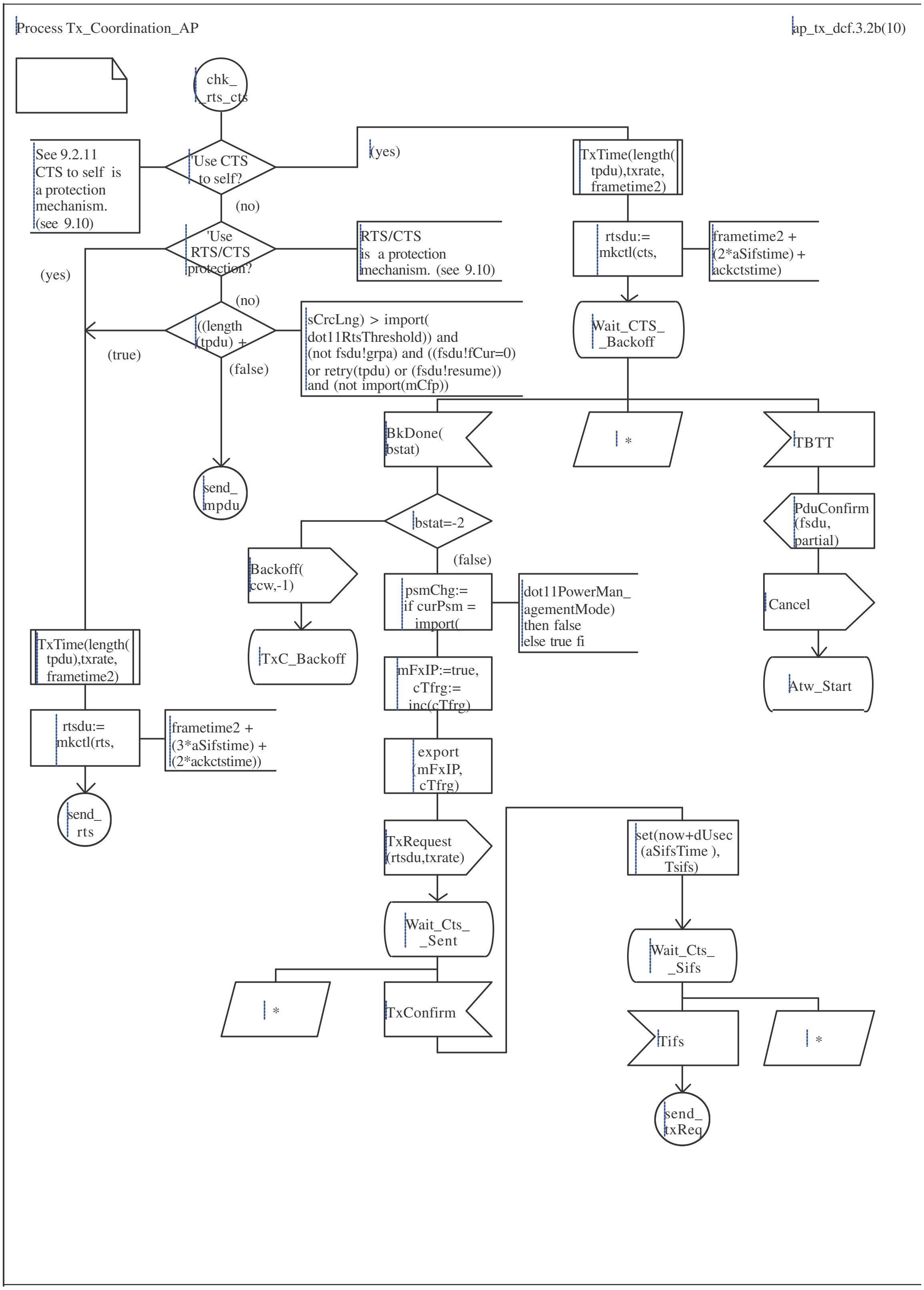
```
dcl exported FxIP  Boolean:= false ;
dcl  cTfrg exported as
    dot11TransmittedFragmentCount,
cTfrm exported as
    dot11TransmittedFrameCount,
cTmcfrm exported as
    dot11MulticastTransmittedFrameCount,
 cFail exported as dot11FailedCount,
 cRtry exported as dot11RetryCount,
 cMrtry exported as dot11MultipleRetryCount,
 cCts exported as dot11RtsSuccessCount,
 cNcts exported as dot11RtsFailureCount,
 cNack exported as dot11AckFailureCount
    Counter32:= 0 ;
```

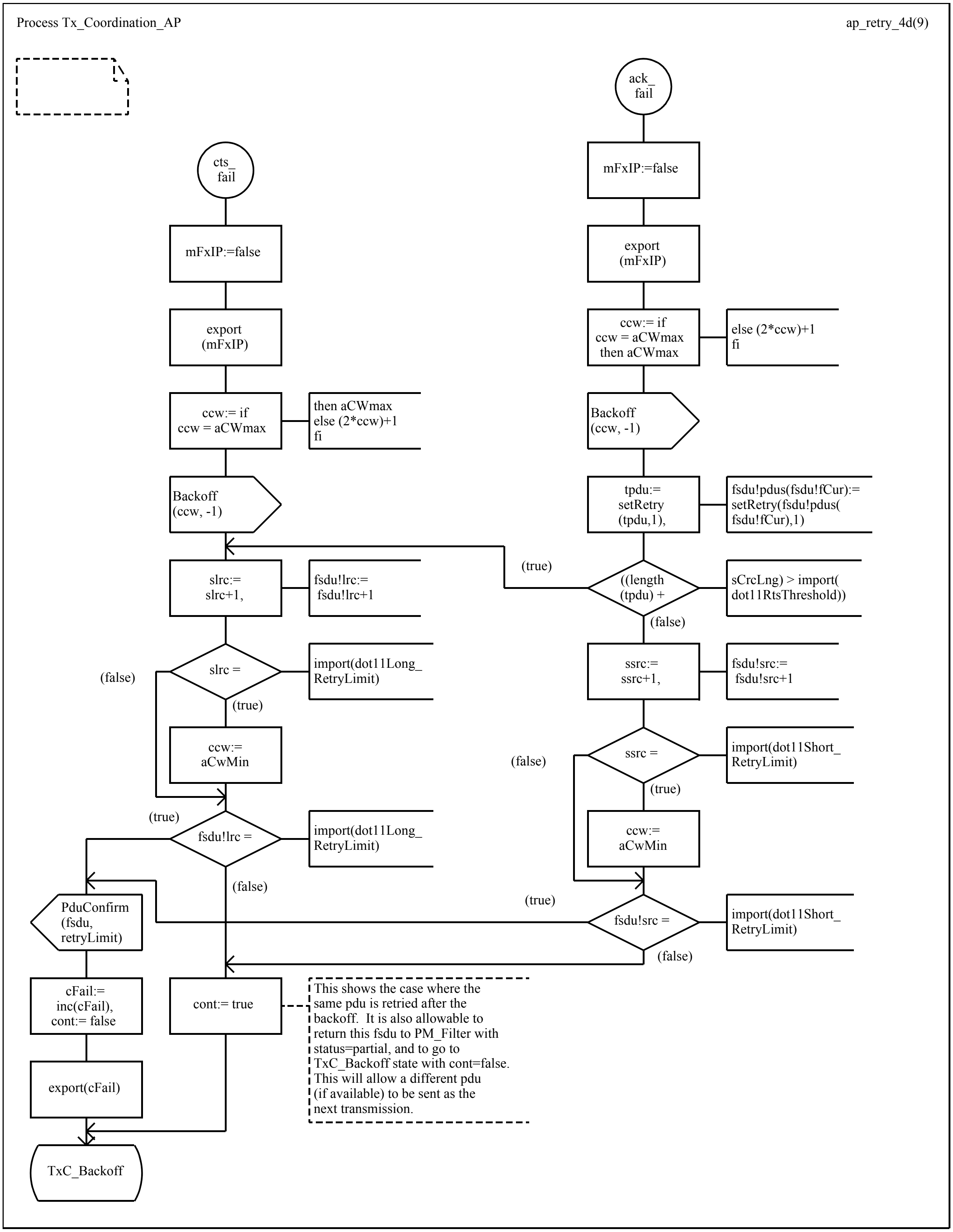

Process Tx_Coordination_AP
ap_tx_idle_2f(10)
Ack, Cfend, Cts, Wake and MmCancel ignored in TxC_Idle state.
TxC_Idle
These transitions are only present at APs with point coordinator.
Pdu_ Request (fsdu)
Entry when station wakes up to transmit.
import (mCfp)
PsPoll (rpdu, endRx, rxrate)
TxCfAck (endRx,)
not import(mBkIP)
txc_ req
TxC_Cfp
PsPolled (addr2(rpdu), AId(rpdu))
tpdu:= mkFrame(Cfack,
import(mBssId), import(mBssId),)
tpdu:= fdsu!pdus (fsdu!fCur)
set(endRx +dSifsDelay, Tifs)
tx_ sifs
fsdu!eol
else
(=0)
Test if fsdu seq nmbr and tx lifetime set.
chk_ data
fsdu!sqf:= seqnum,
seqnum:= if seqnum=4095 then 0 else seqnum+1 fi, fsdu!eol:= now + import (dot11MaxTransmitMsduLifetime)
tpdu:= setSeq(tpdu, fsdu!sqf)
tpdu:= setSeq(tpdu, fsdu!sqf)
send_ frag
'txrate:= selected tx data rate'
See 9.6 for rules about selecting transmit data rate.
TxTime(sAck_ CtsLng/8,txrate, ackctstime)
See corresponding page of station version for comments on use with FH & IR PHYs.
TxTime(
length(fsdu!pdus(fsdu!Cur+1)), txrage, frametime)
tpdu:= setDurID(tpdu,
(2*aSifsTime) + ackctstime + if (fsdu!fTot=(fsdu!fCur+1)) then 0 else ((2*aSifsTime) + ackctstime + frametime) fi)
tpdu:=setPwrMgt (tpdu,import(dot11PowerMan_ agementMode))
Backoff(0,0)
chk_ rts_cts

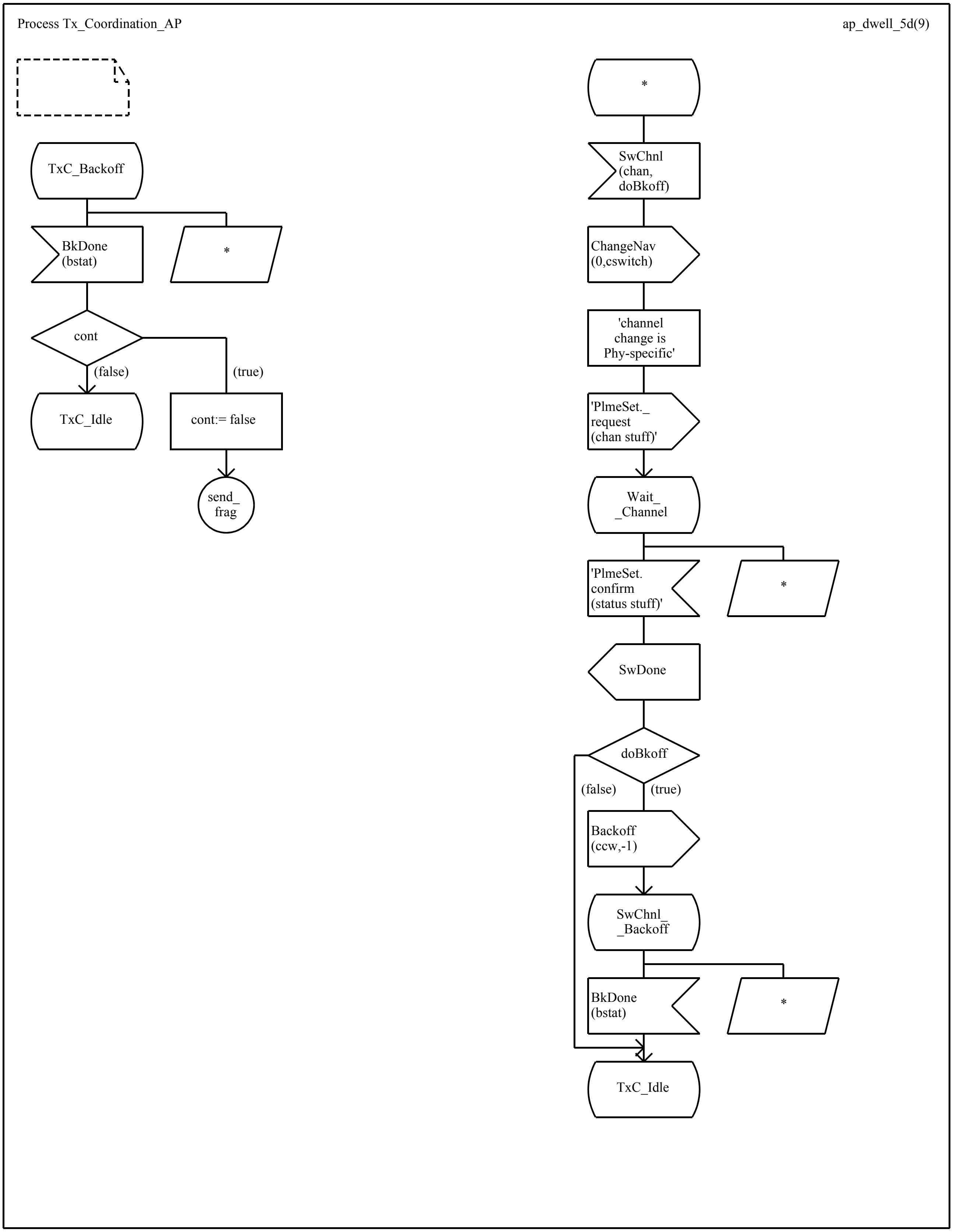

Process Tx_Coordination_AP
ap_tx_idle_2.1a(9)
chk_ data
AP responds to PsPoll after Sifs with Ack or data. Basis for choice of response is unspecified.
'respond with data?'
(false)
(true)
PsPoll_ _Sifs
Tifs
Pdu_ Request (fsdu)
*
rspdu:= mkCtl(ack,O2, addr2(rpdu))
addr2 (rpdu)=
addr1(fsdu!pdus(1))
(false)
(true)
TxRequest (rspdu,rxrate)
PduRe_ quest(fsdu) to self
Sifs_Data_ _Response
tx_ wait
-
Tifs
*
txc_ req

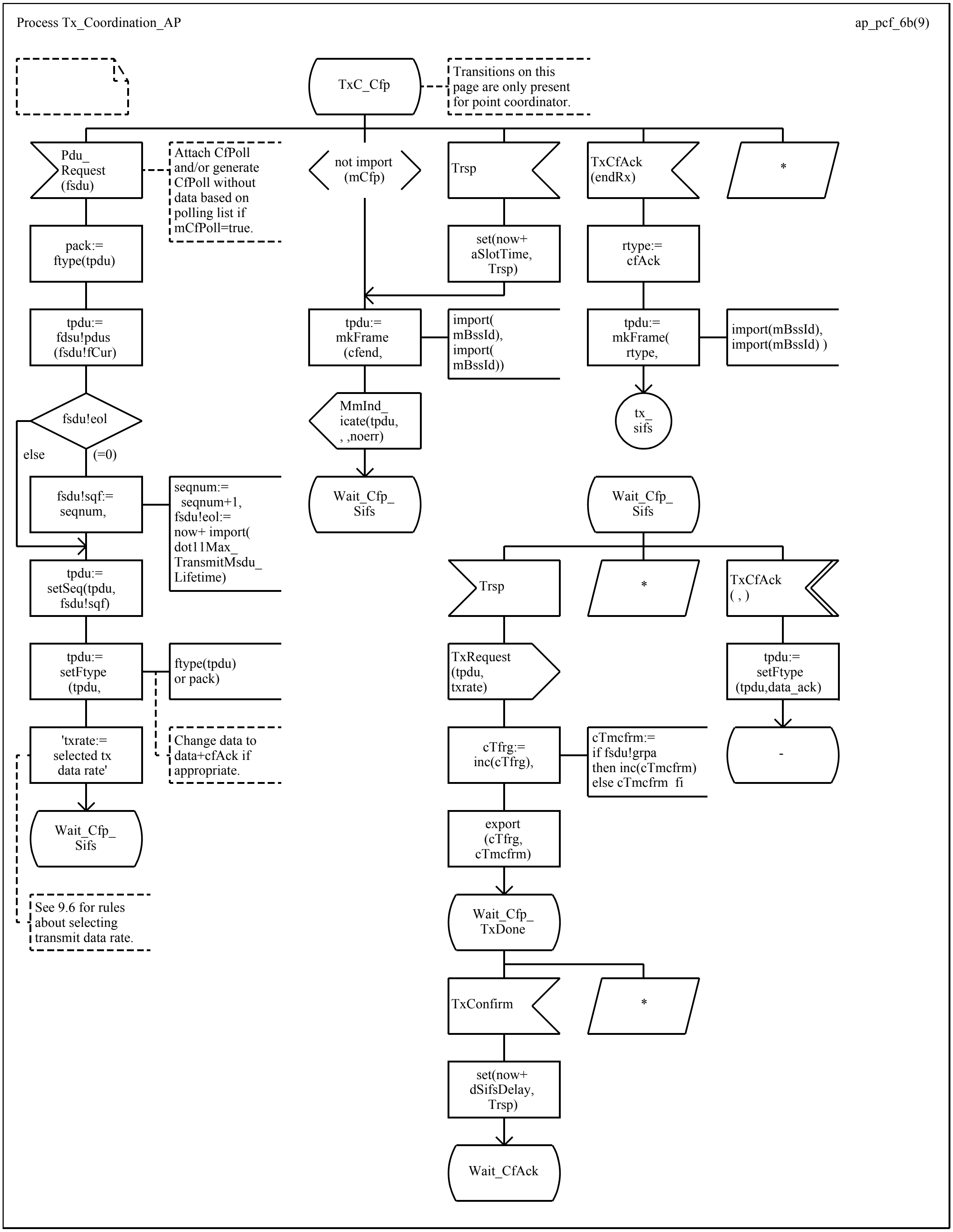

Process Tx_Coordination_AP
ap_tx_dcf_3e(9)
send_ rts
Wait_Rts_ _Backoff
BkDone(bstat)
*
bstat=-2
(true)
(false)
mFxIP:=true, cTfrg:= inc(cTfrg)
Backoff(ccw,-1)
export (mFxIP, cTfrg)
TxC_Backoff
TxRequest (rtsdu,txrate)
Wait_Rts_ _Sent
TxConfirm
*
set(now+dUsec (aSifsTime + calcDur(txrate,
ackctstime + aSlotTime)), Trsp)
Wait_Cts
end_ fx
mFxIP:=false
export (mFxIP)
ccw:= aCWmin
cnfrm_ pdu
send_ mpdu
Wait_Mpdu_ _Backoff
BkDone(bstat)
*
send_ txReq
bstat=-2
(true)
(false)
TxRequest (tpdu,txrate)
Backoff(ccw,-1)
fsdu!grpa
(true)
(false)
TxC_Backoff
Wait_Pdu_ _Sent
TxConfirm
*
Ack signal is generated when Ack, CfAck, or Data+CfAck received.
set(now+dUsec (aSifsTime + calcDur(txrate,
ackctstime +aSlotTime)), Trsp)
Wait_Ack
Ack (endRx, txrate)
*
Trsp
end_ fx
cNack:= inc(cNack)
export(cNack)
ack_ fail

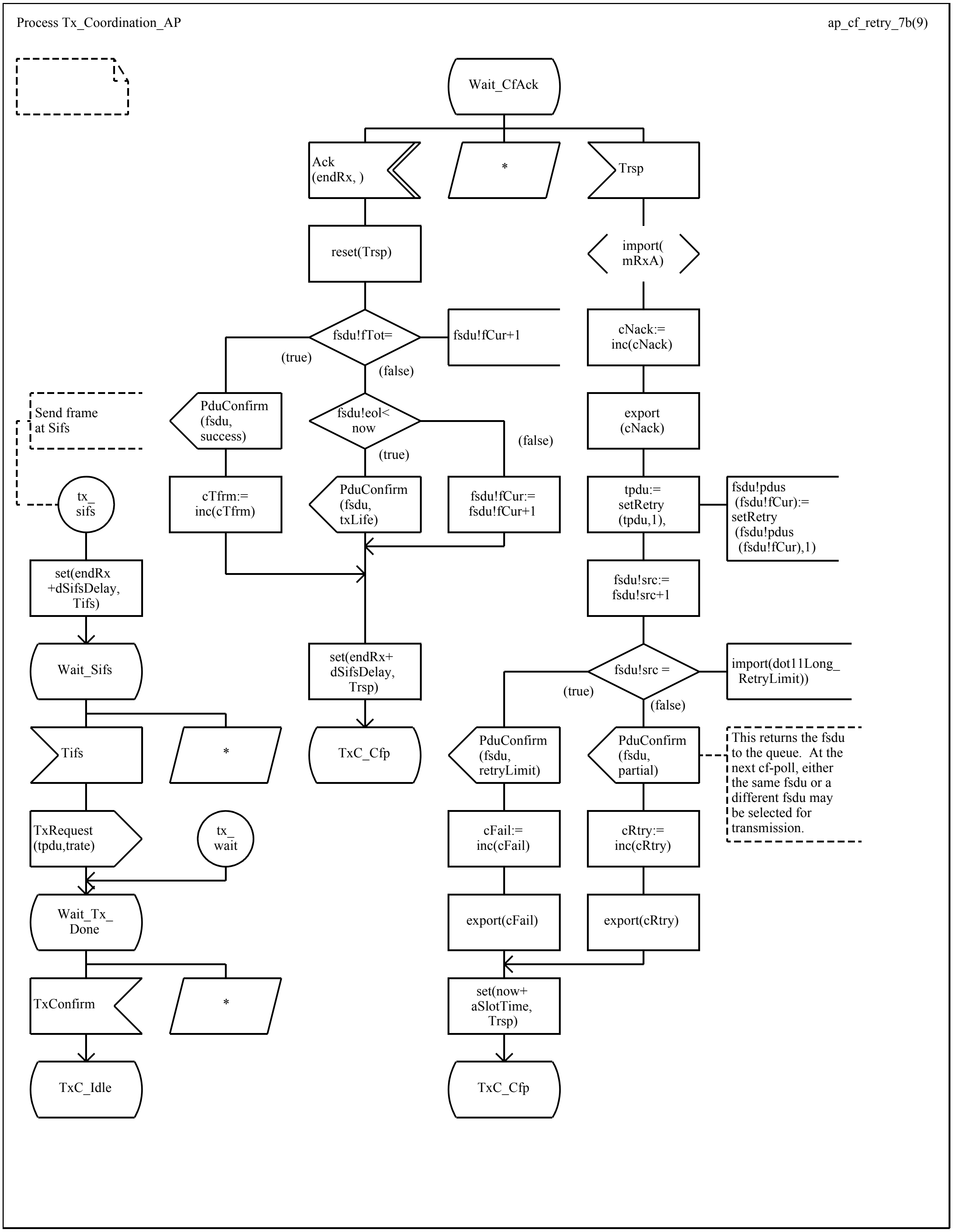

Process Tx_Coordination_AP
ap_tx_dcf_3.1e(9)
Wait_Cts
Cts
(endRx,
txrate)
Trsp
*
reset
(Trsp)
cNcts:=
inc(cNcts)
ssrc:=0,
fsdu!src:=0
export(cNcts)
cCts:=
inc(cCts)
cts_
fail
export(cCts)
set(endRx
+dSifsDelay,
Tifs)
tpdu:=
setDurId
(tpdu,
(aSifsTime + ackctstime
+ if (fsdu!fTot = (fsdu!
fCur+1)) then 0 else
((2*aSifsTime)+ ackctstime +
frametime) fi))
Wait_Cts_
_Sifs
Tifs
send_
txReq
*
cnfrm_
pdu
slrc:=0,
ssrc:=0,
fsdu!lrc:=0,
fsdu!src:=0
cTfrg:=
inc(cTfrg),
cTmcfrm:=
If (fsdu!grpa)
then inc(cTmcfrm)
else cTmcfrm fi
fsdu!fTot=
fsdu!fCur+1
(true)
(false)
cTfrm:=
inc(cTfrm)
fsdu!eol
< now
(true)
(false)
PduConfirm
(fsdu,
success)
PduConfirm
(fsdu,
txLife)
fsdu!fCur:=
fsdu!fCur+1
TxC_Idle
send_
frag
(false)
import
(mCfp)
and (ftype(tpdu)=beacon)
(true)
set(now+
dSifsDelay+
aSlotTime, Trsp)
TxC_Cfp
Backoff
(ccw,-1)
TxC_Backoff

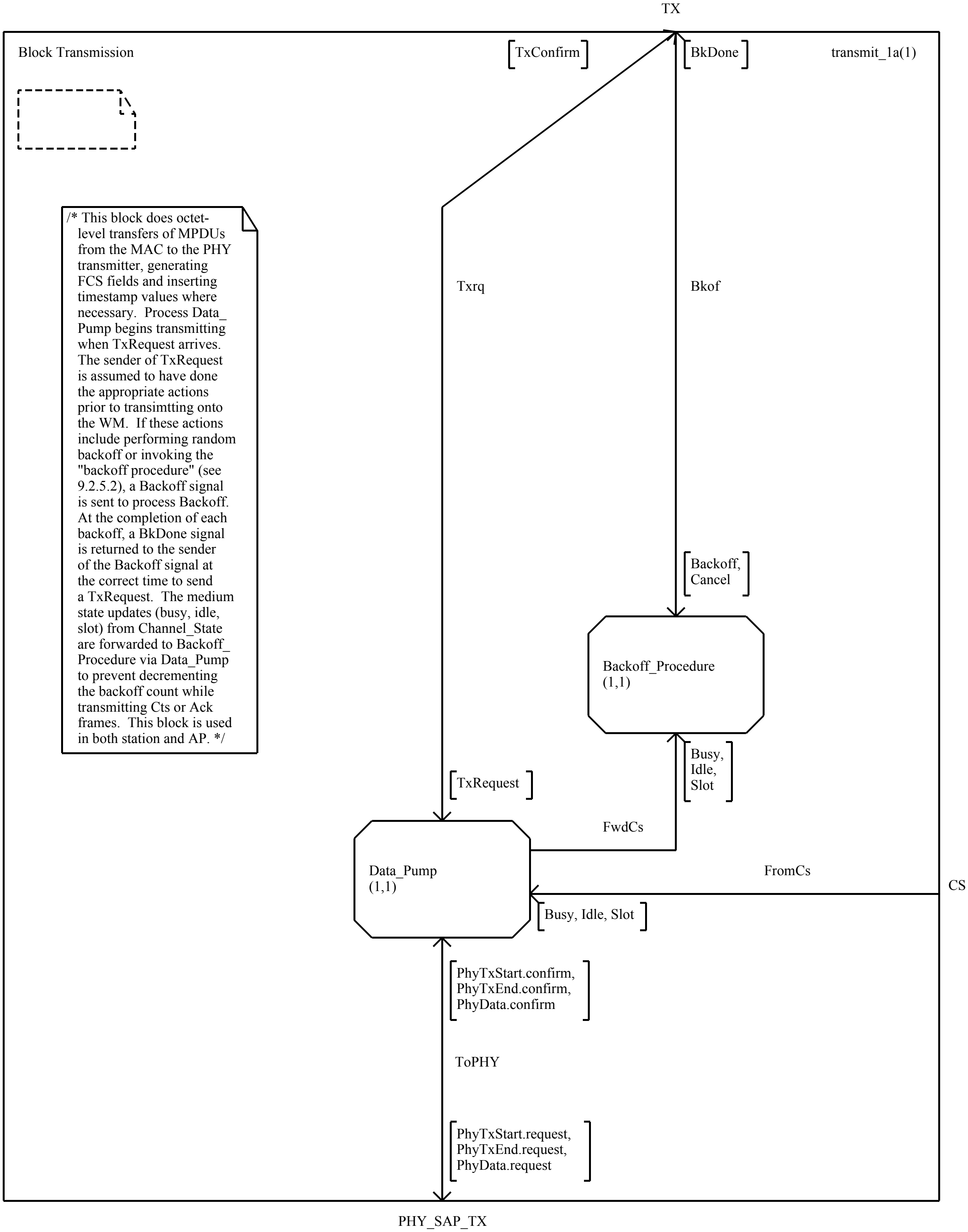

Process Tx_Coordination_AP
ap_tx_dcf.3.2b(10)
chk_
_rts_cts
See 9.2.11
CTS to self is
a protection
mechanism.
(see 9.10)
'Use CTS
to self?
(yes)
(no)
TxTime(length(
tpdu),txrate,
frametime2)
'Use
RTS/CTS
protection?
RTS/CTS
is a protection
mechanism. (see 9.10)
(yes)
rtsdu:=
mkctl(cts,
frametime2 +
(2*aSifstime) +
ackctstime)
(no)
((length
(tpdu) +
sCrcLng) > import(
dot11RtsThreshold)) and
(not fsdu!grpa) and ((fsdu!fCur=0)
or retry(tpdu) or (fsdu!resume))
and (not import(mCfp))
(true)
(false)
Wait_CTS_
_Backoff
BkDone(
bstat)
*
TBTT
send_
mpdu
bstat=-2
PduConfirm
(fsdu,
partial)
(false)
Backoff(
ccw,-1)
psmChg:=
if curPsm =
import(
dot11PowerMan_
agementMode)
then false
else true fi
Cancel
TxTime(length(
tpdu),txrate,
frametime2)
TxC_Backoff
mFxIP:=true,
cTfrg:=
inc(cTfrg)
Atw_Start
rtsdu:=
mkctl(rts,
frametime2 +
(3*aSifstime) +
(2*ackctstime))
export
(mFxIP,
cTfrg)
send_
rts
TxRequest
(rtsdu,txrate)
set(now+dUsec
(aSifsTime),
Tsifs)
Wait_Cts_
_Sent
Wait_Cts_
_Sifs
*
TxConfirm
Tifs
*
send_
txReq


Process Tx_Coordination_AP
ap_retry_4d(9)
cts_ fail
mFxIP:=false
export (mFxIP)
ccw:= if ccw = aCWmax
then aCWmax else (2*ccw)+1 fi
Backoff (ccw, -1)
slrc:= slrc+1,
fsdu!lrc:= fsdu!lrc+1
slrc =
import(dot11Long_ RetryLimit)
(false)
(true)
ccw:= aCwMin
fsdu!lrc =
import(dot11Long_ RetryLimit)
(true)
(false)
PduConfirm (fsdu, retryLimit)
cFail:= inc(cFail), cont:= false
export(cFail)
TxC_Backoff
cont:= true
This shows the case where the same pdu is retried after the backoff. It is also allowable to return this fsdu to PM_Filter with status=partial, and to go to TxC_Backoff state with cont=false. This will allow a different pdu (if available) to be sent as the next transmission.
ack_ fail
mFxIP:=false
export (mFxIP)
ccw:= if ccw = aCWmax then aCWmax
else (2*ccw)+1 fi
Backoff (ccw, -1)
tpdu:= setRetry (tpdu,1),
fsdu!pdus(fsdu!fCur):= setRetry(fsdu!pdus(fsdu!fCur),1)
(true)
((length (tpdu) +
sCrcLng) > import(dot11RtsThreshold))
(false)
ssrc:= ssrc+1,
fsdu!src:= fsdu!src+1
ssrc =
import(dot11Short_ RetryLimit)
(false)
(true)
ccw:= aCwMin
(true)
fsdu!src =
import(dot11Short_ RetryLimit)
(false)

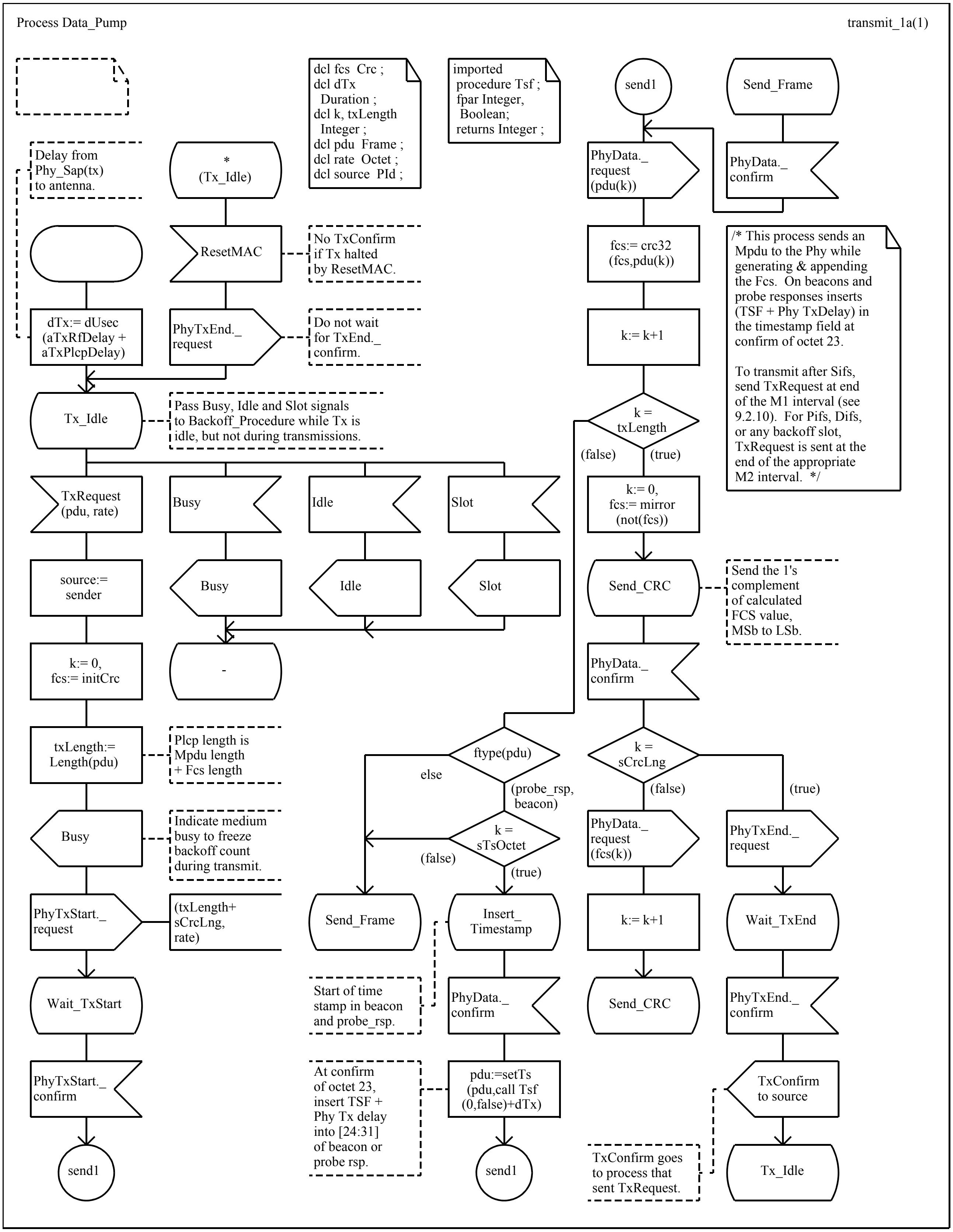

Process Tx_Coordination_AP
ap_dwell_5d(9)
TxC_Backoff
BkDone (bstat)
*
cont
(false)
(true)
TxC_Idle
cont:= false
send_ frag
*
SwChnl (chan, doBkoff)
ChangeNav (0,cswitch)
'channel change is Phy-specific'
'PlmeSet._ request (chan stuff)'
Wait_ _Channel
'PlmeSet. confirm (status stuff)'
*
SwDone
doBkoff
(false)
(true)
Backoff (ccw,-1)
SwChnl_ _Backoff
BkDone (bstat)
*
TxC_Idle

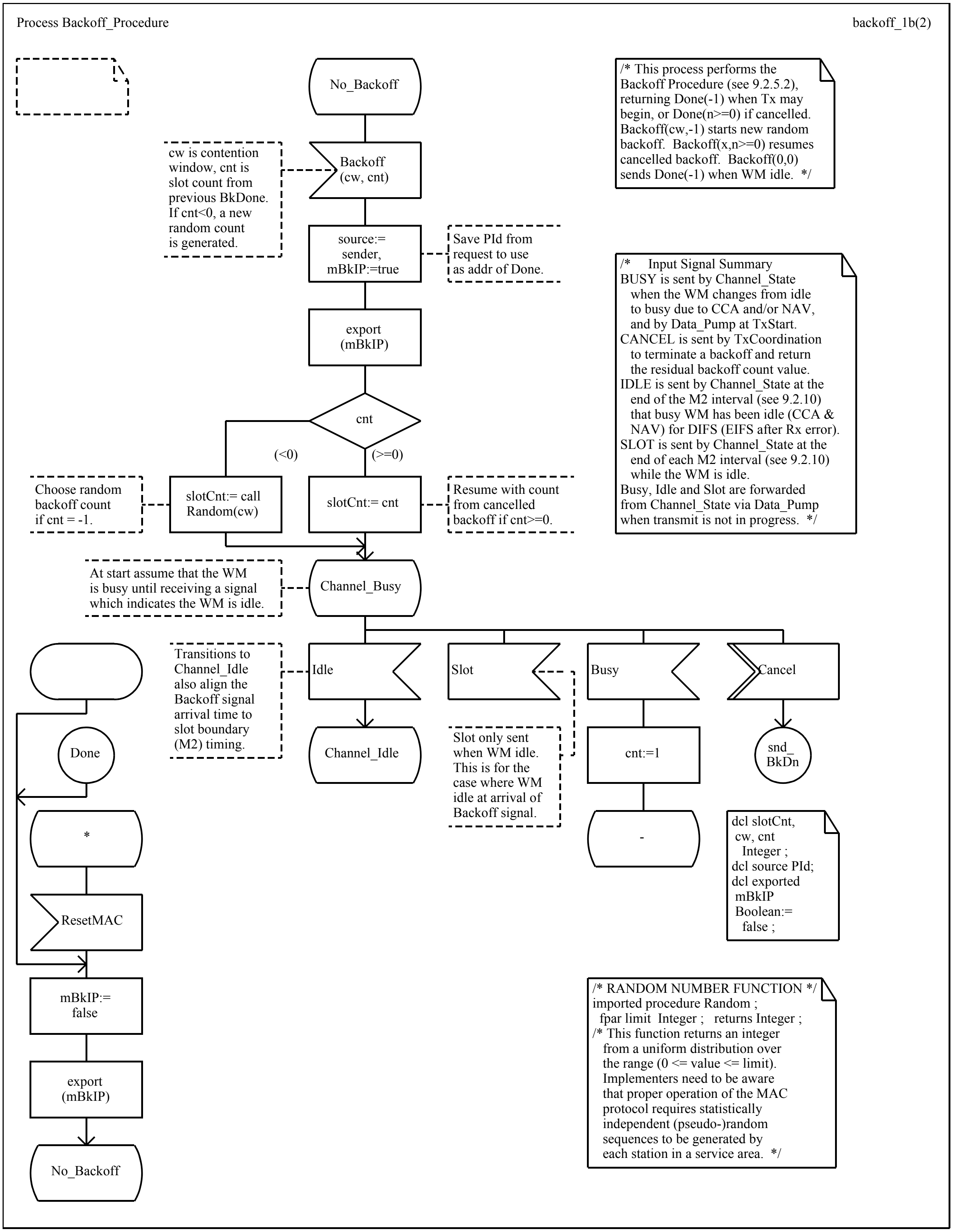

Process Tx_Coordination_AP
ap_pcf_6b(9)
TxC_Cfp
Transitions on this page are only present for point coordinator.
Pdu_ Request (fsdu)
Attach CfPoll and/or generate CfPoll without data based on polling list if mCfPoll=true.
not import (mCfp)
Trsp
TxCfAck (endRx)
*
pack:= ftype(tpdu)
set(now+ aSlotTime, Trsp)
rtype:= cfAck
tpdu:= fdsu!pdus (fsdu!fCur)
tpdu:= mkFrame (cfend,
import(mBssId), import(mBssId))
tpdu:= mkFrame(rtype,
import(mBssId), import(mBssId))
fsdu!eol
else
(=0)
MmInd_ icate(tpdu, , ,noerr)
tx_ sifs
fsdu!sqf:= seqnum,
seqnum:= seqnum+1, fsdu!eol:= now+ import(dot11Max_ TransmitMsdu_ Lifetime)
Wait_Cfp_ Sifs
Wait_Cfp_ Sifs
tpdu:= setSeq(tpdu, fsdu!sqf)
Trsp
*
TxCfAck (,)
tpdu:= setFtype (tpdu,
ftype(tpdu) or pack)
TxRequest (tpdu, txrate)
tpdu:= setFtype (tpdu,data_ack)
'txrate:= selected tx data rate'
Change data to data+cfAck if appropriate.
cTfrg:= inc(cTfrg),
cTmcfrm:= if fsdu!grpa then inc(cTmcfrm) else cTmcfrm fi
-
Wait_Cfp_ Sifs
export (cTfrg, cTmcfrm)
See 9.6 for rules about selecting transmit data rate.
Wait_Cfp_ TxDone
TxConfirm
*
set(now+ dSifsDelay, Trsp)
Wait_CfAck


Process Tx_Coordination_AP
ap_cf_retry_7b(9)
Wait_CfAck
Ack (endRx,)
*
Trsp
reset(Trsp)
import(mRxA)
fsdu!fTot=
fsdu!fCur+1
(true)
(false)
cNack:= inc(cNack)
Send frame at Sifs
PduConfirm (fsdu, success)
fsdu!eol< now
(false)
(true)
export (cNack)
tx_ sifs
cTfrm:= inc(cTfrm)
PduConfirm (fsdu, txLife)
fsdu!fCur:= fsdu!fCur+1
tpdu:= setRetry (tpdu,1),
fsdu!pdus (fsdu!fCur):= setRetry (fsdu!pdus (fsdu!fCur),1)
set(endRx +dSifsDelay, Tifs)
fsdu!src:= fsdu!src+1
Wait_Sifs
set(endRx+ dSifsDelay, Trsp)
fsdu!src =
import(dot11Long_ RetryLimit))
(true)
(false)
Tifs
*
TxC_Cfp
PduConfirm (fsdu, retryLimit)
PduConfirm (fsdu, partial)
This returns the fsdu to the queue. At the next cf-poll, either the same fsdu or a different fsdu may be selected for transmission.
TxRequest (tpdu,trate)
tx_ wait
cFail:= inc(cFail)
cRtry:= inc(cRtry)
Wait_Tx_ Done
export(cFail)
export(cRtry)
TxConfirm
*
set(now+ aSlotTime, Trsp)
TxC_Idle
TxC_Cfp



/* This block does octet-level transfers of MPDUs from the MAC to the PHY transmitter, generating FCS fields and inserting timestamp values where necessary. Process Data_ Pump begins transmitting when TxRequest arrives. The sender of TxRequest is assumed to have done the appropriate actions prior to transimtting onto the WM. If these actions include performing random backoff or invoking the "backoff procedure" (see 9.2.5.2), a Backoff signal is sent to process Backoff. At the completion of each backoff, a BkDone signal is returned to the sender of the Backoff signal at the correct time to send a TxRequest. The medium state updates (busy, idle, slot) from Channel_State are forwarded to Backoff_ Procedure via Data_Pump to prevent decrementing the backoff count while transmitting Cts or Ack frames. This block is used in both station and AP. */


Process Data_Pump
transmit_1a(1)
dcl fcs Crc ;
dcl dTx Duration ;
dcl k, txLength Integer ;
dcl pdu Frame ;
dcl rate Octet ;
dcl source PId ;
imported procedure Tsf ;
fpar Integer, Boolean;
returns Integer ;
send1
Send_Frame
Delay from Phy_Sap(tx) to antenna.
*
(Tx_Idle)
PhyData._ request (pdu(k))
PhyData._ confirm
ResetMAC
No TxConfirm if Tx halted by ResetMAC.
fcs:= crc32 (fcs,pdu(k))
/* This process sends an Mpdu to the Phy while generating & appending the Fcs. On beacons and probe responses inserts (TSF + Phy TxDelay) in the timestamp field at confirm of octet 23.
To transmit after Sifs, send TxRequest at end of the M1 interval (see 9.2.10). For Pifs, Difs, or any backoff slot, TxRequest is sent at the end of the appropriate M2 interval. */
dTx:= dUsec (aTxRfDelay + aTxPlcpDelay)
PhyTxEnd._ request
Do not wait for TxEnd._ confirm.
k:= k+1
Tx_Idle
Pass Busy, Idle and Slot signals to Backoff_Procedure while Tx is idle, but not during transmissions.
k = txLength
(false)
(true)
TxRequest (pdu, rate)
Busy
Idle
Slot
k:= 0, fcs:= mirror (not(fcs))
source:= sender
Busy
Idle
Slot
Send_CRC
Send the 1's complement of calculated FCS value, MSb to LSb.
k:= 0, fcs:= initCrc
-
PhyData._ confirm
txLength:= Length(pdu)
Plcp length is Mpdu length + Fcs length
ftype(pdu)
else
k = sCrcLng
(probe_rsp, beacon)
(false)
(true)
Busy
Indicate medium busy to freeze backoff count during transmit.
k = sTsOctet
(false)
(true)
PhyData._ request (fcs(k))
PhyTxEnd._ request
PhyTxStart._ request
(txLength+ sCrcLng, rate)
Send_Frame
Insert_ Timestamp
k:= k+1
Wait_TxEnd
Wait_TxStart
Start of time stamp in beacon and probe_rsp.
PhyData._ confirm
Send_CRC
PhyTxEnd._ confirm
PhyTxStart._ confirm
At confirm of octet 23, insert TSF + Phy Tx delay into [24:31] of beacon or probe rsp.
pdu:=setTs (pdu,call Tsf (0,false)+dTx)
TxConfirm to source
send1
send1
TxConfirm goes to process that sent TxRequest.
Tx_Idle



cw is contention window, cnt is slot count from previous BkDone. If cnt<0, a new random count is generated.

Save PId from request to use as addr of Done.

Choose random backoff count if cnt = -1.

Resume with count from cancelled backoff if cnt>=0.

At start assume that the WM is busy until receiving a signal which indicates the WM is idle.

Transitions to Channel_Idle also align the Backoff signal arrival time to slot boundary (M2) timing.

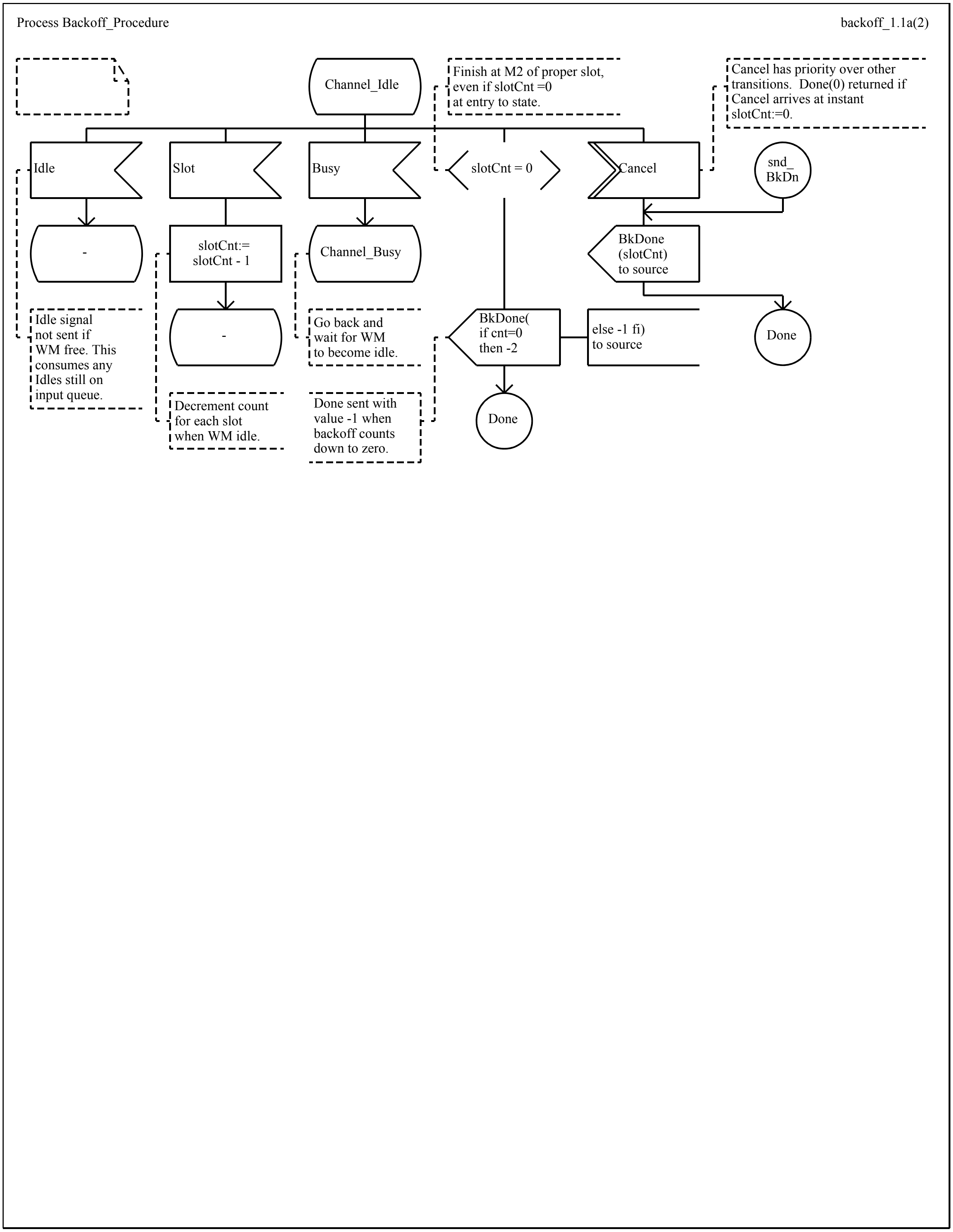
Slot only sent when WM idle. This is for the case where WM idle at arrival of Backoff signal.

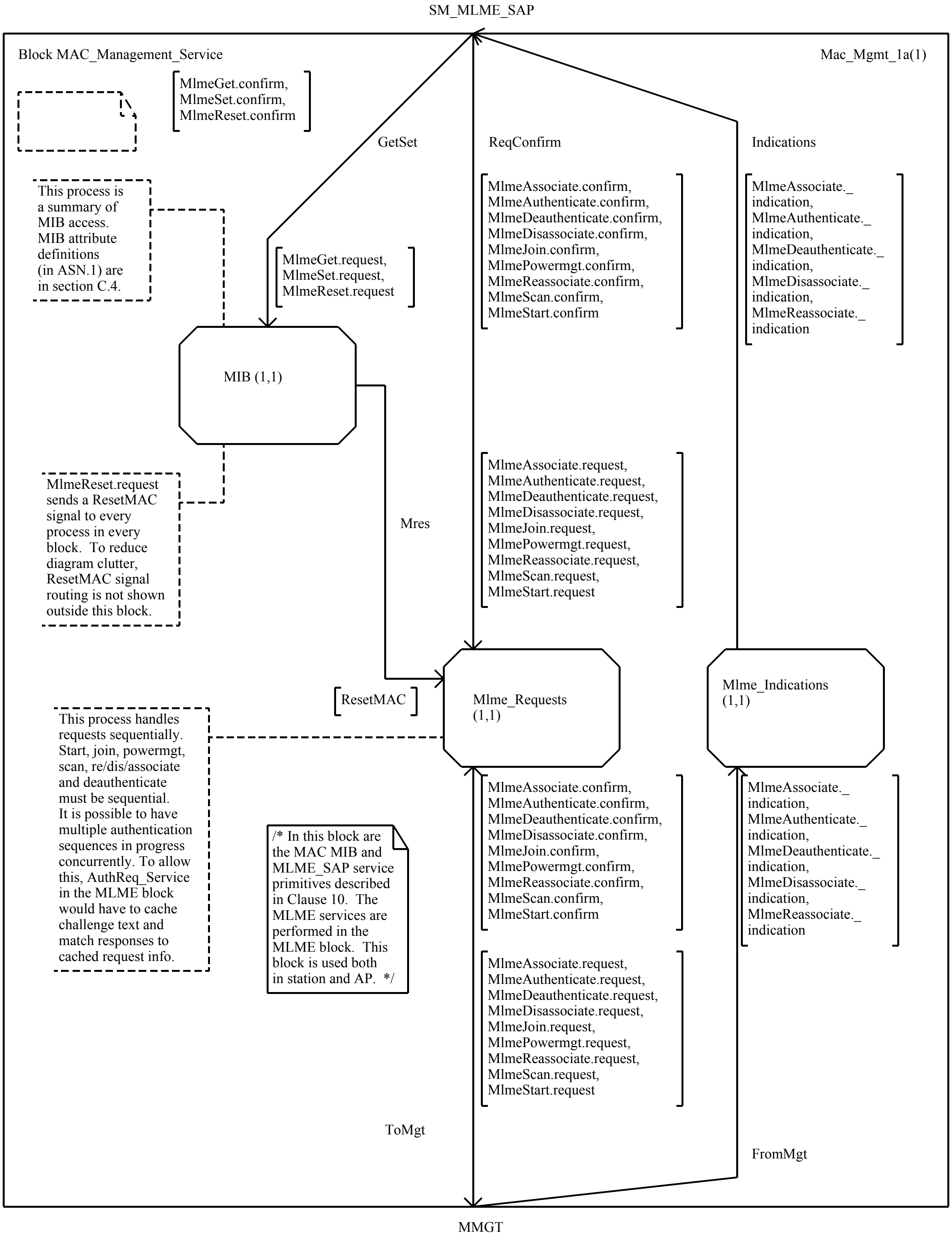
/* This process performs the Backoff Procedure (see 9.2.5.2), returning Done(-1) when Tx may begin, or Done(n>=0) if cancelled. Backoff(cw,-1) starts new random backoff. Backoff(x,n>=0) resumes cancelled backoff. Backoff(0,0) sends Done(-1) when WM idle. */

/* Input Signal Summary
BUSY is sent by Channel_State when the WM changes from idle to busy due to CCA and/or NAV, and by Data_Pump at TxStart.
CANCEL is sent by TxCoordination to terminate a backoff and return the residual backoff count value.
IDLE is sent by Channel_State at the end of the M2 interval (see 9.2.10) that busy WM has been idle (CCA & NAV) for DIFS (EIFS after Rx error).
SLOT is sent by Channel_State at the end of each M2 interval (see 9.2.10) while the WM is idle.
Busy, Idle and Slot are forwarded from Channel_State via Data_Pump when transmit is not in progress. */

```
dcl slotCnt,
 cw, cnt
   Integer ;
dcl source PId;
dcl exported
 mBkIP
 Boolean:=
   false ;
```

```
/* RANDOM NUMBER FUNCTION */
imported procedure Random ;
  fpar limit  Integer ;   returns Integer ;
/* This function returns an integer
   from a uniform distribution over
   the range (0 <= value <= limit).
   Implementers need to be aware
   that proper operation of the MAC
   protocol requires statistically
   independent (pseudo-)random
   sequences to be generated by
   each station in a service area.  */
```


Process Backoff_Procedure
backoff_1.1a(2)
Channel_Idle
Finish at M2 of proper slot, even if slotCnt =0 at entry to state.
Cancel has priority over other transitions. Done(0) returned if Cancel arrives at instant slotCnt:=0.
Idle
Slot
Busy
slotCnt = 0
Cancel
snd_ BkDn
-
slotCnt:= slotCnt - 1
Channel_Busy
BkDone (slotCnt) to source
-
Idle signal not sent if WM free. This consumes any Idles still on input queue.
Go back and wait for WM to become idle.
BkDone(if cnt=0 then -2
else -1 fi) to source
Done
Decrement count for each slot when WM idle.
Done sent with value -1 when backoff counts down to zero.
Done

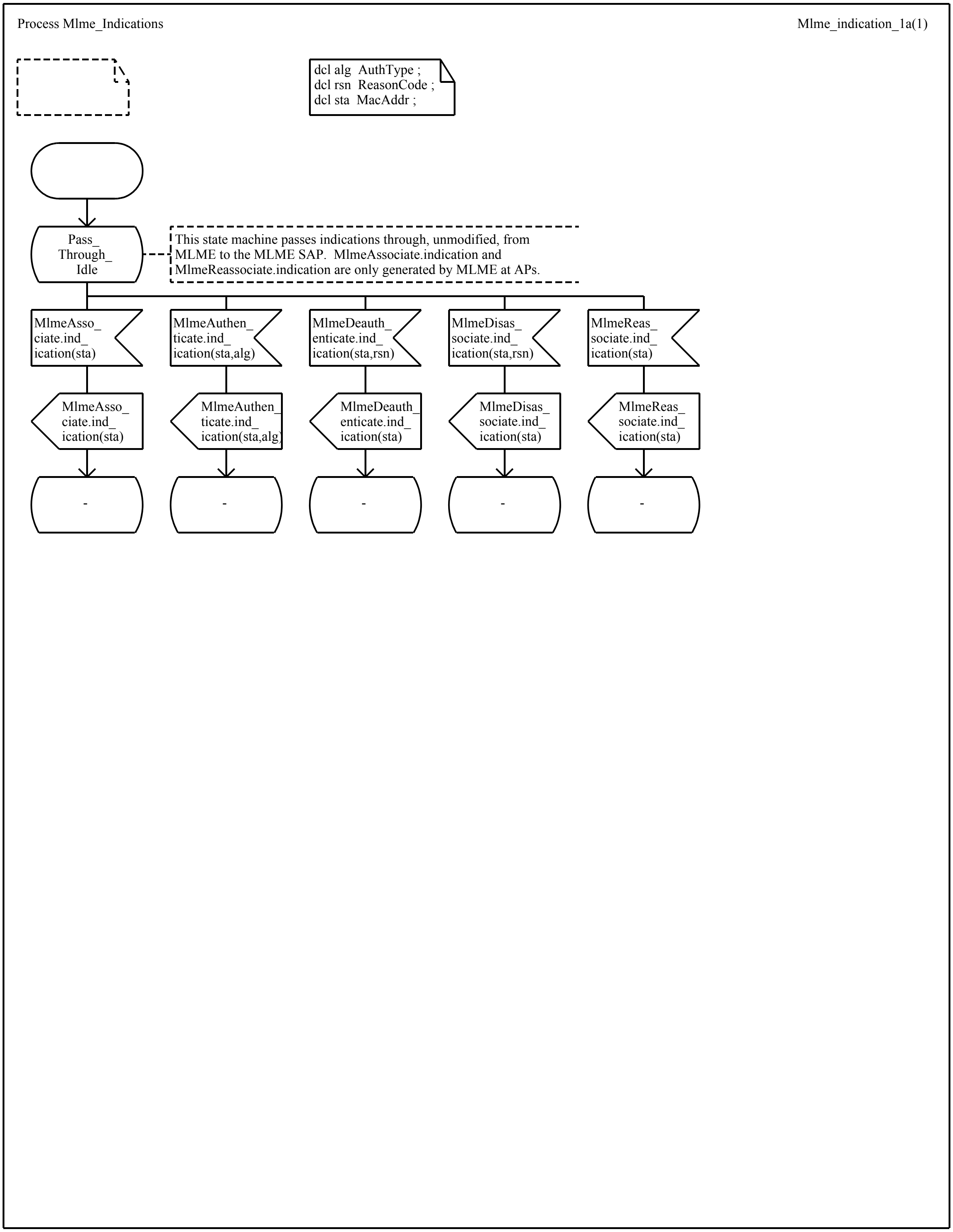


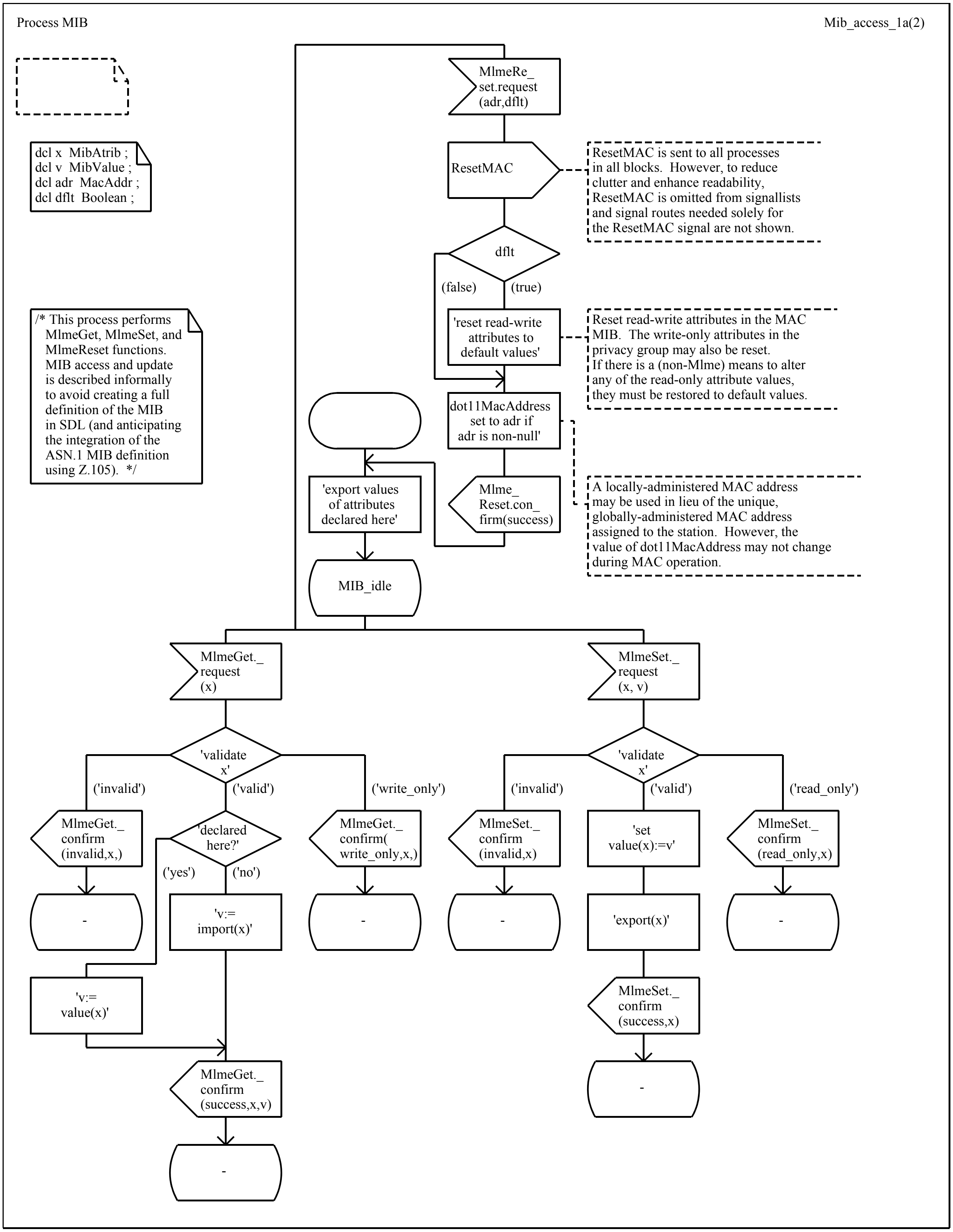
This process is a summary of MIB access. MIB attribute definitions (in ASN.1) are in section C.4.

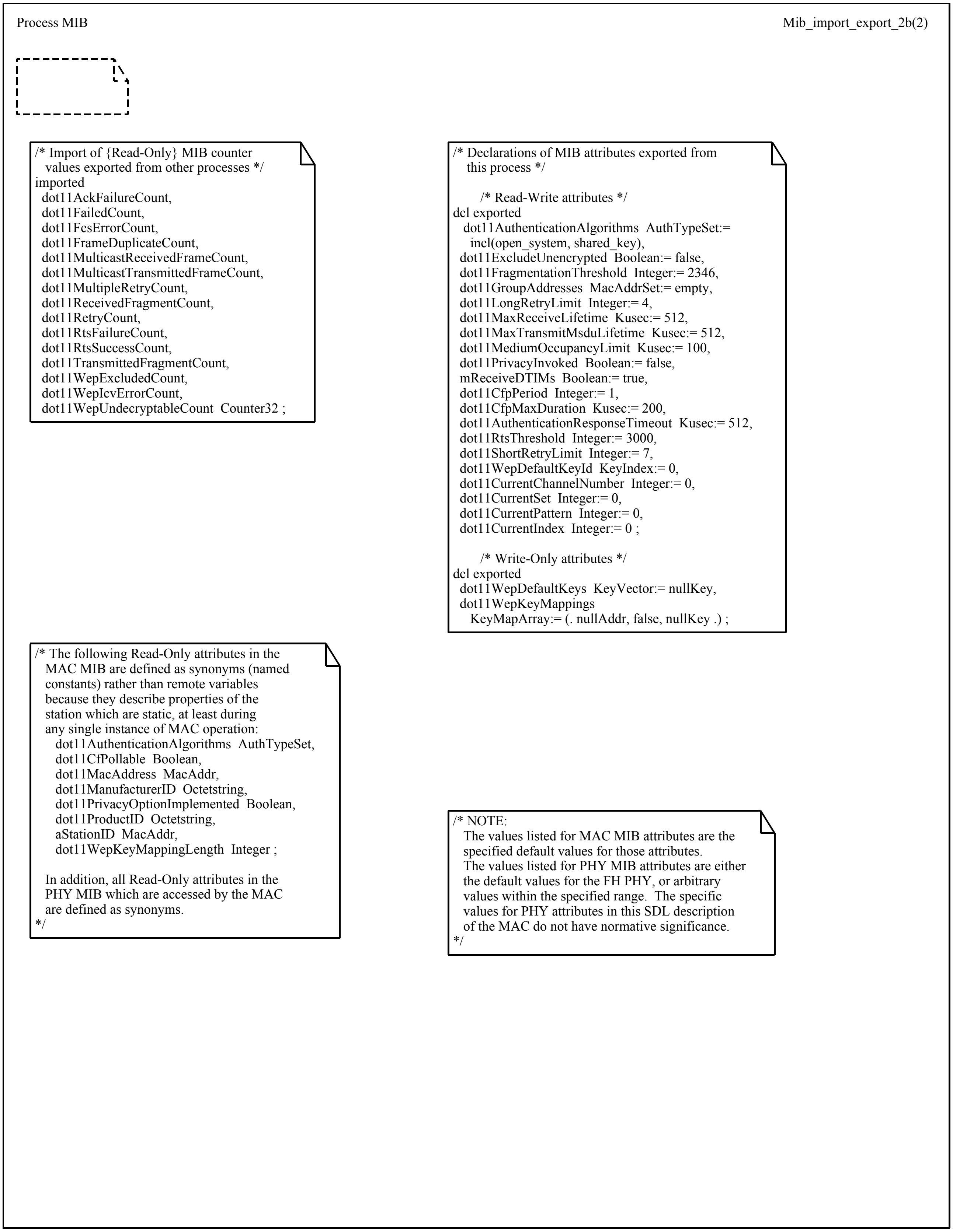
MlmeReset.request sends a ResetMAC signal to every process in every block. To reduce diagram clutter, ResetMAC signal routing is not shown outside this block.

This process handles requests sequentially. Start, join, powermgt, scan, re/dis/associate and deauthenticate must be sequential. It is possible to have multiple authentication sequences in progress concurrently. To allow this, AuthReq_Service in the MLME block would have to cache challenge text and match responses to cached request info.

/* In this block are the MAC MIB and MLME_SAP service primitives described in Clause 10. The MLME services are performed in the MLME block. This block is used both in station and AP. */


Process Mlme_Indications
Mlme_indication_1a(1)
dcl alg AuthType ;
dcl rsn ReasonCode ;
dcl sta MacAddr ;
Pass_ Through_ Idle
This state machine passes indications through, unmodified, from MLME to the MLME SAP. MlmeAssociate.indication and MlmeReassociate.indication are only generated by MLME at APs.
MlmeAsso_ ciate.ind_ ication(sta)
MlmeAuthen_ ticate.ind_ ication(sta,alg)
MlmeDeauth_ enticate.ind_ ication(sta,rsn)
MlmeDisas_ sociate.ind_ ication(sta,rsn)
MlmeReas_ sociate.ind_ ication(sta)
MlmeAsso_ ciate.ind_ ication(sta)
MlmeAuthen_ ticate.ind_ ication(sta,alg)
MlmeDeauth_ enticate.ind_ ication(sta)
MlmeDisas_ sociate.ind_ ication(sta)
MlmeReas_ sociate.ind_ ication(sta)
-
-
-
-
-


Process MIB
Mib_access_1a(2)
dcl x MibAtrib ;
dcl v MibValue ;
dcl adr MacAddr ;
dcl dflt Boolean ;
/* This process performs MlmeGet, MlmeSet, and MlmeReset functions. MIB access and update is described informally to avoid creating a full definition of the MIB in SDL (and anticipating the integration of the ASN.1 MIB definition using Z.105). */
MlmeRe_ set.request (adr,dflt)
ResetMAC
ResetMAC is sent to all processes in all blocks. However, to reduce clutter and enhance readability, ResetMAC is omitted from signallists and signal routes needed solely for the ResetMAC signal are not shown.
dflt
(false)
(true)
'reset read-write attributes to default values'
Reset read-write attributes in the MAC MIB. The write-only attributes in the privacy group may also be reset. If there is a (non-Mlme) means to alter any of the read-only attribute values, they must be restored to default values.
dot11MacAddress set to adr if adr is non-null'
A locally-administered MAC address may be used in lieu of the unique, globally-administered MAC address assigned to the station. However, the value of dot11MacAddress may not change during MAC operation.
Mlme_ Reset.con_ firm(success)
'export values of attributes declared here'
MIB_idle
MlmeGet._ request (x)
'validate x'
('invalid')
('valid')
('write_only')
MlmeGet._ confirm (invalid,x,)
-
'declared here?'
('yes')
('no')
'v:= import(x)'
'v:= value(x)'
MlmeGet._ confirm (success,x,v)
-
MlmeGet._ confirm(write_only,x,)
-
MlmeSet._ request (x, v)
'validate x'
('invalid')
('valid')
('read_only')
MlmeSet._ confirm (invalid,x)
-
'set value(x):=v'
'export(x)'
MlmeSet._ confirm (success,x)
-
MlmeSet._ confirm (read_only,x)
-



Process MIB

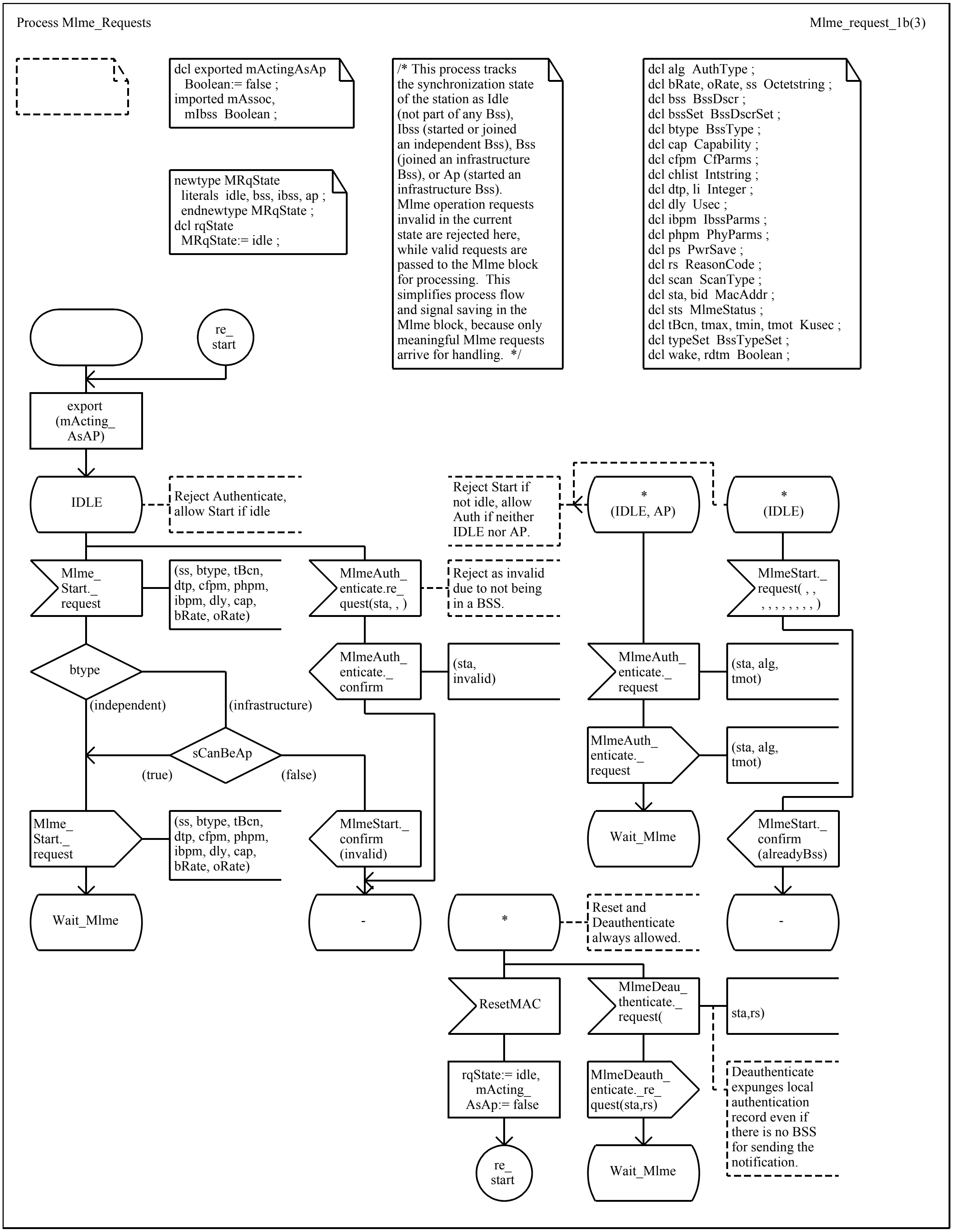
Mib_import_export_2b(2)

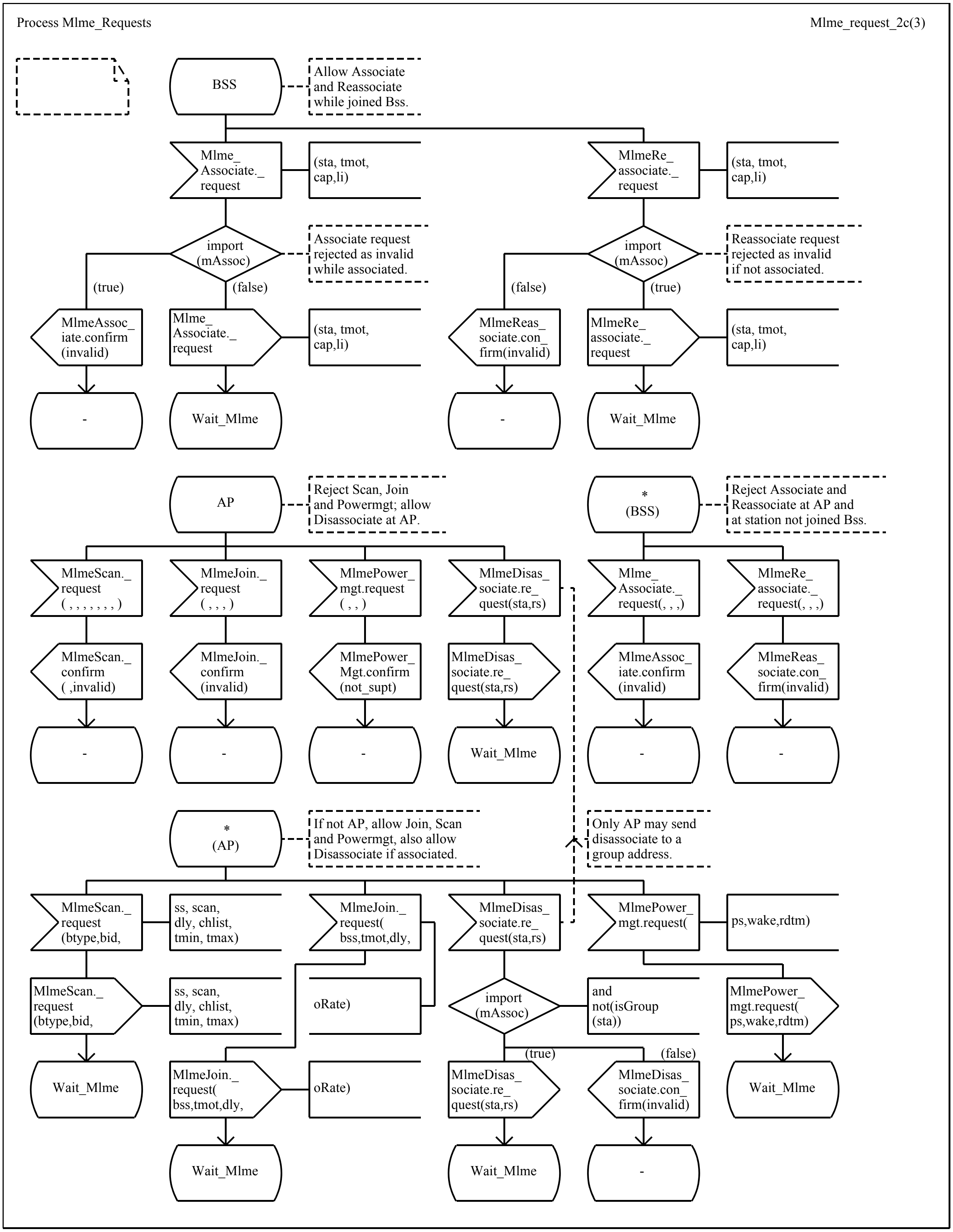
```
/* Import of {Read-Only} MIB counter
   values exported from other processes */
imported
  dot11AckFailureCount,
  dot11FailedCount,
  dot11FcsErrorCount,
  dot11FrameDuplicateCount,
  dot11MulticastReceivedFrameCount,
  dot11MulticastTransmittedFrameCount,
  dot11MultipleRetryCount,
  dot11ReceivedFragmentCount,
  dot11RetryCount,
  dot11RtsFailureCount,
  dot11RtsSuccessCount,
  dot11TransmittedFragmentCount,
  dot11WepExcludedCount,
  dot11WepIcvErrorCount,
  dot11WepUndecryptableCount  Counter32 ;
```

```
/* Declarations of MIB attributes exported from
    this process */

      /* Read-Write attributes */
dcl exported
   dot11AuthenticationAlgorithms  AuthTypeSet:=
     incl(open_system, shared_key),
  dot11ExcludeUnencrypted  Boolean:= false,
  dot11FragmentationThreshold  Integer:= 2346,
  dot11GroupAddresses  MacAddrSet:= empty,
  dot11LongRetryLimit  Integer:= 4,
  dot11MaxReceiveLifetime  Kusec:= 512,
  dot11MaxTransmitMsduLifetime  Kusec:= 512,
  dot11MediumOccupancyLimit  Kusec:= 100,
  dot11PrivacyInvoked  Boolean:= false,
  mReceiveDTIMs  Boolean:= true,
  dot11CfpPeriod  Integer:= 1,
  dot11CfpMaxDuration  Kusec:= 200,
  dot11AuthenticationResponseTimeout  Kusec:= 512,
  dot11RtsThreshold  Integer:= 3000,
  dot11ShortRetryLimit  Integer:= 7,
  dot11WepDefaultKeyId  KeyIndex:= 0,
  dot11CurrentChannelNumber  Integer:= 0,
  dot11CurrentSet  Integer:= 0,
  dot11CurrentPattern  Integer:= 0,
  dot11CurrentIndex  Integer:= 0 ;

      /* Write-Only attributes */
dcl exported
  dot11WepDefaultKeys  KeyVector:= nullKey,
  dot11WepKeyMappings
    KeyMapArray:= (. nullAddr, false, nullKey .) ;
```

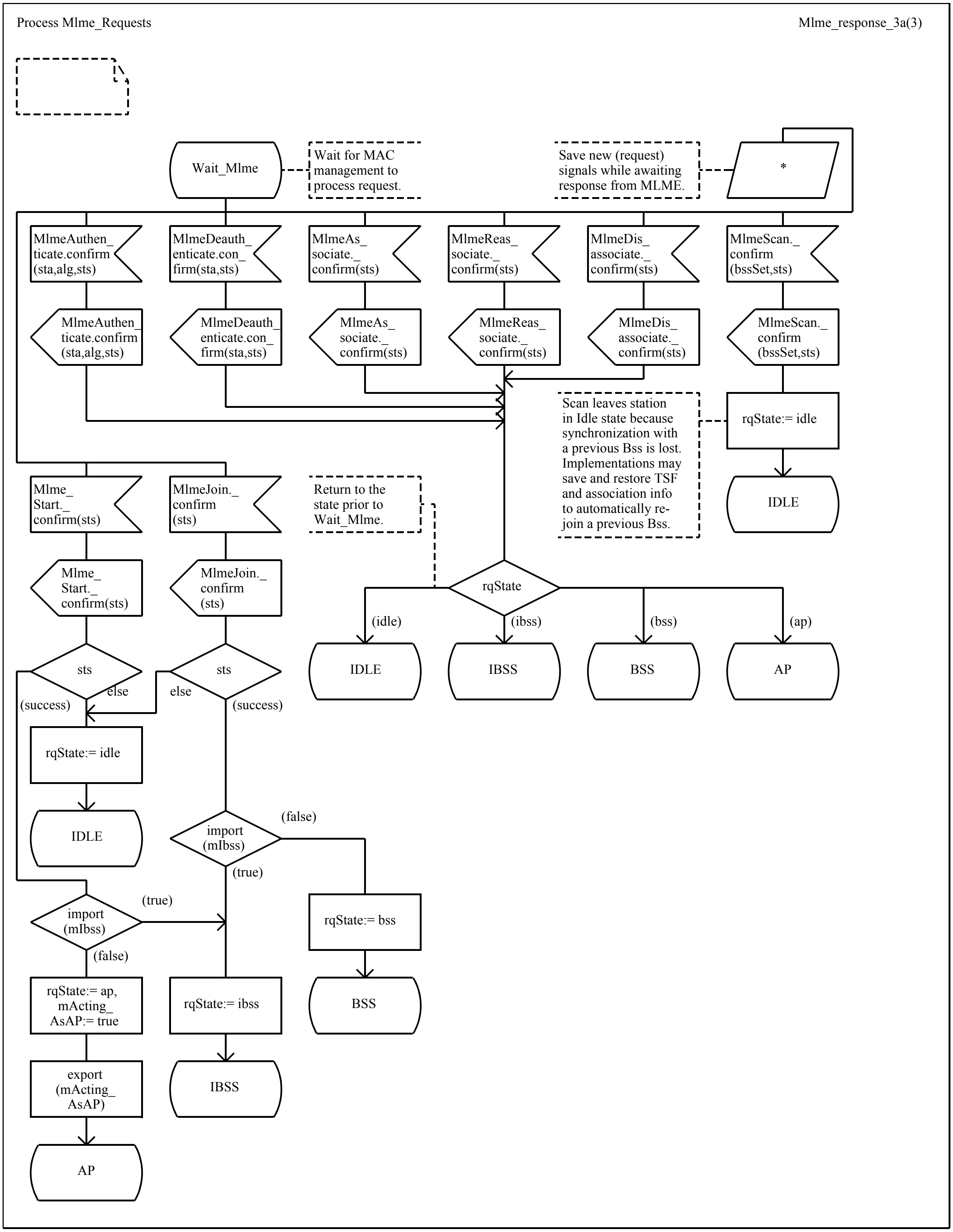
```
/* The following Read-Only attributes in the
   MAC MIB are defined as synonyms (named
   constants) rather than remote variables
   because they describe properties of the
   station which are static, at least during
   any single instance of MAC operation:
     dot11AuthenticationAlgorithms  AuthTypeSet,
     dot11CfPollable  Boolean,
     dot11MacAddress  MacAddr,
     dot11ManufacturerID  Octetstring,
     dot11PrivacyOptionImplemented  Boolean,
     dot11ProductID  Octetstring,
     aStationID  MacAddr,
     dot11WepKeyMappingLength  Integer ;

   In addition, all Read-Only attributes in the
   PHY MIB which are accessed by the MAC
   are defined as synonyms.
*/
```

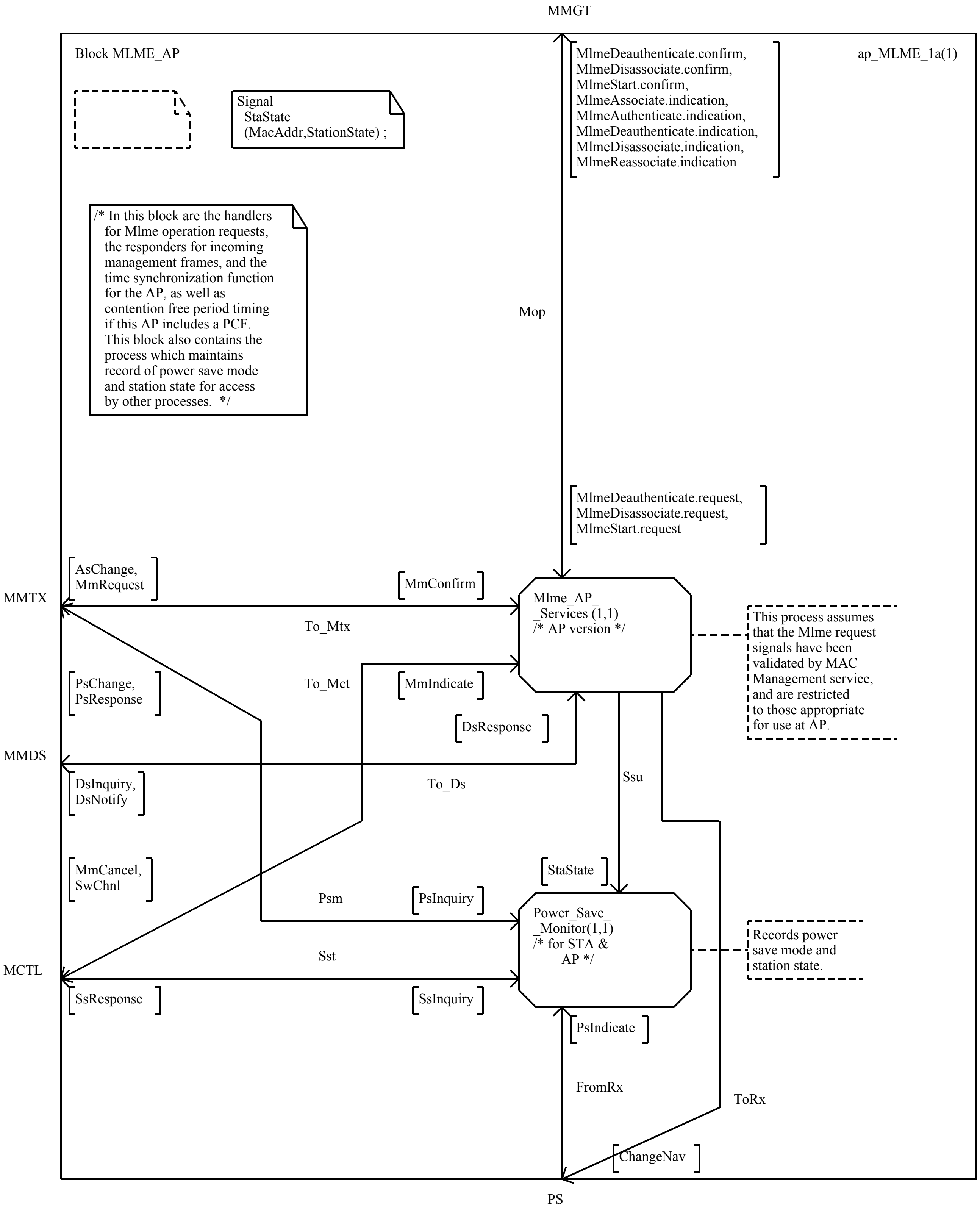
```
/* NOTE:
   The values listed for MAC MIB attributes are the
   specified default values for those attributes.
   The values listed for PHY MIB attributes are either
   the default values for the FH PHY, or arbitrary
   values within the specified range.  The specific
   values for PHY attributes in this SDL description
   of the MAC do not have normative significance.
*/
```

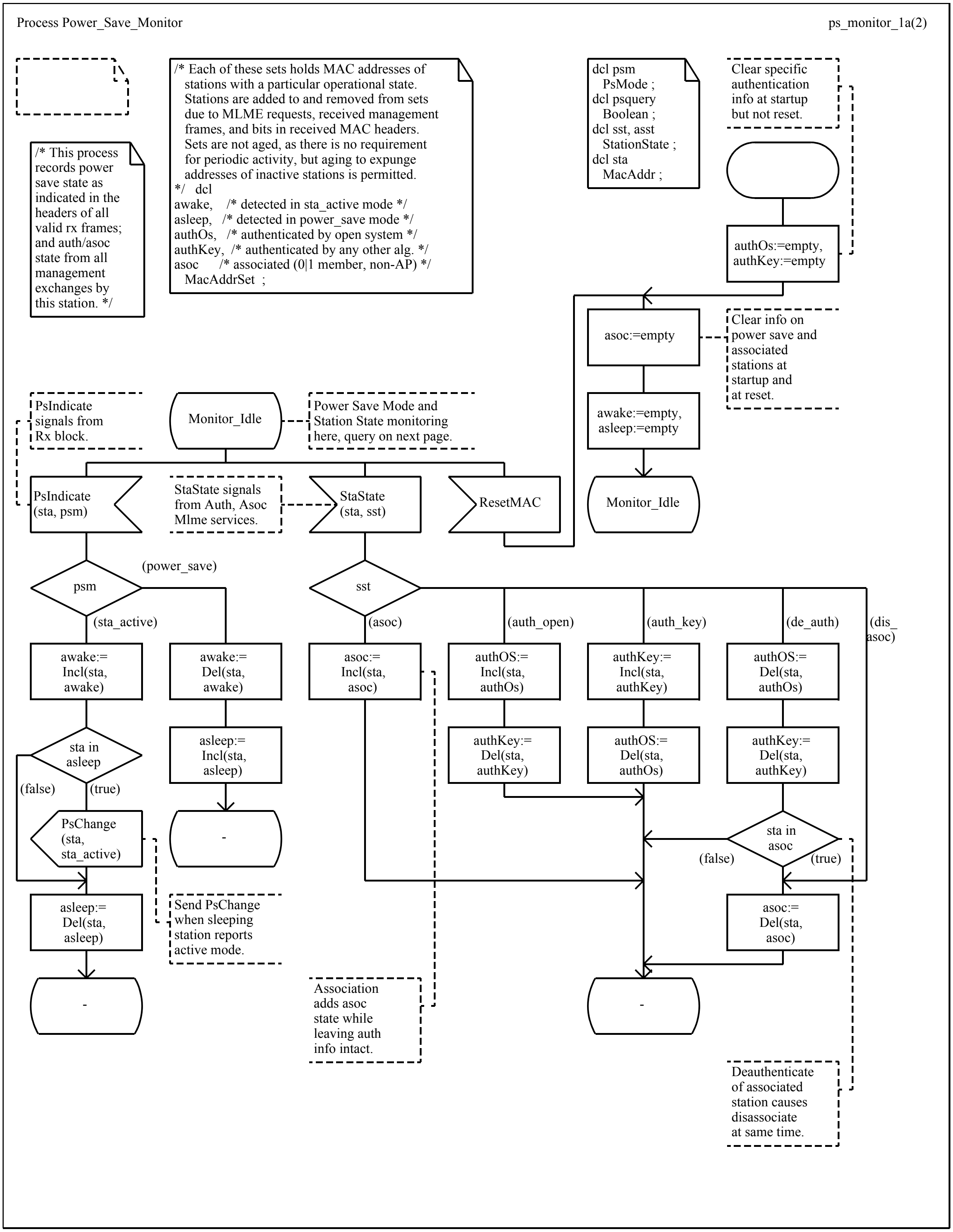

Process Mlme_Requests
Mlme_request_1b(3)
dcl exported mActingAsAp
Boolean:= false ;
imported mAssoc,
mIbss Boolean ;
newtype MRqState
literals idle, bss, ibss, ap ;
endnewtype MRqState ;
dcl rqState
MRqState:= idle ;
/* This process tracks the synchronization state of the station as Idle (not part of any Bss), Ibss (started or joined an independent Bss), Bss (joined an infrastructure Bss), or Ap (started an infrastructure Bss). Mlme operation requests invalid in the current state are rejected here, while valid requests are passed to the Mlme block for processing. This simplifies process flow and signal saving in the Mlme block, because only meaningful Mlme requests arrive for handling. */
dcl alg AuthType ;
dcl bRate, oRate, ss Octetstring ;
dcl bss BssDscr ;
dcl bssSet BssDscrSet ;
dcl btype BssType ;
dcl cap Capability ;
dcl cfpm CfParms ;
dcl chlist Intstring ;
dcl dtp, li Integer ;
dcl dly Usec ;
dcl ibpm IbssParms ;
dcl phpm PhyParms ;
dcl ps PwrSave ;
dcl rs ReasonCode ;
dcl scan ScanType ;
dcl sta, bid MacAddr ;
dcl sts MlmeStatus ;
dcl tBcn, tmax, tmin, tmot Kusec ;
dcl typeSet BssTypeSet ;
dcl wake, rdtm Boolean ;
re_
start
export
(mActing_
AsAP)
IDLE
Reject Authenticate,
allow Start if idle
Reject Start if
not idle, allow
Auth if neither
IDLE nor AP.
*
(IDLE, AP)
*
(IDLE)
Mlme_
Start._
request
(ss, btype, tBcn,
dtp, cfpm, phpm,
ibpm, dly, cap,
bRate, oRate)
MlmeAuth_
enticate.re_
quest(sta, ,)
Reject as invalid
due to not being
in a BSS.
MlmeStart._
request(, ,
, , , , , , , ,)
btype
MlmeAuth_
enticate._
confirm
(sta,
invalid)
MlmeAuth_
enticate._
request
(sta, alg,
tmot)
(independent)
(infrastructure)
sCanBeAp
(true)
(false)
MlmeAuth_
enticate._
request
(sta, alg,
tmot)
Mlme_
Start._
request
(ss, btype, tBcn,
dtp, cfpm, phpm,
ibpm, dly, cap,
bRate, oRate)
MlmeStart._
confirm
(invalid)
Wait_Mlme
MlmeStart._
confirm
(alreadyBss)
Wait_Mlme
-
*
Reset and
Deauthenticate
always allowed.
-
ResetMAC
MlmeDeau_
thenticate._
request(
sta,rs)
rqState:= idle,
mActing_
AsAp:= false
MlmeDeauth_
enticate._re_
quest(sta,rs)
Deauthenticate
expunges local
authentication
record even if
there is no BSS
for sending the
notification.
re_
start
Wait_Mlme

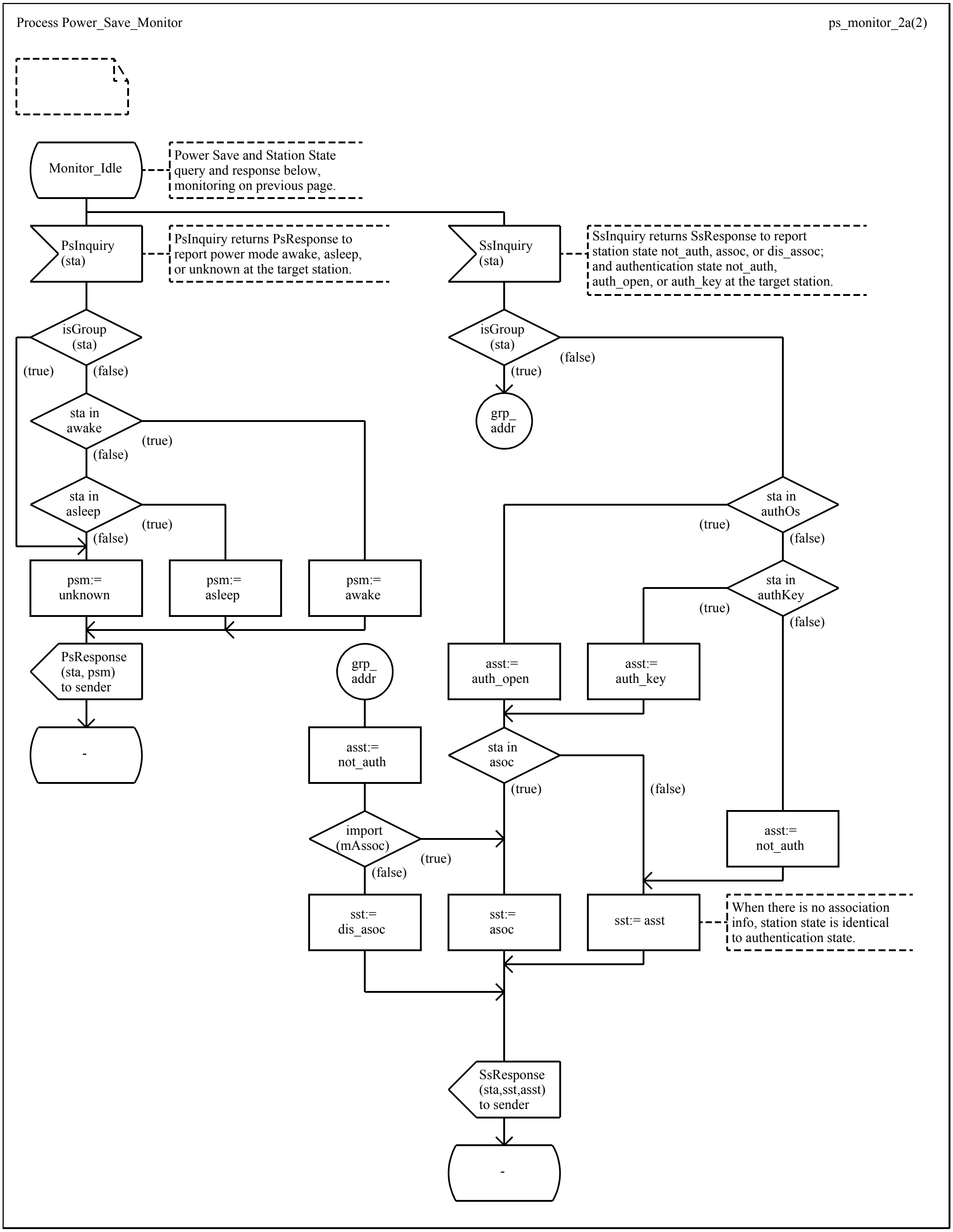

Process Mlme_Requests
Mlme_request_2c(3)
BSS
Allow Associate and Reassociate while joined Bss.
Mlme_ Associate._ request
(sta, tmot, cap,li)
MlmeRe_ associate._ request
(sta, tmot, cap,li)
import (mAssoc)
Associate request rejected as invalid while associated.
(true)
(false)
MlmeAssoc_ iate.confirm (invalid)
Mlme_ Associate._ request
(sta, tmot, cap,li)
-
Wait_Mlme
import (mAssoc)
Reassociate request rejected as invalid if not associated.
(false)
(true)
MlmeReas_ sociate.con_ firm(invalid)
MlmeRe_ associate._ request
(sta, tmot, cap,li)
-
Wait_Mlme
AP
Reject Scan, Join and Powermgt; allow Disassociate at AP.
*
(BSS)
Reject Associate and Reassociate at AP and at station not joined Bss.
MlmeScan._ request (, , , , , , ,)
MlmeJoin._ request (, , ,)
MlmePower_ mgt.request (, ,)
MlmeDisas_ sociate.re_ quest(sta,rs)
Mlme_ Associate._ request(, , ,)
MlmeRe_ associate._ request(, , ,)
MlmeScan._ confirm (,invalid)
MlmeJoin._ confirm (invalid)
MlmePower_ Mgt.confirm (not_supt)
MlmeDisas_ sociate.re_ quest(sta,rs)
MlmeAssoc_ iate.confirm (invalid)
MlmeReas_ sociate.con_ firm(invalid)
-
-
-
Wait_Mlme
-
-
*
(AP)
If not AP, allow Join, Scan and Powermgt, also allow Disassociate if associated.
Only AP may send disassociate to a group address.
MlmeScan._ request (btype,bid,
ss, scan, dly, chlist, tmin, tmax)
MlmeJoin._ request(bss,tmot,dly,
MlmeDisas_ sociate.re_ quest(sta,rs)
MlmePower_ mgt.request(
ps,wake,rdtm)
MlmeScan._ request (btype,bid,
ss, scan, dly, chlist, tmin, tmax)
oRate)
import (mAssoc)
and not(isGroup (sta))
MlmePower_ mgt.request(ps,wake,rdtm)
(true)
(false)
Wait_Mlme
MlmeJoin._ request(bss,tmot,dly,
oRate)
MlmeDisas_ sociate.re_ quest(sta,rs)
MlmeDisas_ sociate.con_ firm(invalid)
Wait_Mlme
Wait_Mlme
Wait_Mlme
Wait_Mlme
-

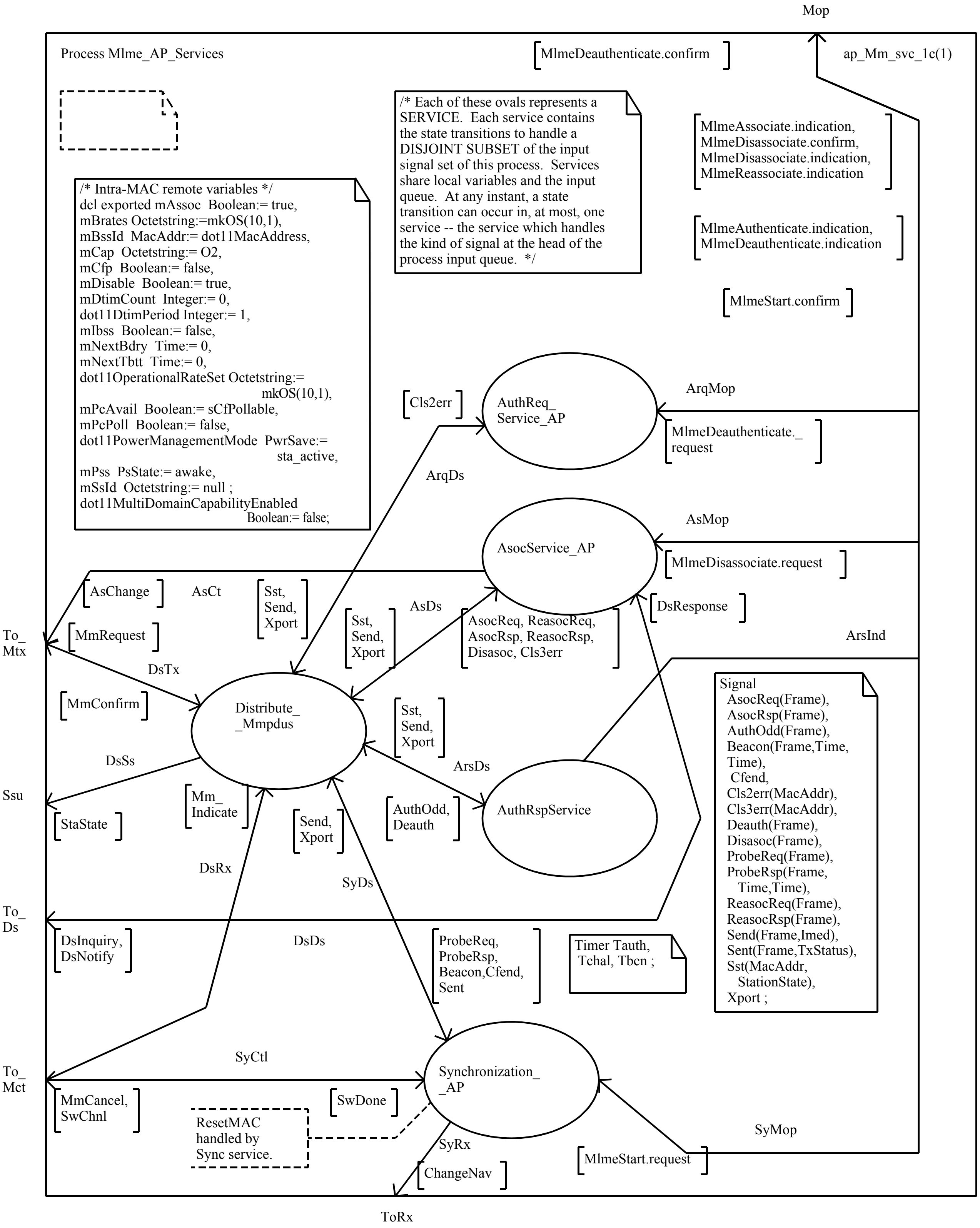
]


Process Mlme_Requests
Mlme_response_3a(3)
Wait_Mlme
Wait for MAC management to process request.
Save new (request) signals while awaiting response from MLME.
*
MlmeAuthen_ ticate.confirm (sta,alg,sts)
MlmeDeauth_ enticate.con_ firm(sta,sts)
MlmeAs_ sociate._ confirm(sts)
MlmeReas_ sociate._ confirm(sts)
MlmeDis_ associate._ confirm(sts)
MlmeScan._ confirm (bssSet,sts)
MlmeAuthen_ ticate.confirm (sta,alg,sts)
MlmeDeauth_ enticate.con_ firm(sta,sts)
MlmeAs_ sociate._ confirm(sts)
MlmeReas_ sociate._ confirm(sts)
MlmeDis_ associate._ confirm(sts)
MlmeScan._ confirm (bssSet,sts)
Scan leaves station in Idle state because synchronization with a previous Bss is lost. Implementations may save and restore TSF and association info to automatically re-join a previous Bss.
rqState:= idle
IDLE
Mlme_ Start._ confirm(sts)
MlmeJoin._ confirm (sts)
Return to the state prior to Wait_Mlme.
rqState
(idle)
(ibss)
(bss)
(ap)
IDLE
IBSS
BSS
AP
Mlme_ Start._ confirm(sts)
MlmeJoin._ confirm (sts)
sts
else
(success)
sts
else
(success)
rqState:= idle
IDLE
import (mIbss)
(false)
(true)
rqState:= bss
BSS
import (mIbss)
(true)
(false)
rqState:= ap, mActing_ AsAP:= true
rqState:= ibss
export (mActing_ AsAP)
IBSS
AP


MMGT
Block MLME_AP
ap_MLME_1a(1)
Signal
StaState
(MacAddr,StationState) ;
MlmeDeauthenticate.confirm,
MlmeDisassociate.confirm,
MlmeStart.confirm,
MlmeAssociate.indication,
MlmeAuthenticate.indication,
MlmeDeauthenticate.indication,
MlmeDisassociate.indication,
MlmeReassociate.indication
/* In this block are the handlers for Mlme operation requests, the responders for incoming management frames, and the time synchronization function for the AP, as well as contention free period timing if this AP includes a PCF. This block also contains the process which maintains record of power save mode and station state for access by other processes. */
Mop
MlmeDeauthenticate.request,
MlmeDisassociate.request,
MlmeStart.request
AsChange,
MmRequest
MmConfirm
MMTX
Mlme_AP_
_Services (1,1)
/* AP version */
To_Mtx
This process assumes that the Mlme request signals have been validated by MAC Management service, and are restricted to those appropriate for use at AP.
PsChange,
PsResponse
To_Mct
MmIndicate
DsResponse
MMDS
DsInquiry,
DsNotify
To_Ds
Ssu
MmCancel,
SwChnl
StaState
Psm
PsInquiry
Power_Save_
_Monitor(1,1)
/* for STA &
AP */
Records power save mode and station state.
Sst
MCTL
SsResponse
SsInquiry
PsIndicate
FromRx
ToRx
ChangeNav
PS


Process Power_Save_Monitor
ps_monitor_1a(2)
/* This process records power save state as indicated in the headers of all valid rx frames; and auth/asoc state from all management exchanges by this station. */
/* Each of these sets holds MAC addresses of stations with a particular operational state. Stations are added to and removed from sets due to MLME requests, received management frames, and bits in received MAC headers. Sets are not aged, as there is no requirement for periodic activity, but aging to expunge addresses of inactive stations is permitted.
*/ dcl
awake, /* detected in sta_active mode */
asleep, /* detected in power_save mode */
authOs, /* authenticated by open system */
authKey, /* authenticated by any other alg. */
asoc /* associated (0|1 member, non-AP) */
MacAddrSet ;
dcl psm PsMode ;
dcl psquery Boolean ;
dcl sst, asst StationState ;
dcl sta MacAddr ;
Clear specific authentication info at startup but not reset.
authOs:=empty, authKey:=empty
asoc:=empty
Clear info on power save and associated stations at startup and at reset.
awake:=empty, asleep:=empty
Monitor_Idle
PsIndicate signals from Rx block.
Monitor_Idle
Power Save Mode and Station State monitoring here, query on next page.
PsIndicate (sta, psm)
StaState signals from Auth, Asoc Mlme services.
StaState (sta, sst)
ResetMAC
psm
(power_save)
(sta_active)
awake:= Incl(sta, awake)
awake:= Del(sta, awake)
sta in asleep
(false)
(true)
asleep:= Incl(sta, asleep)
-
PsChange (sta, sta_active)
Send PsChange when sleeping station reports active mode.
asleep:= Del(sta, asleep)
-
sst
(asoc)
(auth_open)
(auth_key)
(de_auth)
(dis_ asoc)
asoc:= Incl(sta, asoc)
authOS:= Incl(sta, authOs)
authKey:= Incl(sta, authKey)
authOS:= Del(sta, authOs)
authKey:= Del(sta, authKey)
authOS:= Del(sta, authOs)
authKey:= Del(sta, authKey)
sta in asoc
(false)
(true)
asoc:= Del(sta, asoc)
-
Association adds asoc state while leaving auth info intact.
Deauthenticate of associated station causes disassociate at same time.


Process Power_Save_Monitor
ps_monitor_2a(2)
Monitor_Idle
Power Save and Station State query and response below, monitoring on previous page.
PsInquiry (sta)
PsInquiry returns PsResponse to report power mode awake, asleep, or unknown at the target station.
SsInquiry (sta)
SsInquiry returns SsResponse to report station state not_auth, assoc, or dis_assoc; and authentication state not_auth, auth_open, or auth_key at the target station.
isGroup (sta)
(true)
(false)
sta in awake
(true)
(false)
sta in asleep
(true)
(false)
psm:= unknown
psm:= asleep
psm:= awake
PsResponse (sta, psm) to sender
-
isGroup (sta)
(false)
(true)
grp_ addr
sta in authOs
(true)
(false)
sta in authKey
(true)
(false)
asst:= auth_open
asst:= auth_key
grp_ addr
asst:= not_auth
sta in asoc
(true)
(false)
import (mAssoc)
(true)
(false)
asst:= not_auth
sst:= dis_asoc
sst:= asoc
sst:= asst
When there is no association info, station state is identical to authentication state.
SsResponse (sta,sst,asst) to sender
-



Process Mlme_AP_Services

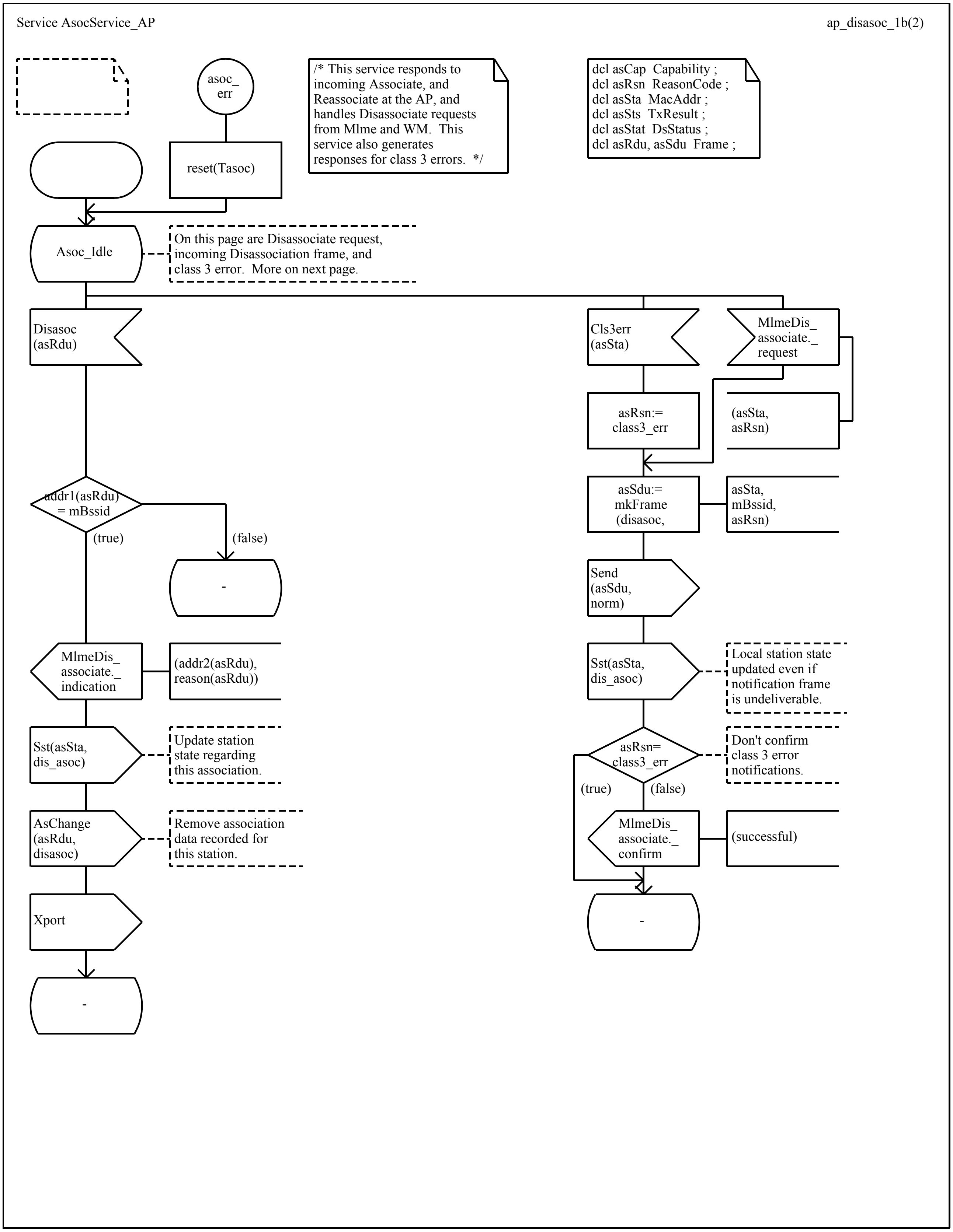
```
/* Intra-MAC remote variables */
dcl exported mAssoc  Boolean:= true,
mBrates Octetstring:=mkOS(10,1),
mBssId  MacAddr:= dot11MacAddress,
mCap  Octetstring:= O2,
mCfp  Boolean:= false,
mDisable  Boolean:= true,
mDtimCount  Integer:= 0,
dot11DtimPeriod Integer:= 1,
mIbss  Boolean:= false,
mNextBdry  Time:= 0,
mNextTbtt  Time:= 0,
dot11OperationalRateSet Octetstring:=
                                  mkOS(10,1),
mPcAvail  Boolean:= sCfPollable,
mPcPoll  Boolean:= false,
dot11PowerManagementMode  PwrSave:=
                                   sta_active,
mPss  PsState:= awake,
mSsId  Octetstring:= null ;
dot11MultiDomainCapabilityEnabled
                                Boolean:= false;
```

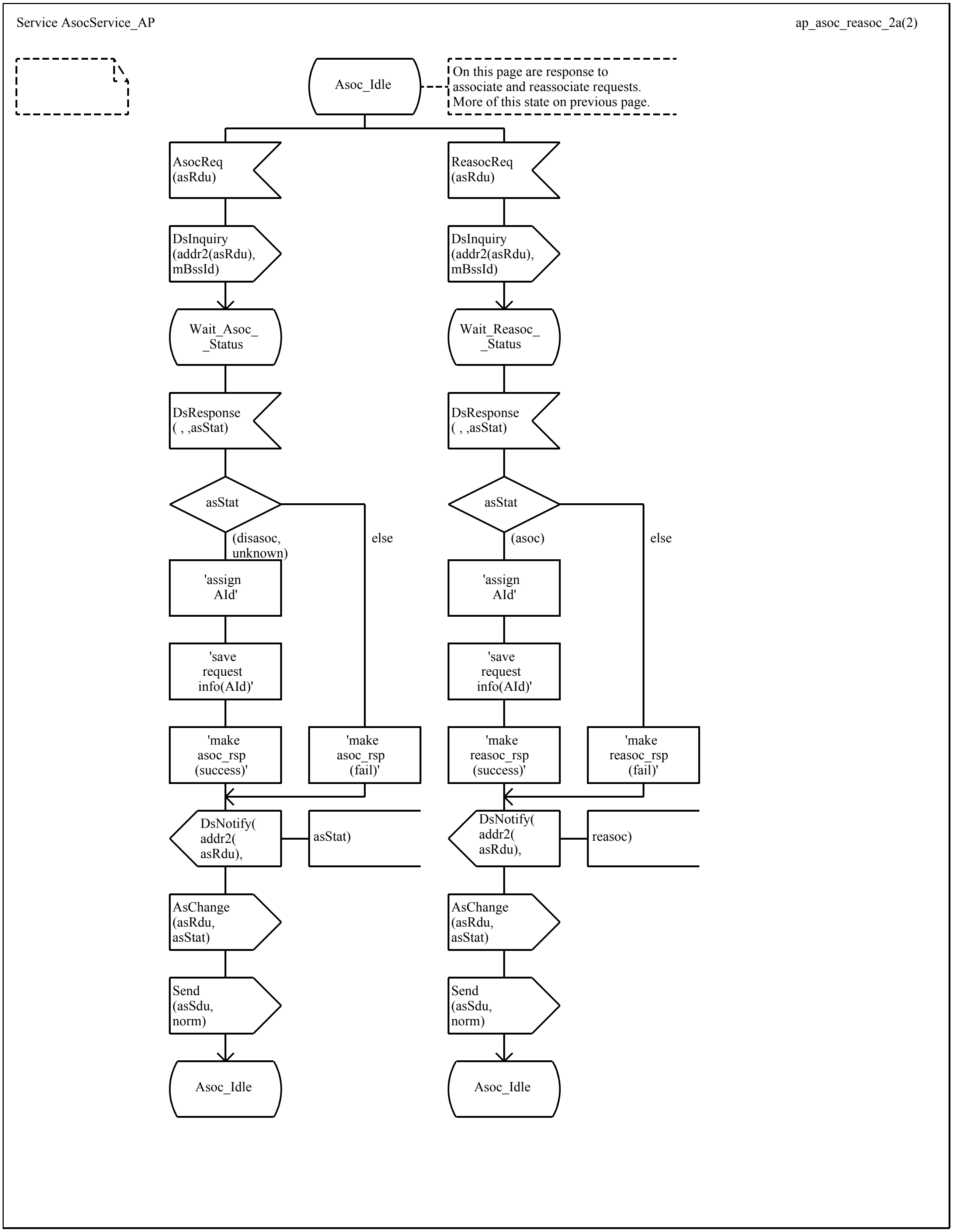
/* Each of these ovals represents a SERVICE. Each service contains the state transitions to handle a DISJOINT SUBSET of the input signal set of this process. Services share local variables and the input queue. At any instant, a state transition can occur in, at most, one service -- the service which handles the kind of signal at the head of the process input queue. */

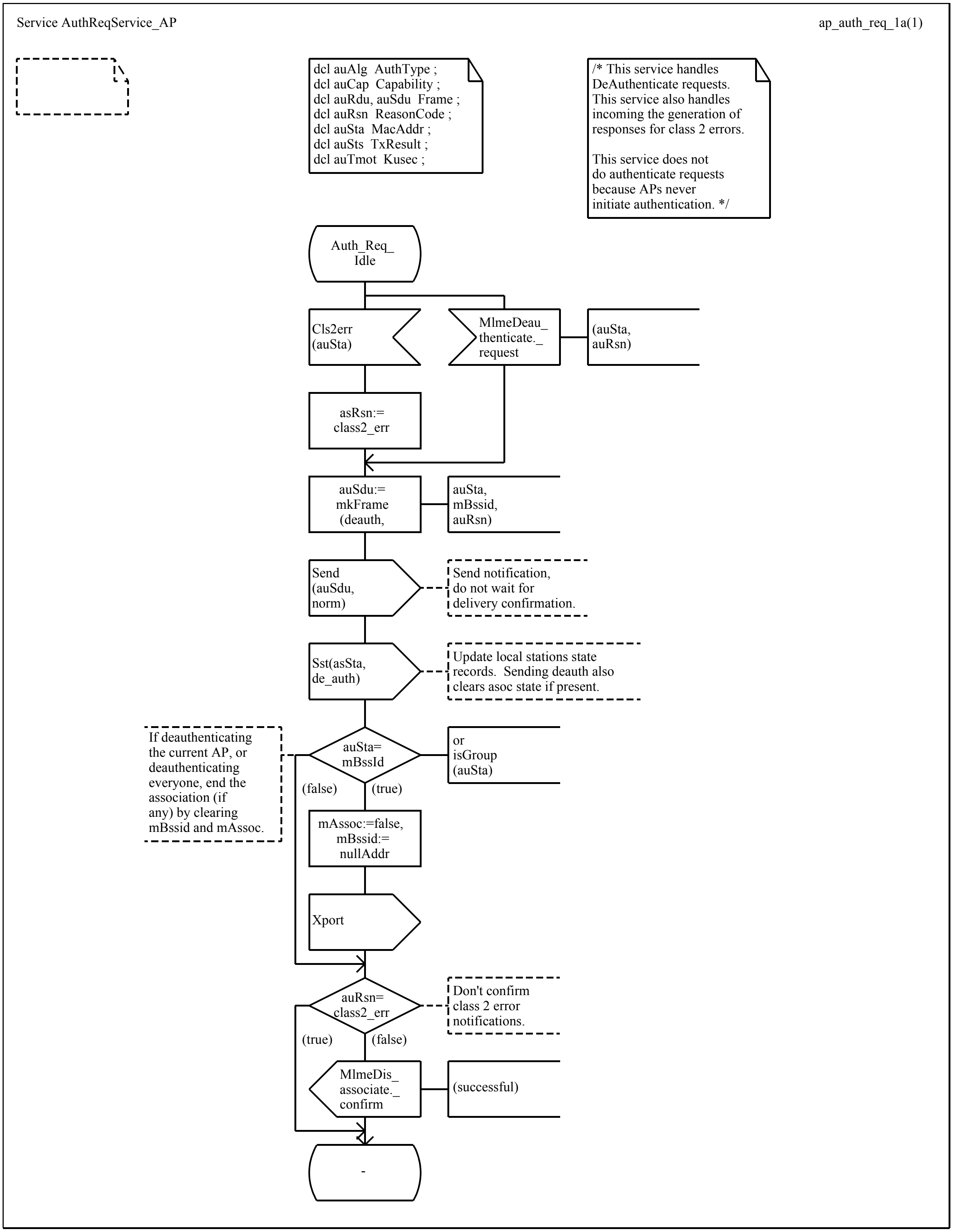
```
Signal
  AsocReq(Frame),
  AsocRsp(Frame),
  AuthOdd(Frame),
  Beacon(Frame,Time,
  Time),
   Cfend,
  Cls2err(MacAddr),
  Cls3err(MacAddr),
  Deauth(Frame),
  Disasoc(Frame),
  ProbeReq(Frame),
  ProbeRsp(Frame,
     Time,Time),
  ReasocReq(Frame),
  ReasocRsp(Frame),
  Send(Frame,Imed),
  Sent(Frame,TxStatus),
  Sst(MacAddr,
     StationState),
  Xport ;
```

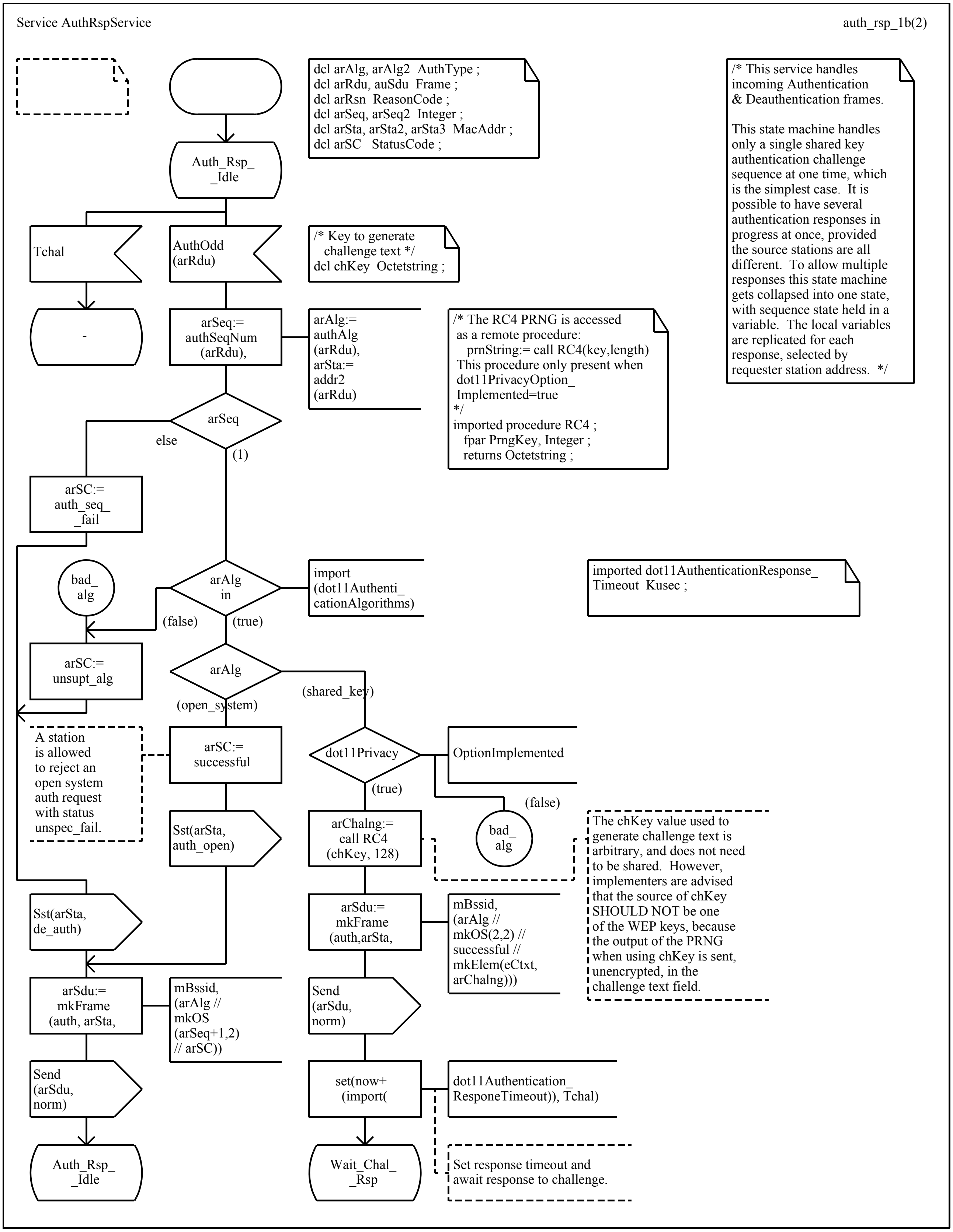
```
Timer Tauth,
 Tchal, Tbcn ;
```

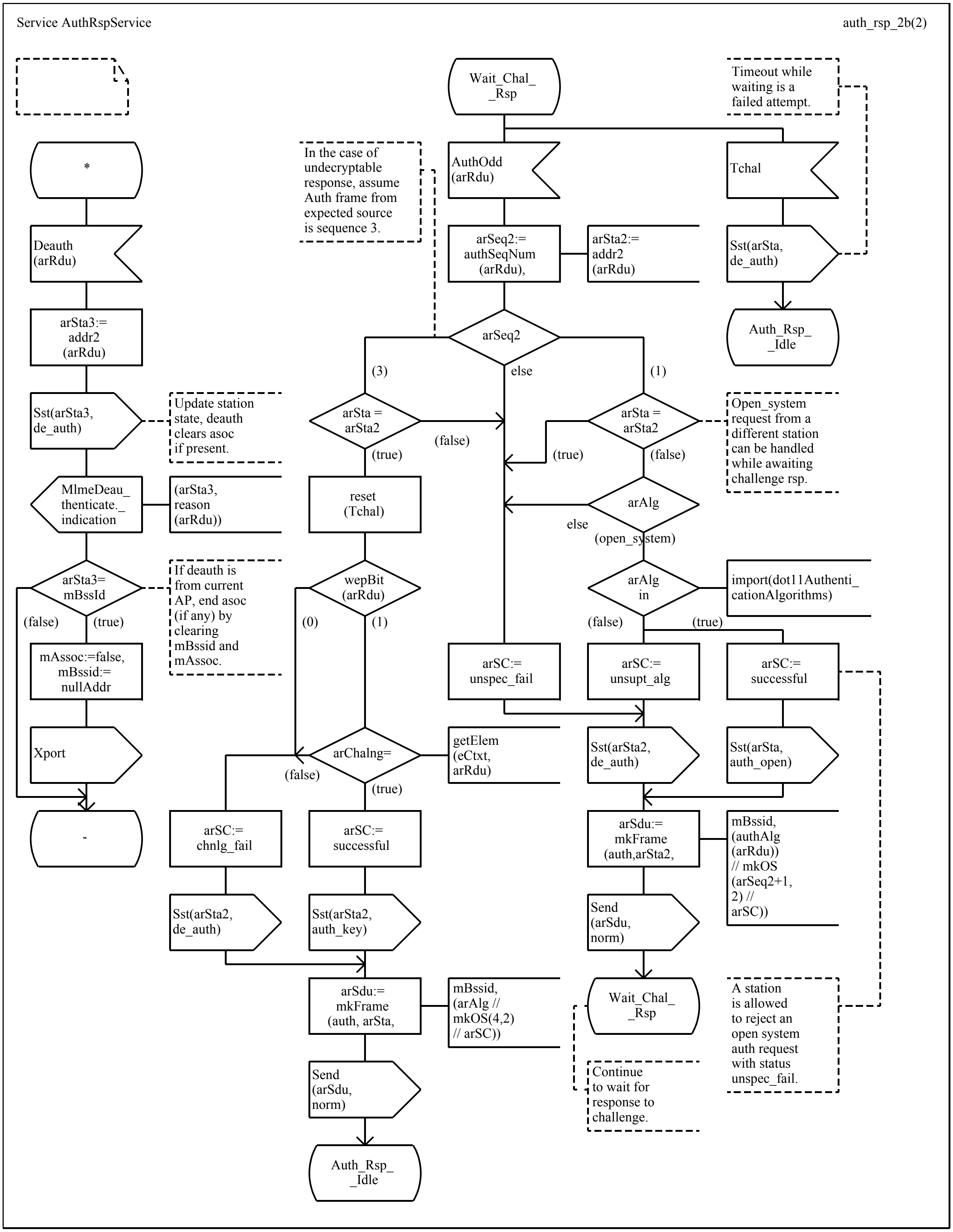
ResetMAC handled by Sync service.

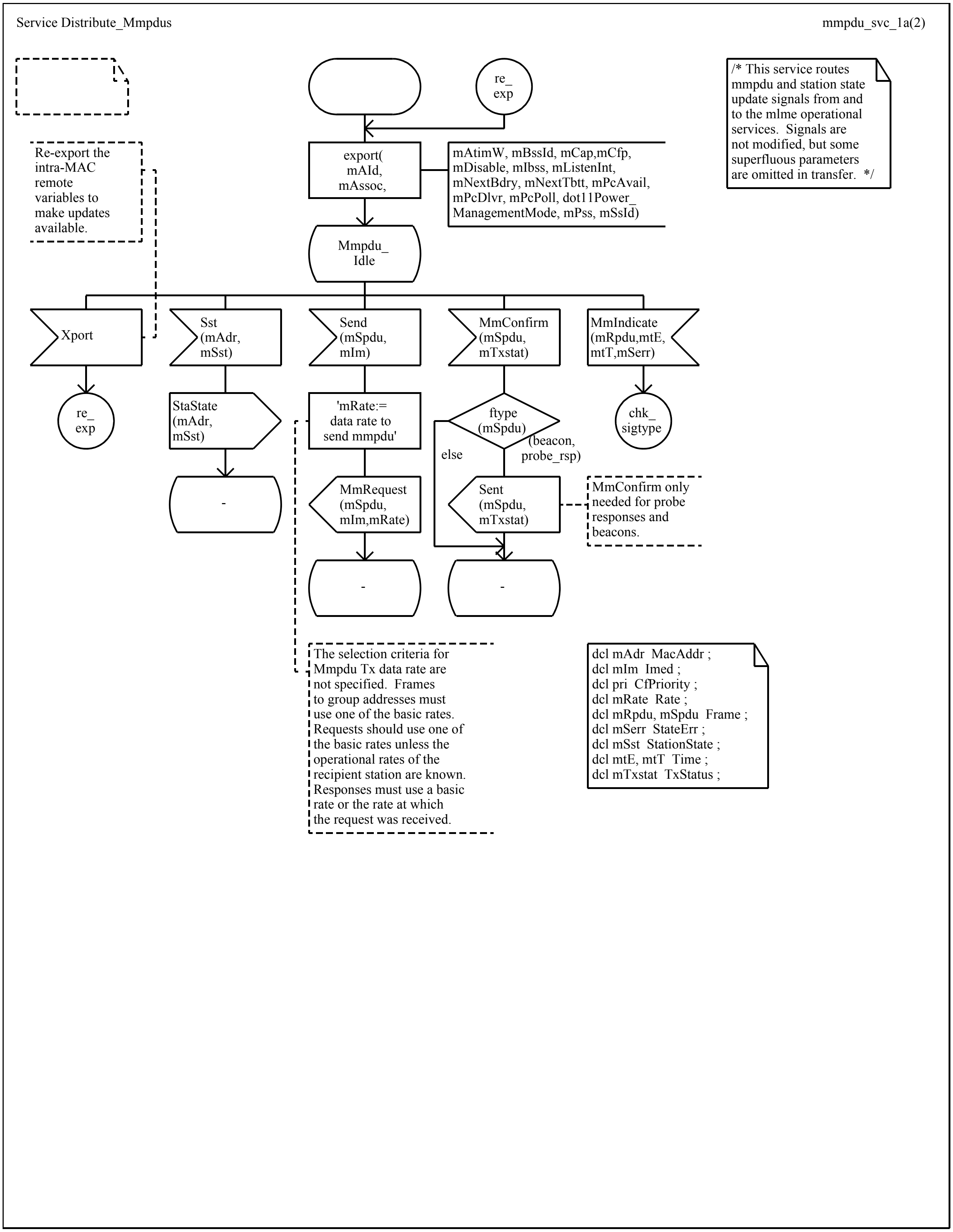

Service AsocService_AP
ap_disasoc_1b(2)
asoc_
err
reset(Tasoc)
/* This service responds to
incoming Associate, and
Reassociate at the AP, and
handles Disassociate requests
from Mlme and WM. This
service also generates
responses for class 3 errors. */
dcl asCap Capability ;
dcl asRsn ReasonCode ;
dcl asSta MacAddr ;
dcl asSts TxResult ;
dcl asStat DsStatus ;
dcl asRdu, asSdu Frame ;
Asoc_Idle
On this page are Disassociate request,
incoming Disassociation frame, and
class 3 error. More on next page.
Disasoc
(asRdu)
addr1(asRdu)
= mBssid
(true)
(false)
-
MlmeDis_
associate._
indication
(addr2(asRdu),
reason(asRdu))
Sst(asSta,
dis_asoc)
Update station
state regarding
this association.
AsChange
(asRdu,
disasoc)
Remove association
data recorded for
this station.
Xport
-
Cls3err
(asSta)
MlmeDis_
associate._
request
asRsn:=
class3_err
(asSta,
asRsn)
asSdu:=
mkFrame
(disasoc,
asSta,
mBssid,
asRsn)
Send
(asSdu,
norm)
Sst(asSta,
dis_asoc)
Local station state
updated even if
notification frame
is undeliverable.
asRsn=
class3_err
Don't confirm
class 3 error
notifications.
(true)
(false)
MlmeDis_
associate._
confirm
(successful)
-


Service AsocService_AP
ap_asoc_reasoc_2a(2)
Asoc_Idle
On this page are response to
associate and reassociate requests.
More of this state on previous page.
AsocReq
(asRdu)
ReasocReq
(asRdu)
DsInquiry
(addr2(asRdu),
mBssId)
DsInquiry
(addr2(asRdu),
mBssId)
Wait_Asoc_
_Status
Wait_Reasoc_
_Status
DsResponse
(, ,asStat)
DsResponse
(, ,asStat)
asStat
asStat
(disasoc,
unknown)
else
(asoc)
else
'assign
AId'
'assign
AId'
'save
request
info(AId)'
'save
request
info(AId)'
'make
asoc_rsp
(success)'
'make
asoc_rsp
(fail)'
'make
reasoc_rsp
(success)'
'make
reasoc_rsp
(fail)'
DsNotify(
addr2(
asRdu),
asStat)
DsNotify(
addr2(
asRdu),
reasoc)
AsChange
(asRdu,
asStat)
AsChange
(asRdu,
asStat)
Send
(asSdu,
norm)
Send
(asSdu,
norm)
Asoc_Idle
Asoc_Idle


Service AuthReqService_AP
ap_auth_req_1a(1)
dcl auAlg AuthType ;
dcl auCap Capability ;
dcl auRdu, auSdu Frame ;
dcl auRsn ReasonCode ;
dcl auSta MacAddr ;
dcl auSts TxResult ;
dcl auTmot Kusec ;
/* This service handles DeAuthenticate requests. This service also handles incoming the generation of responses for class 2 errors.
This service does not do authenticate requests because APs never initiate authentication. */
Auth_Req_ Idle
Cls2err (auSta)
MlmeDeau_ thenticate._ request
(auSta, auRsn)
asRsn:= class2_err
auSdu:= mkFrame (deauth,
auSta, mBssid, auRsn)
Send (auSdu, norm)
Send notification, do not wait for delivery confirmation.
Sst(asSta, de_auth)
Update local stations state records. Sending deauth also clears asoc state if present.
If deauthenticating the current AP, or deauthenticating everyone, end the association (if any) by clearing mBssid and mAssoc.
auSta= mBssId
or isGroup (auSta)
(false)
(true)
mAssoc:=false, mBssid:= nullAddr
Xport
auRsn= class2_err
Don't confirm class 2 error notifications.
(true)
(false)
MlmeDis_ associate._ confirm
(successful)
-


Service AuthRspService
auth_rsp_1b(2)
dcl arAlg, arAlg2 AuthType ;
dcl arRdu, auSdu Frame ;
dcl arRsn ReasonCode ;
dcl arSeq, arSeq2 Integer ;
dcl arSta, arSta2, arSta3 MacAddr ;
dcl arSC StatusCode ;
/* This service handles incoming Authentication & Deauthentication frames.
This state machine handles only a single shared key authentication challenge sequence at one time, which is the simplest case. It is possible to have several authentication responses in progress at once, provided the source stations are all different. To allow multiple responses this state machine gets collapsed into one state, with sequence state held in a variable. The local variables are replicated for each response, selected by requester station address. */
Auth_Rsp_ _Idle
Tchal
AuthOdd (arRdu)
/* Key to generate challenge text */
dcl chKey Octetstring ;
-
arSeq:= authSeqNum (arRdu),
arAlg:= authAlg (arRdu), arSta:= addr2 (arRdu)
/* The RC4 PRNG is accessed as a remote procedure:
prnString:= call RC4(key,length)
This procedure only present when dot11PrivacyOption_ Implemented=true
*/
imported procedure RC4 ;
fpar PrngKey, Integer ;
returns Octetstring ;
arSeq
else
(1)
arSC:= auth_seq_ _fail
bad_ alg
arAlg in
import (dot11Authenti_ cationAlgorithms)
imported dot11AuthenticationResponse_ Timeout Kusec ;
(false)
(true)
arSC:= unsupt_alg
arAlg
(shared_key)
(open_system)
A station is allowed to reject an open system auth request with status unspec_fail.
arSC:= successful
dot11Privacy
OptionImplemented
(true)
(false)
Sst(arSta, auth_open)
arChalng:= call RC4 (chKey, 128)
bad_ alg
The chKey value used to generate challenge text is arbitrary, and does not need to be shared. However, implementers are advised that the source of chKey SHOULD NOT be one of the WEP keys, because the output of the PRNG when using chKey is sent, unencrypted, in the challenge text field.
Sst(arSta, de_auth)
arSdu:= mkFrame (auth,arSta,
mBssid, (arAlg // mkOS(2,2) // successful // mkElem(eCtxt, arChalng)))
arSdu:= mkFrame (auth, arSta,
mBssid, (arAlg // mkOS (arSeq+1,2) // arSC))
Send (arSdu, norm)
Send (arSdu, norm)
set(now+ (import(
dot11Authentication_ ResponeTimeout)), Tchal)
Auth_Rsp_ _Idle
Wait_Chal_ _Rsp
Set response timeout and await response to challenge.


Service AuthRspService
auth_rsp_2b(2)
*
Deauth (arRdu)
arSta3:= addr2 (arRdu)
Sst(arSta3, de_auth)
Update station state, deauth clears asoc if present.
MlmeDeau_ thenticate._ indication
(arSta3, reason (arRdu))
arSta3= mBssId
If deauth is from current AP, end asoc (if any) by clearing mBssid and mAssoc.
(false)
(true)
mAssoc:=false, mBssid:= nullAddr
Xport
-
Wait_Chal_ _Rsp
Timeout while waiting is a failed attempt.
AuthOdd (arRdu)
Tchal
In the case of undecryptable response, assume Auth frame from expected source is sequence 3.
arSeq2:= authSeqNum (arRdu),
arSta2:= addr2 (arRdu)
Sst(arSta, de_auth)
Auth_Rsp_ _Idle
arSeq2
(3)
else
(1)
arSta = arSta2
(false)
(true)
arSta = arSta2
(true)
(false)
Open_system request from a different station can be handled while awaiting challenge rsp.
reset (Tchal)
arAlg
else
(open_system)
wepBit (arRdu)
(0)
(1)
arAlg in
import(dot11Authenti_ cationAlgorithms)
(false)
(true)
arSC:= unspec_fail
arSC:= unsupt_alg
arSC:= successful
arChalng=
getElem (eCtxt, arRdu)
(false)
(true)
Sst(arSta2, de_auth)
Sst(arSta, auth_open)
arSC:= chnlg_fail
arSC:= successful
arSdu:= mkFrame (auth,arSta2,
mBssid, (authAlg (arRdu)) // mkOS (arSeq2+1, 2) // arSC))
Sst(arSta2, de_auth)
Sst(arSta2, auth_key)
Send (arSdu, norm)
arSdu:= mkFrame (auth, arSta,
mBssid, (arAlg // mkOS(4,2) // arSC))
Wait_Chal_ _Rsp
A station is allowed to reject an open system auth request with status unspec_fail.
Continue to wait for response to challenge.
Send (arSdu, norm)
Auth_Rsp_ _Idle



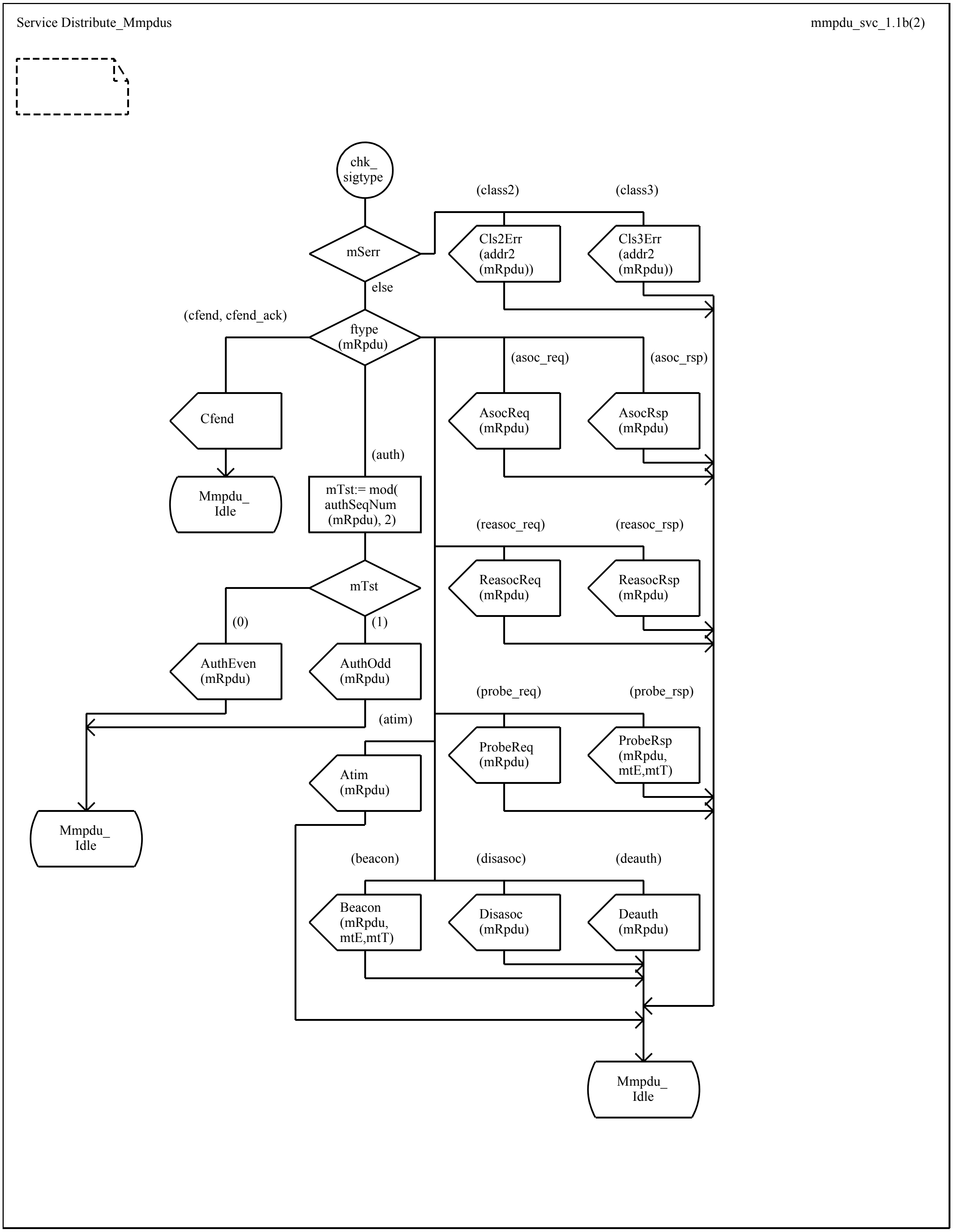
/* This service routes mmpdu and station state update signals from and to the mlme operational services. Signals are not modified, but some superfluous parameters are omitted in transfer. */

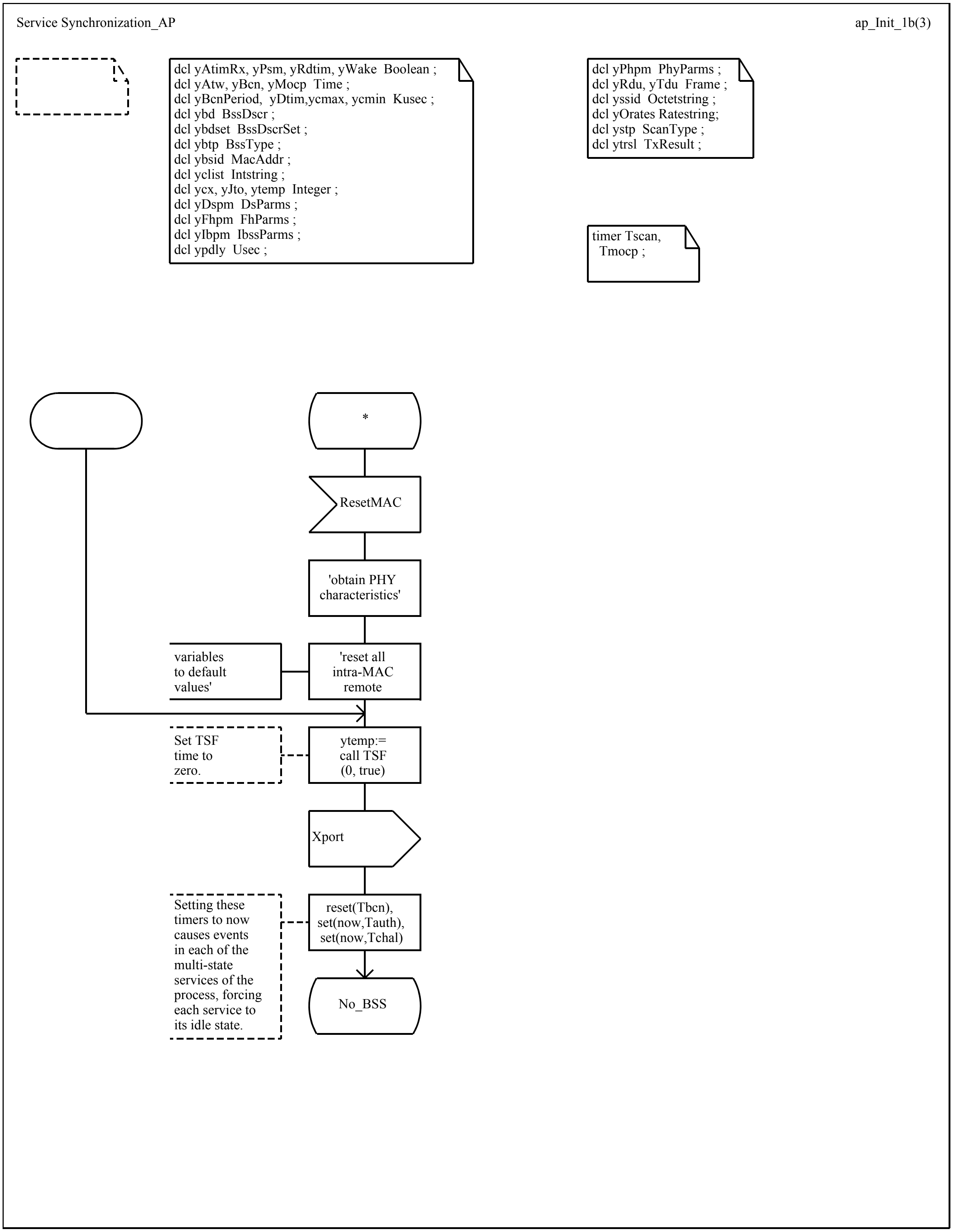
Re-export the intra-MAC remote variables to make updates available.

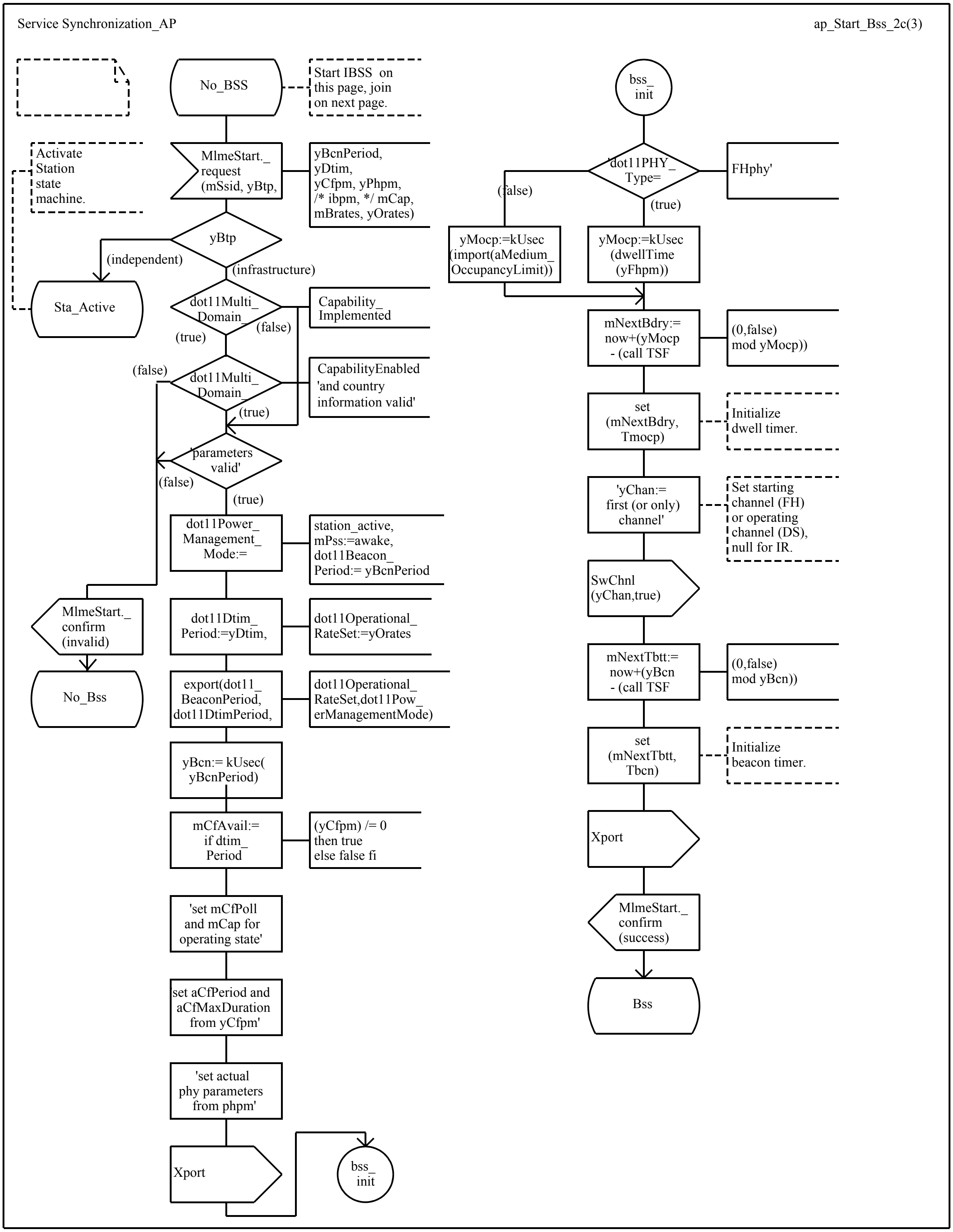
MmConfirm only needed for probe responses and beacons.

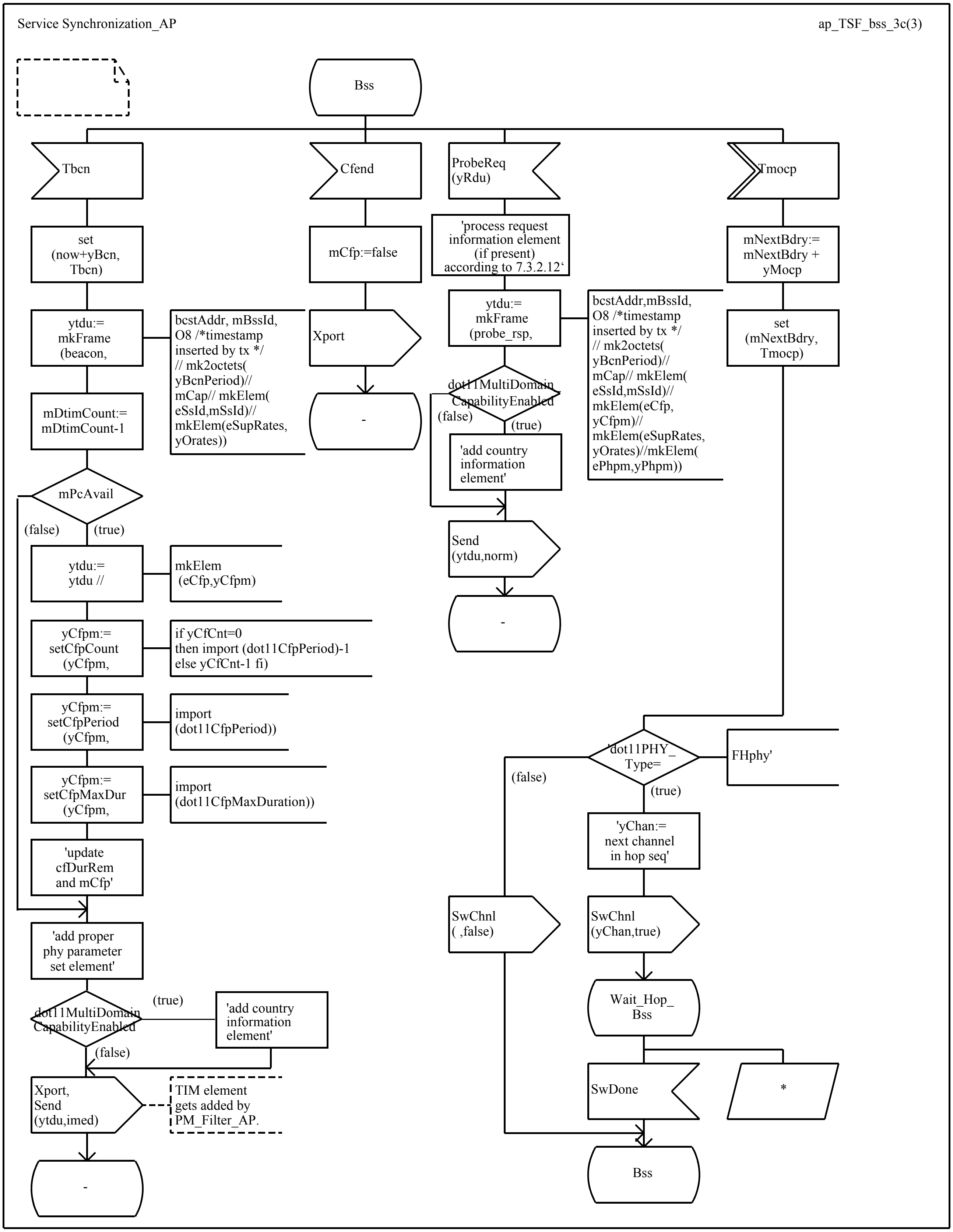
The selection criteria for Mmpdu Tx data rate are not specified. Frames to group addresses must use one of the basic rates. Requests should use one of the basic rates unless the operational rates of the recipient station are known. Responses must use a basic rate or the rate at which the request was received.

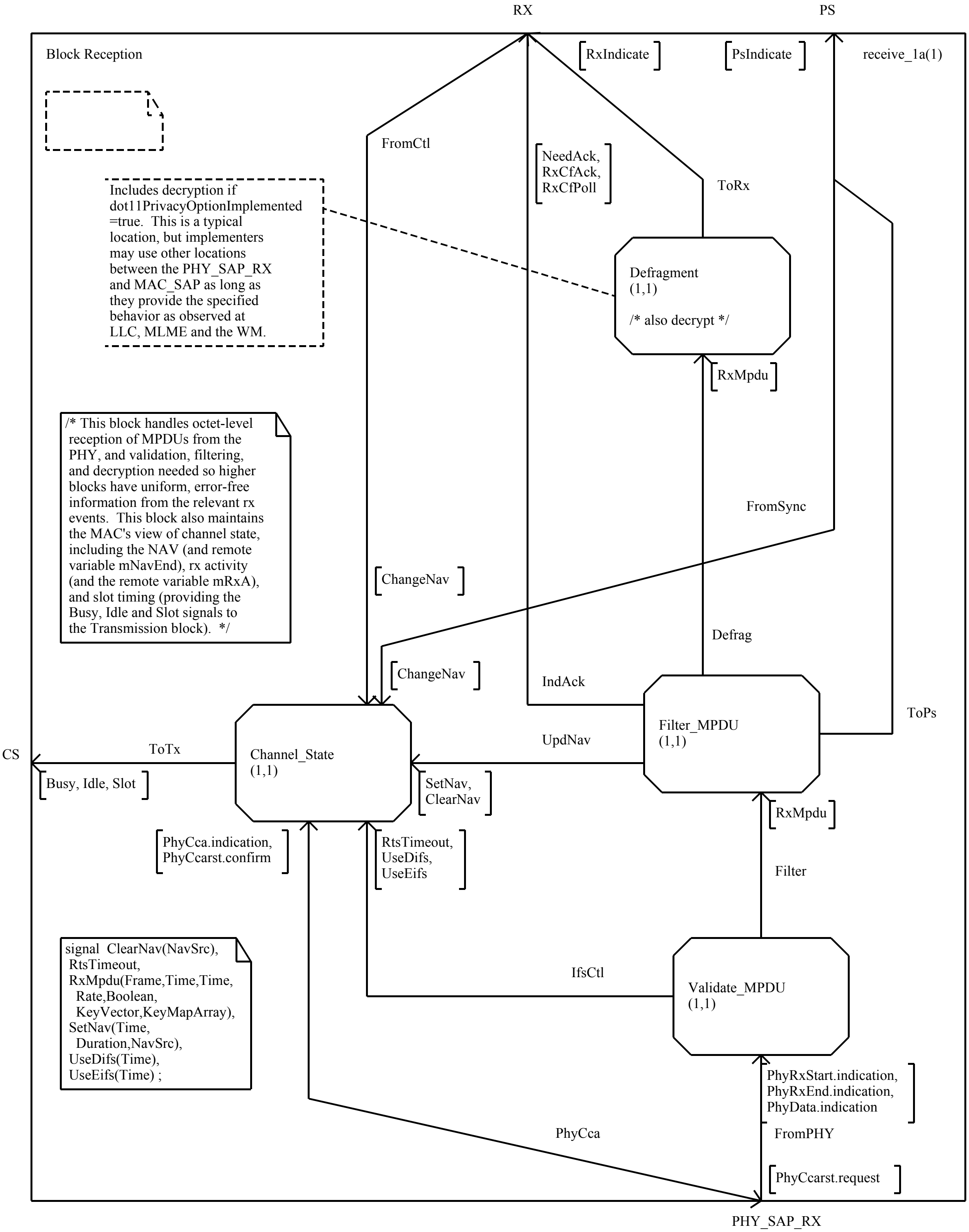
```
dcl mAdr  MacAddr ;
dcl mIm  Imed ;
dcl pri  CfPriority ;
dcl mRate  Rate ;
dcl mRpdu, mSpdu  Frame ;
dcl mSerr  StateErr ;
dcl mSst  StationState ;
dcl mtE, mtT  Time ;
dcl mTxstat  TxStatus ;
```

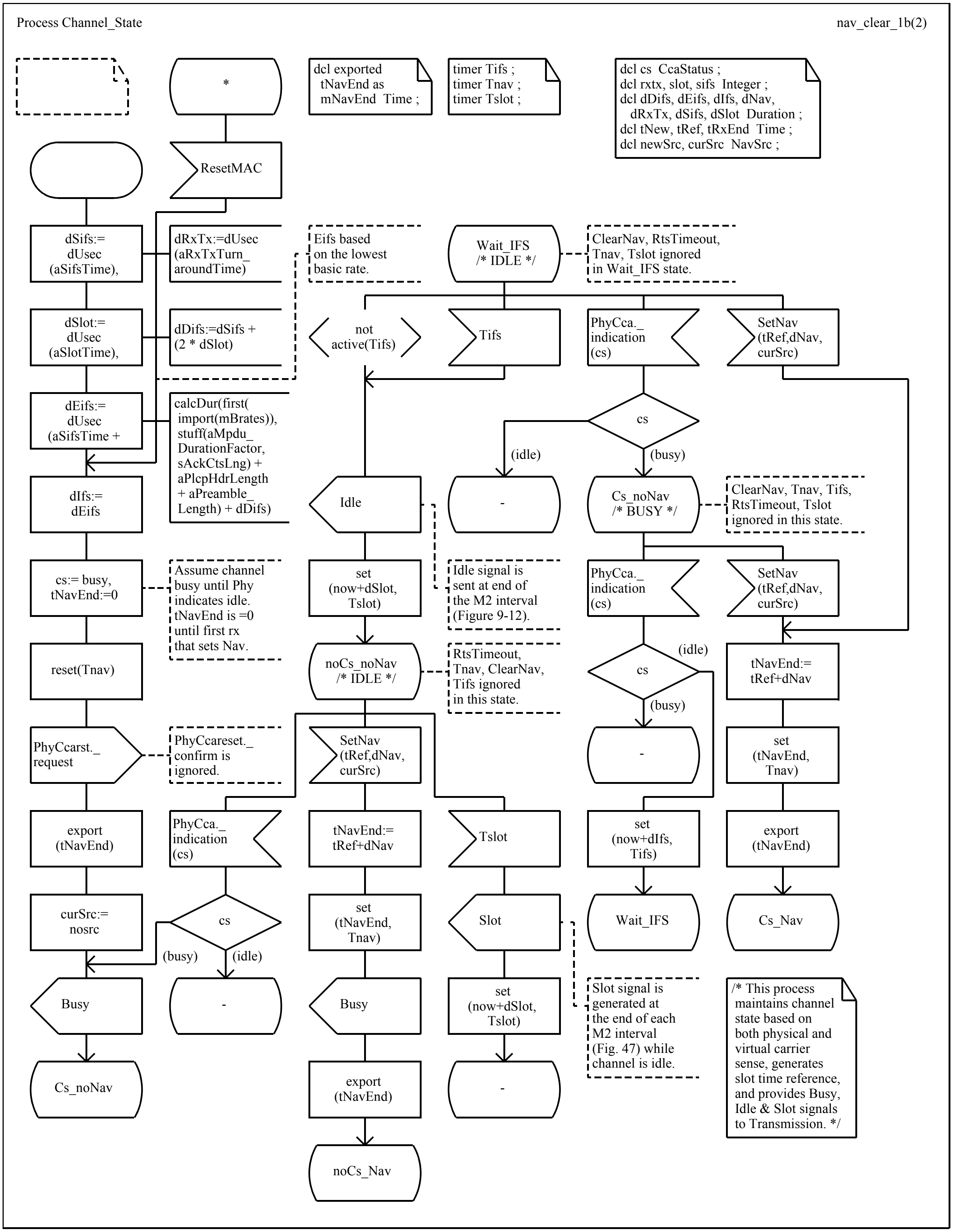

Service Distribute_Mmpdus
mmpdu_svc_1.1b(2)
chk_ sigtype
mSerr
(class2)
(class3)
Cls2Err (addr2 (mRpdu))
Cls3Err (addr2 (mRpdu))
else
(cfend, cfend_ack)
ftype (mRpdu)
(asoc_req)
(asoc_rsp)
Cfend
AsocReq (mRpdu)
AsocRsp (mRpdu)
(auth)
Mmpdu_ Idle
mTst:= mod(authSeqNum (mRpdu), 2)
(reasoc_req)
(reasoc_rsp)
mTst
ReasocReq (mRpdu)
ReasocRsp (mRpdu)
(0)
(1)
AuthEven (mRpdu)
AuthOdd (mRpdu)
(probe_req)
(probe_rsp)
(atim)
ProbeReq (mRpdu)
ProbeRsp (mRpdu, mtE,mtT)
Atim (mRpdu)
Mmpdu_ Idle
(beacon)
(disasoc)
(deauth)
Beacon (mRpdu, mtE,mtT)
Disasoc (mRpdu)
Deauth (mRpdu)
Mmpdu_ Idle

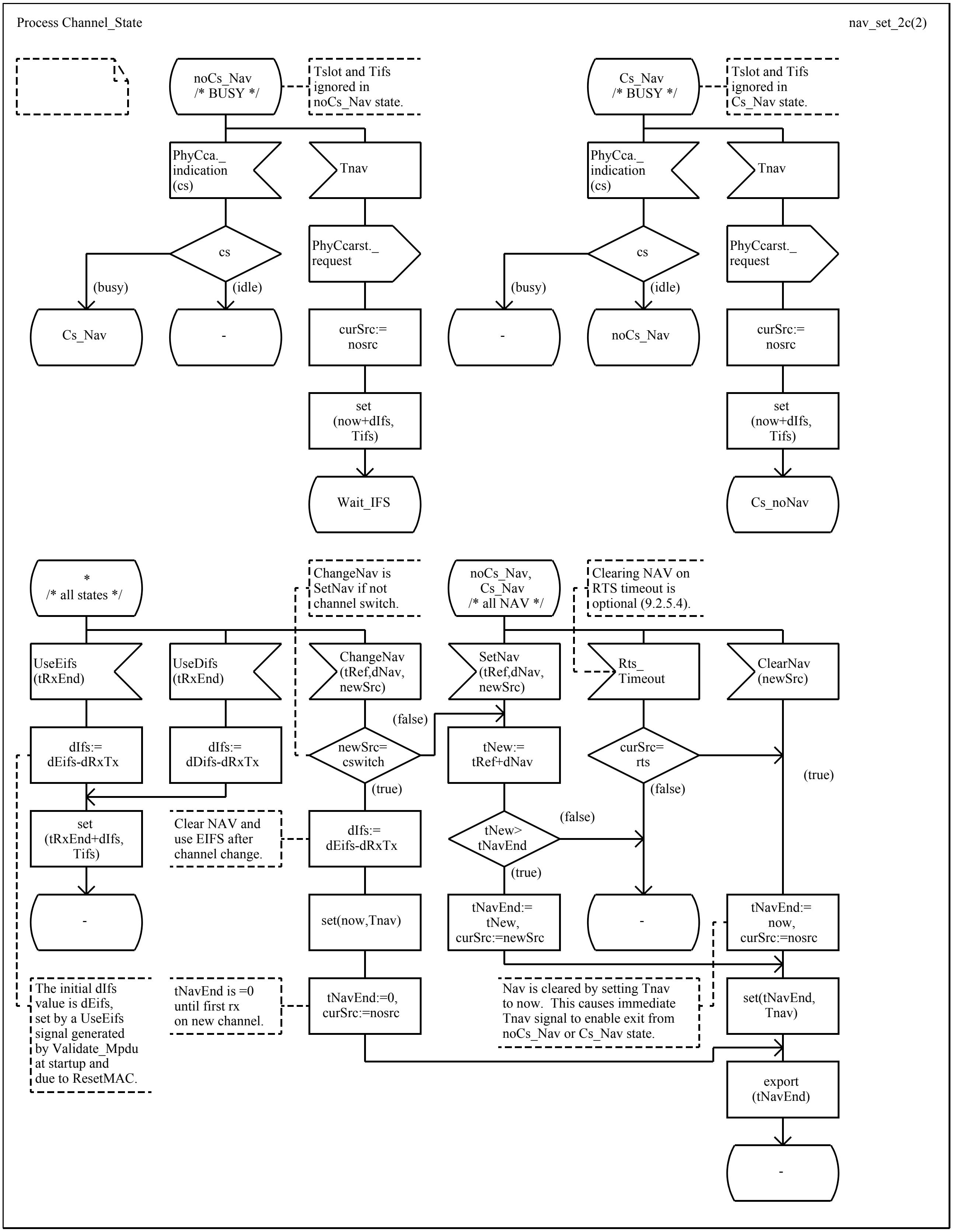

Service Synchronization_AP
ap_Init_1b(3)
dcl yAtimRx, yPsm, yRdtim, yWake Boolean ;
dcl yAtw, yBcn, yMocp Time ;
dcl yBcnPeriod, yDtim,ycmax, ycmin Kusec ;
dcl ybd BssDscr ;
dcl ybdset BssDscrSet ;
dcl ybtp BssType ;
dcl ybsid MacAddr ;
dcl yclist Intstring ;
dcl ycx, yJto, ytemp Integer ;
dcl yDspm DsParms ;
dcl yFhpm FhParms ;
dcl yIbpm IbssParms ;
dcl ypdly Usec ;
dcl yPhpm PhyParms ;
dcl yRdu, yTdu Frame ;
dcl yssid Octetstring ;
dcl yOrates Ratestring;
dcl ystp ScanType ;
dcl ytrsl TxResult ;
timer Tscan,
Tmocp ;
*
ResetMAC
'obtain PHY characteristics'
'reset all intra-MAC remote
variables to default values'
ytemp:= call TSF (0, true)
Set TSF time to zero.
Xport
reset(Tbcn), set(now,Tauth), set(now,Tchal)
Setting these timers to now causes events in each of the multi-state services of the process, forcing each service to its idle state.
No_BSS

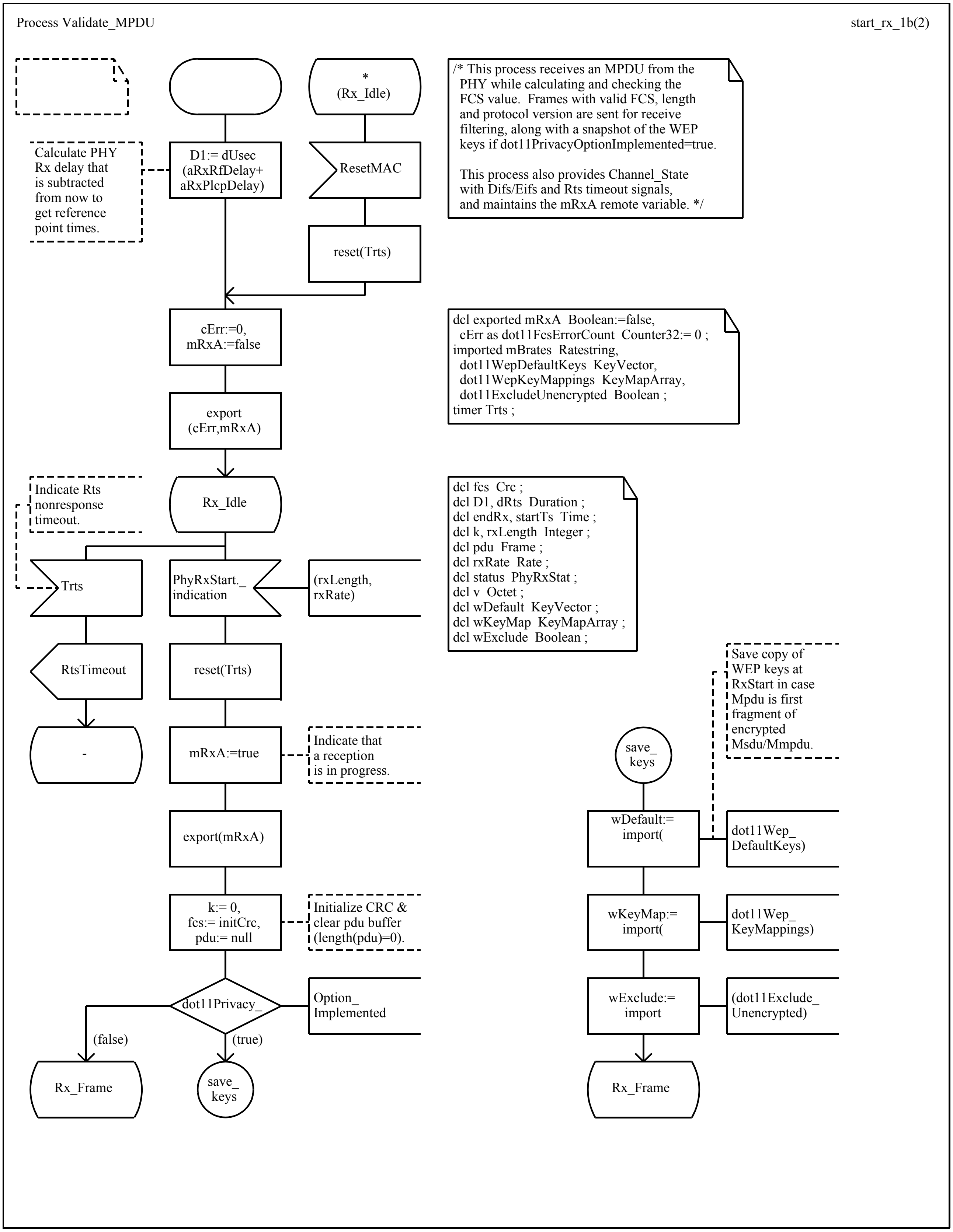

Service Synchronization_AP
ap_Start_Bss_2c(3)
No_BSS
Start IBSS on this page, join on next page.
Activate Station state machine.
MlmeStart._ request (mSsid, yBtp,
yBcnPeriod, yDtim, yCfpm, yPhpm, /* ibpm, */ mCap, mBrates, yOrates)
yBtp
(independent)
(infrastructure)
Sta_Active
dot11Multi_ Domain_
Capability_ Implemented
(true)
(false)
(false)
dot11Multi_ Domain_
CapabilityEnabled 'and country information valid'
(true)
'parameters valid'
(false)
(true)
dot11Power_ Management_ Mode:=
station_active, mPss:=awake, dot11Beacon_ Period:= yBcnPeriod
MlmeStart._ confirm (invalid)
No_Bss
dot11Dtim_ Period:=yDtim,
dot11Operational_ RateSet:=yOrates
export(dot11_ BeaconPeriod, dot11DtimPeriod,
dot11Operational_ RateSet,dot11Pow_ erManagementMode)
yBcn:= kUsec(yBcnPeriod)
mCfAvail:= if dtim_ Period
(yCfpm) /= 0 then true else false fi
'set mCfPoll and mCap for operating state'
set aCfPeriod and aCfMaxDuration from yCfpm'
'set actual phy parameters from phpm'
Xport
bss_ init
bss_ init
'dot11PHY_ Type=
FHphy'
(false)
(true)
yMocp:=kUsec (import(aMedium_ OccupancyLimit))
yMocp:=kUsec (dwellTime (yFhpm))
mNextBdry:= now+(yMocp - (call TSF
(0,false) mod yMocp))
set (mNextBdry, Tmocp)
Initialize dwell timer.
'yChan:= first (or only) channel'
Set starting channel (FH) or operating channel (DS), null for IR.
SwChnl (yChan,true)
mNextTbtt:= now+(yBcn - (call TSF
(0,false) mod yBcn))
set (mNextTbtt, Tbcn)
Initialize beacon timer.
Xport
MlmeStart._ confirm (success)
Bss


Service Synchronization_AP
ap_TSF_bss_3c(3)
Bss
Tbcn
Cfend
ProbeReq (yRdu)
Tmocp
set (now+yBcn, Tbcn)
mCfp:=false
'process request information element (if present) according to 7.3.2.12'
mNextBdry:= mNextBdry + yMocp
ytdu:= mkFrame (beacon,
bcstAddr, mBssId, O8 /*timestamp inserted by tx */ // mk2octets(yBcnPeriod)// mCap// mkElem(eSsId,mSsId)// mkElem(eSupRates, yOrates))
Xport
ytdu:= mkFrame (probe_rsp,
bcstAddr,mBssId, O8 /*timestamp inserted by tx */ // mk2octets(yBcnPeriod)// mCap// mkElem(eSsId,mSsId)// mkElem(eCfp, yCfpm)// mkElem(eSupRates, yOrates)//mkElem(ePhpm,yPhpm))
set (mNextBdry, Tmocp)
dot11MultiDomain CapabilityEnabled
(false)
(true)
'add country information element'
mDtimCount:= mDtimCount-1
-
mPcAvail
(false)
(true)
Send (ytdu,norm)
ytdu:= ytdu //
mkElem (eCfp,yCfpm)
-
yCfpm:= setCfpCount (yCfpm,
if yCfCnt=0 then import (dot11CfpPeriod)-1 else yCfCnt-1 fi)
yCfpm:= setCfpPeriod (yCfpm,
import (dot11CfpPeriod))
dot11PHY_ Type=
FHphy'
(false)
(true)
yCfpm:= setCfpMaxDur (yCfpm,
import (dot11CfpMaxDuration))
'yChan:= next channel in hop seq'
'update cfDurRem and mCfp'
SwChnl (,false)
SwChnl (yChan,true)
'add proper phy parameter set element'
Wait_Hop_ Bss
dot11MultiDomain CapabilityEnabled
(true)
'add country information element'
(false)
SwDone
*
Xport, Send (ytdu,imed)
TIM element gets added by PM_Filter_AP.
-
Bss


RX
PS
Block Reception
RxIndicate
PsIndicate
receive_1a(1)
FromCtl
NeedAck,
RxCfAck,
RxCfPoll
ToRx
Includes decryption if
dot11PrivacyOptionImplemented
=true. This is a typical
location, but implementers
may use other locations
between the PHY_SAP_RX
and MAC_SAP as long as
they provide the specified
behavior as observed at
LLC, MLME and the WM.
Defragment
(1,1)
/* also decrypt */
RxMpdu
/* This block handles octet-level
reception of MPDUs from the
PHY, and validation, filtering,
and decryption needed so higher
blocks have uniform, error-free
information from the relevant rx
events. This block also maintains
the MAC's view of channel state,
including the NAV (and remote
variable mNavEnd), rx activity
(and the remote variable mRxA),
and slot timing (providing the
Busy, Idle and Slot signals to
the Transmission block). */
FromSync
ChangeNav
Defrag
ChangeNav
IndAck
ToPs
Filter_MPDU
(1,1)
UpdNav
CS
ToTx
Channel_State
(1,1)
Busy, Idle, Slot
SetNav,
ClearNav
RxMpdu
PhyCca.indication,
PhyCcarst.confirm
RtsTimeout,
UseDifs,
UseEifs
Filter
signal ClearNav(NavSrc),
RtsTimeout,
RxMpdu(Frame,Time,Time,
Rate,Boolean,
KeyVector,KeyMapArray),
SetNav(Time,
Duration,NavSrc),
UseDifs(Time),
UseEifs(Time) ;
IfsCtl
Validate_MPDU
(1,1)
PhyRxStart.indication,
PhyRxEnd.indication,
PhyData.indication
PhyCca
FromPHY
PhyCcarst.request
PHY_SAP_RX


Process Channel_State
nav_clear_1b(2)
*
ResetMAC
dcl exported
tNavEnd as
mNavEnd Time ;
timer Tifs ;
timer Tnav ;
timer Tslot ;
dcl cs CcaStatus ;
dcl rxtx, slot, sifs Integer ;
dcl dDifs, dEifs, dIfs, dNav,
dRxTx, dSifs, dSlot Duration ;
dcl tNew, tRef, tRxEnd Time ;
dcl newSrc, curSrc NavSrc ;
dSifs:=
dUsec
(aSifsTime),
dRxTx:=dUsec
(aRxTxTurn_
aroundTime)
Eifs based
on the lowest
basic rate.
Wait_IFS
/* IDLE */
ClearNav, RtsTimeout,
Tnav, Tslot ignored
in Wait_IFS state.
dSlot:=
dUsec
(aSlotTime),
dDifs:=dSifs +
(2 * dSlot)
not
active(Tifs)
Tifs
PhyCca._
indication
(cs)
SetNav
(tRef,dNav,
curSrc)
dEifs:=
dUsec
(aSifsTime +
calcDur(first(
import(mBrates)),
stuff(aMpdu_
DurationFactor,
sAckCtsLng) +
aPlcpHdrLength
+ aPreamble_
Length) + dDifs)
cs
(idle)
(busy)
dIfs:=
dEifs
Idle
-
Cs_noNav
/* BUSY */
ClearNav, Tnav, Tifs,
RtsTimeout, Tslot
ignored in this state.
cs:= busy,
tNavEnd:=0
Assume channel
busy until Phy
indicates idle.
tNavEnd is =0
until first rx
that sets Nav.
set
(now+dSlot,
Tslot)
Idle signal is
sent at end of
the M2 interval
(Figure 9-12).
PhyCca._
indication
(cs)
SetNav
(tRef,dNav,
curSrc)
reset(Tnav)
noCs_noNav
/* IDLE */
RtsTimeout,
Tnav, ClearNav,
Tifs ignored
in this state.
cs
(idle)
(busy)
tNavEnd:=
tRef+dNav
PhyCcarst._
request
PhyCcareset._
confirm is
ignored.
SetNav
(tRef,dNav,
curSrc)
-
set
(tNavEnd,
Tnav)
export
(tNavEnd)
PhyCca._
indication
(cs)
tNavEnd:=
tRef+dNav
Tslot
set
(now+dIfs,
Tifs)
export
(tNavEnd)
curSrc:=
nosrc
cs
(busy)
(idle)
set
(tNavEnd,
Tnav)
Slot
Wait_IFS
Cs_Nav
Busy
-
Busy
set
(now+dSlot,
Tslot)
Slot signal is
generated at
the end of each
M2 interval
(Fig. 47) while
channel is idle.
/* This process
maintains channel
state based on
both physical and
virtual carrier
sense, generates
slot time reference,
and provides Busy,
Idle & Slot signals
to Transmission. */
Cs_noNav
export
(tNavEnd)
-
noCs_Nav


Process Channel_State
nav_set_2c(2)
noCs_Nav /* BUSY */
Tslot and Tifs ignored in noCs_Nav state.
PhyCca._ indication (cs)
Tnav
cs
(busy)
(idle)
Cs_Nav
-
PhyCcarst._ request
curSrc:= nosrc
set (now+dIfs, Tifs)
Wait_IFS
Cs_Nav /* BUSY */
Tslot and Tifs ignored in Cs_Nav state.
PhyCca._ indication (cs)
Tnav
cs
(busy)
(idle)
-
noCs_Nav
PhyCcarst._ request
curSrc:= nosrc
set (now+dIfs, Tifs)
Cs_noNav
* /* all states */
ChangeNav is SetNav if not channel switch.
noCs_Nav, Cs_Nav /* all NAV */
Clearing NAV on RTS timeout is optional (9.2.5.4).
UseEifs (tRxEnd)
UseDifs (tRxEnd)
ChangeNav (tRef,dNav, newSrc)
SetNav (tRef,dNav, newSrc)
Rts_ Timeout
ClearNav (newSrc)
(false)
dIfs:= dEifs-dRxTx
dIfs:= dDifs-dRxTx
newSrc= cswitch
tNew:= tRef+dNav
curSrc= rts
(true)
(true)
(false)
set (tRxEnd+dIfs, Tifs)
Clear NAV and use EIFS after channel change.
dIfs:= dEifs-dRxTx
tNew> tNavEnd
(false)
(true)
-
set(now,Tnav)
tNavEnd:= tNew, curSrc:=newSrc
-
tNavEnd:= now, curSrc:=nosrc
The initial dIfs value is dEifs, set by a UseEifs signal generated by Validate_Mpdu at startup and due to ResetMAC.
tNavEnd is =0 until first rx on new channel.
tNavEnd:=0, curSrc:=nosrc
Nav is cleared by setting Tnav to now. This causes immediate Tnav signal to enable exit from noCs_Nav or Cs_Nav state.
set(tNavEnd, Tnav)
export (tNavEnd)
-

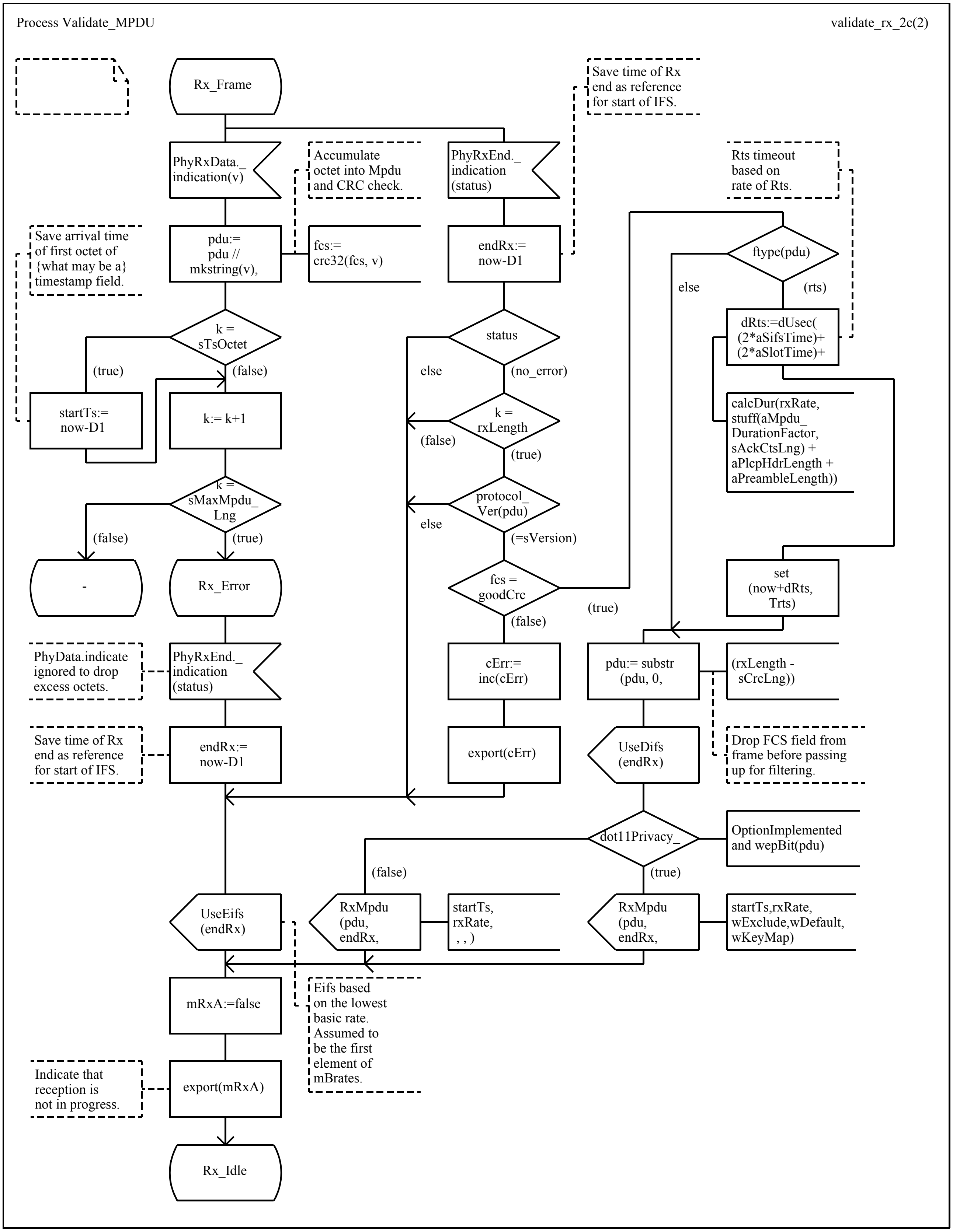

Process Validate_MPDU
start_rx_1b(2)
*
(Rx_Idle)
ResetMAC
reset(Trts)
Calculate PHY Rx delay that is subtracted from now to get reference point times.
D1:= dUsec (aRxRfDelay+ aRxPlcpDelay)
cErr:=0, mRxA:=false
export (cErr,mRxA)
Rx_Idle
Indicate Rts nonresponse timeout.
Trts
RtsTimeout
-
PhyRxStart._ indication
(rxLength, rxRate)
reset(Trts)
mRxA:=true
Indicate that a reception is in progress.
export(mRxA)
k:= 0, fcs:= initCrc, pdu:= null
Initialize CRC & clear pdu buffer (length(pdu)=0).
dot11Privacy_
Option_ Implemented
(false)
(true)
Rx_Frame
save_ keys
save_ keys
Save copy of WEP keys at RxStart in case Mpdu is first fragment of encrypted Msdu/Mmpdu.
wDefault:= import(
dot11Wep_ DefaultKeys)
wKeyMap:= import(
dot11Wep_ KeyMappings)
wExclude:= import
(dot11Exclude_ Unencrypted)
Rx_Frame

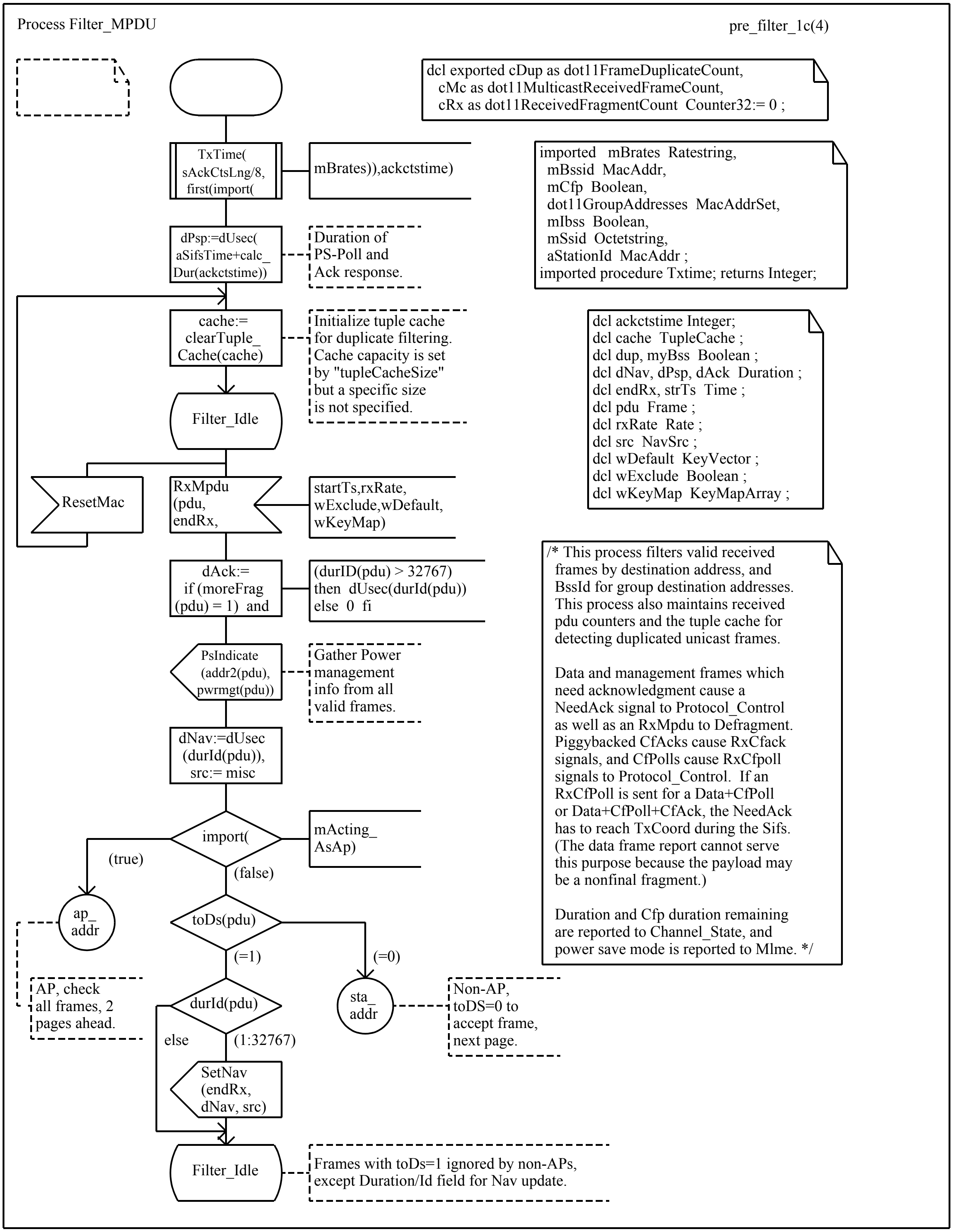
/* This process receives an MPDU from the PHY while calculating and checking the FCS value. Frames with valid FCS, length and protocol version are sent for receive filtering, along with a snapshot of the WEP keys if dot11PrivacyOptionImplemented=true.

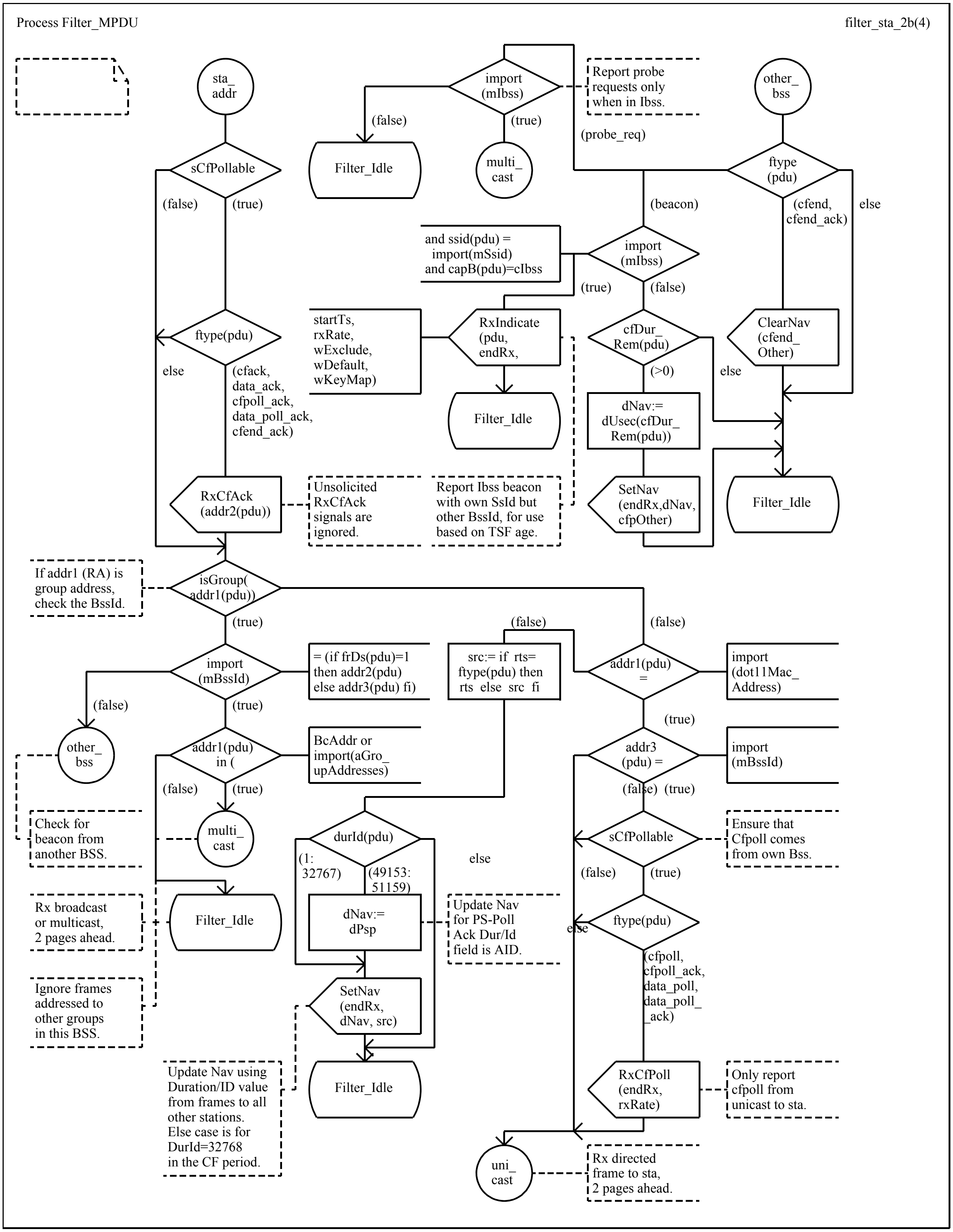
This process also provides Channel_State with Difs/Eifs and Rts timeout signals, and maintains the mRxA remote variable. */

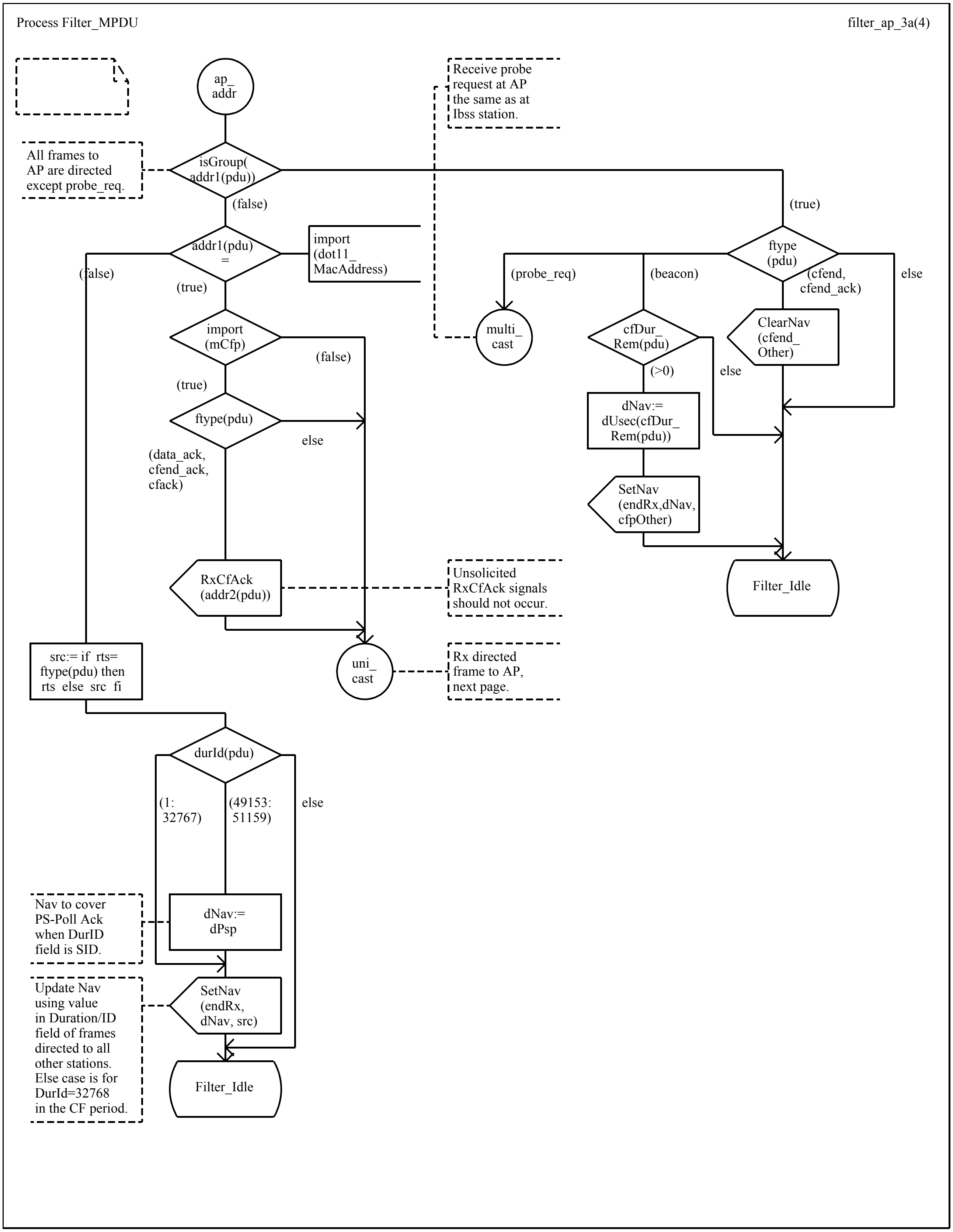
```
dcl exported mRxA  Boolean:=false,
  cErr as dot11FcsErrorCount  Counter32:= 0 ;
imported mBrates  Ratestring,
  dot11WepDefaultKeys  KeyVector,
  dot11WepKeyMappings  KeyMapArray,
  dot11ExcludeUnencrypted  Boolean ;
timer Trts ;
```

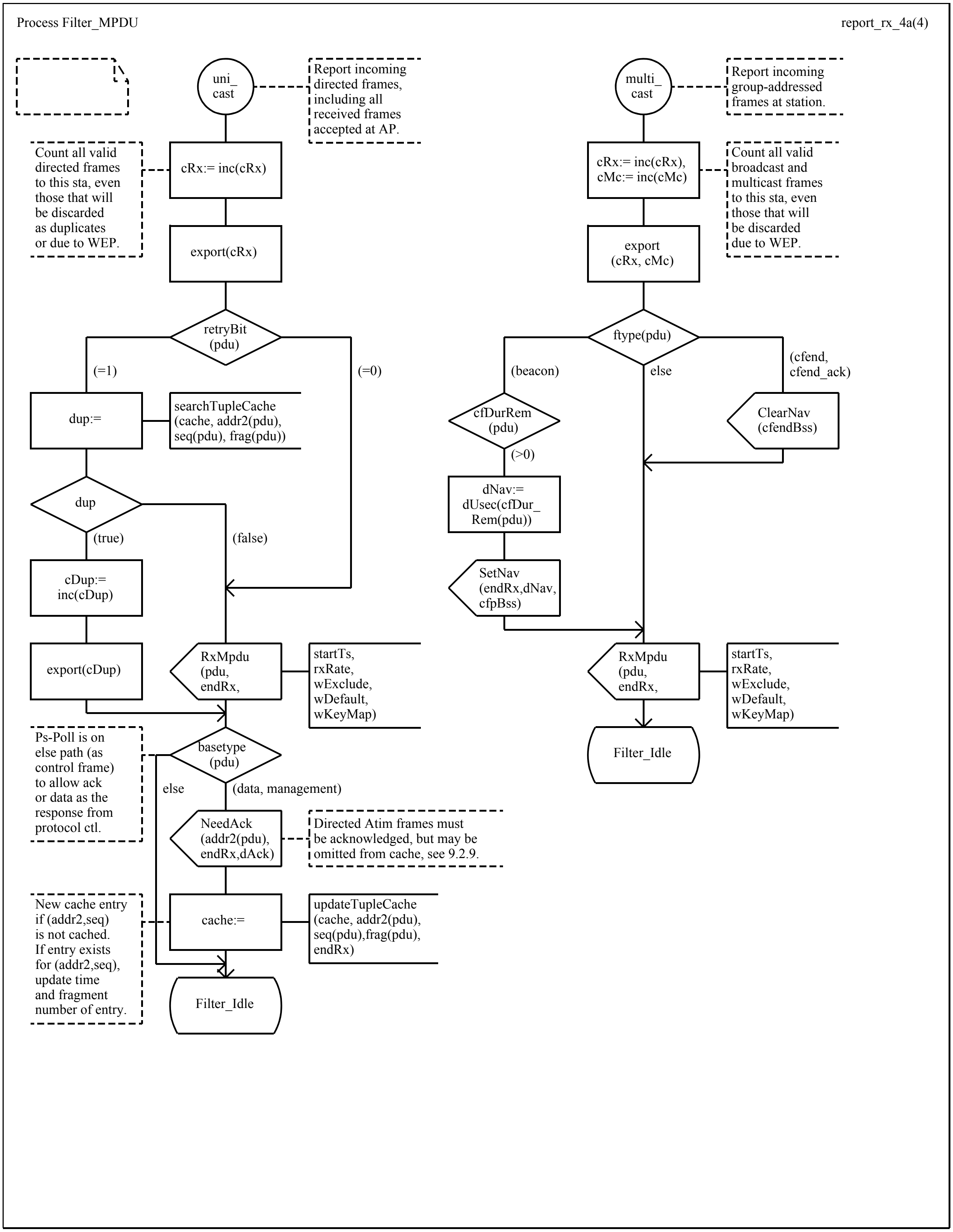
```
dcl fcs  Crc ;
dcl D1, dRts  Duration ;
dcl endRx, startTs  Time ;
dcl k, rxLength  Integer ;
dcl pdu  Frame ;
dcl rxRate  Rate ;
dcl status  PhyRxStat ;
dcl v  Octet ;
dcl wDefault  KeyVector ;
dcl wKeyMap  KeyMapArray ;
dcl wExclude  Boolean ;
```

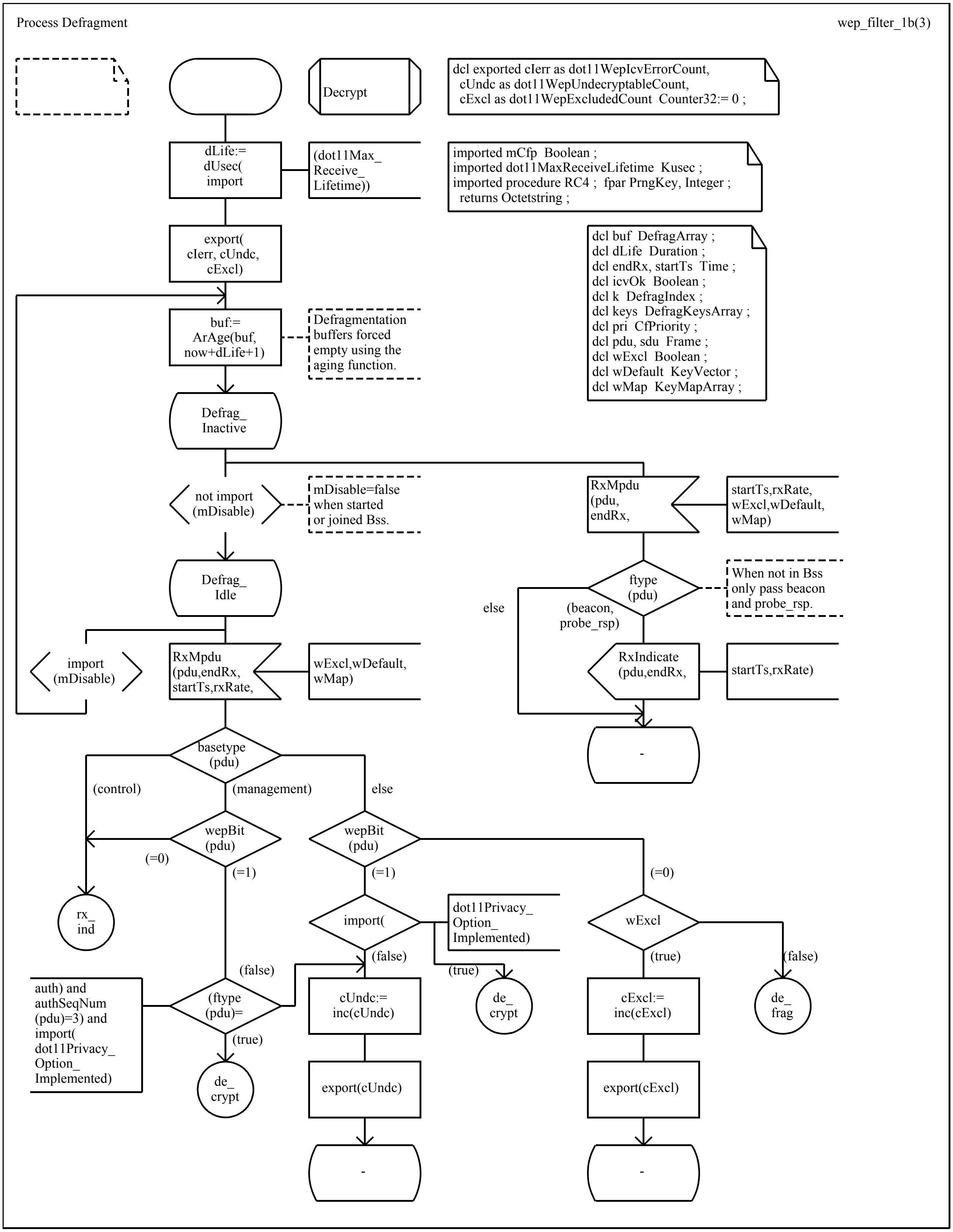

Process Validate_MPDU
validate_rx_2c(2)
Rx_Frame
PhyRxData._ indication(v)
Accumulate octet into Mpdu and CRC check.
PhyRxEnd._ indication (status)
Save time of Rx end as reference for start of IFS.
Rts timeout based on rate of Rts.
Save arrival time of first octet of {what may be a} timestamp field.
pdu:= pdu // mkstring(v),
fcs:= crc32(fcs, v)
endRx:= now-D1
ftype(pdu)
else
(rts)
k = sTsOctet
(true)
(false)
status
else
(no_error)
dRts:=dUsec((2*aSifsTime)+ (2*aSlotTime)+
startTs:= now-D1
k:= k+1
k = rxLength
(false)
(true)
calcDur(rxRate, stuff(aMpdu_ DurationFactor, sAckCtsLng) + aPlcpHdrLength + aPreambleLength))
k = sMaxMpdu_ Lng
protocol_ Ver(pdu)
else
(false)
(true)
(=sVersion)
-
Rx_Error
fcs = goodCrc
(true)
set (now+dRts, Trts)
(false)
PhyData.indicate ignored to drop excess octets.
PhyRxEnd._ indication (status)
cErr:= inc(cErr)
pdu:= substr (pdu, 0,
(rxLength - sCrcLng))
Save time of Rx end as reference for start of IFS.
endRx:= now-D1
export(cErr)
UseDifs (endRx)
Drop FCS field from frame before passing up for filtering.
dot11Privacy_
OptionImplemented and wepBit(pdu)
(false)
(true)
UseEifs (endRx)
RxMpdu (pdu, endRx,
startTs, rxRate, , ,)
RxMpdu (pdu, endRx,
startTs,rxRate, wExclude,wDefault, wKeyMap)
mRxA:=false
Eifs based on the lowest basic rate. Assumed to be the first element of mBrates.
Indicate that reception is not in progress.
export(mRxA)
Rx_Idle

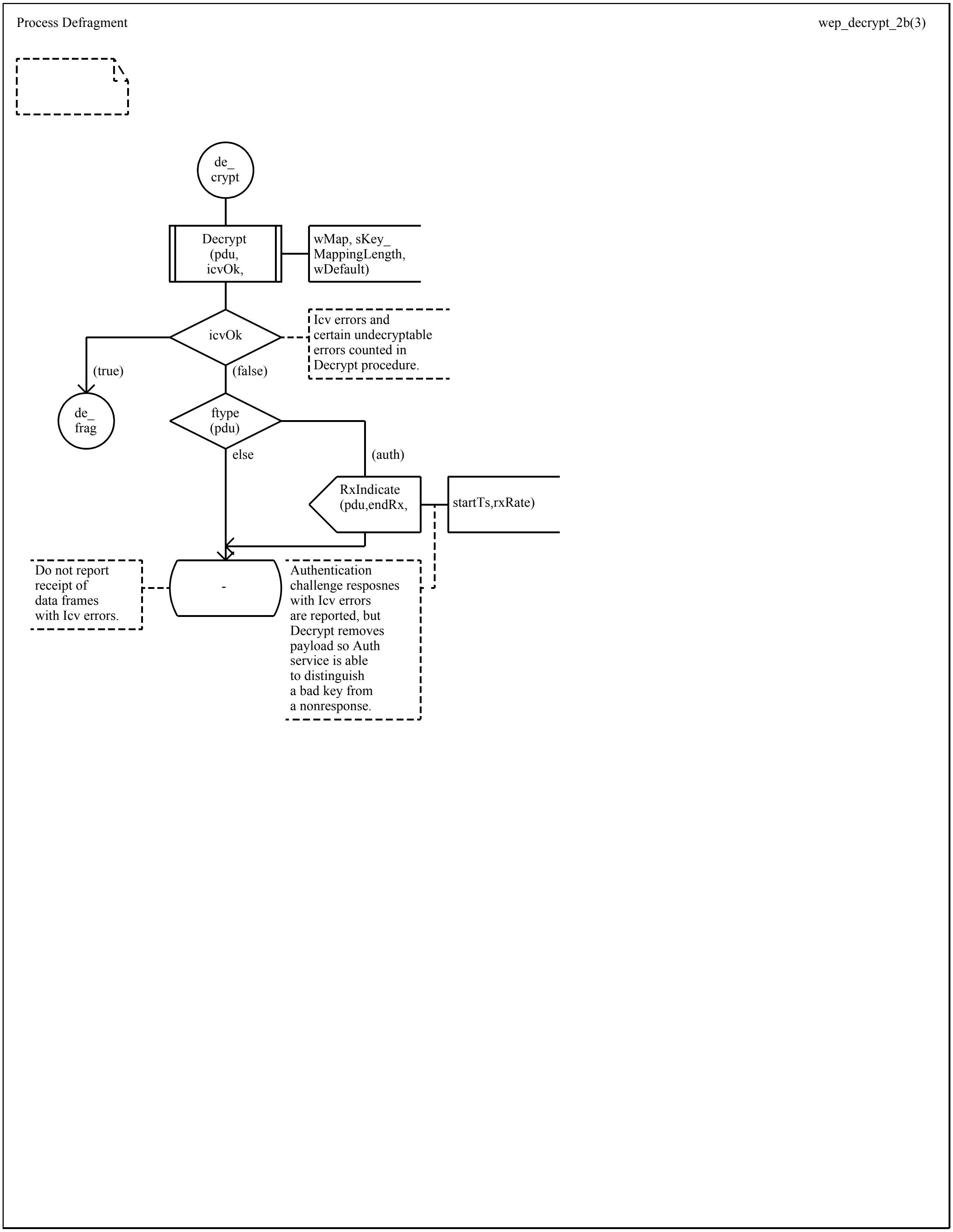

Process Filter_MPDU
pre_filter_1c(4)
dcl exported cDup as dot11FrameDuplicateCount,
cMc as dot11MulticastReceivedFrameCount,
cRx as dot11ReceivedFragmentCount Counter32:= 0 ;
TxTime(sAckCtsLng/8, first(import(
mBrates)),ackctstime)
imported mBrates Ratestring,
mBssid MacAddr,
mCfp Boolean,
dot11GroupAddresses MacAddrSet,
mIbss Boolean,
mSsid Octetstring,
aStationId MacAddr ;
imported procedure Txtime; returns Integer;
dPsp:=dUsec(aSifsTime+calc_ Dur(ackctstime))
Duration of PS-Poll and Ack response.
cache:= clearTuple_ Cache(cache)
Initialize tuple cache for duplicate filtering. Cache capacity is set by "tupleCacheSize" but a specific size is not specified.
dcl ackctstime Integer;
dcl cache TupleCache ;
dcl dup, myBss Boolean ;
dcl dNav, dPsp, dAck Duration ;
dcl endRx, strTs Time ;
dcl pdu Frame ;
dcl rxRate Rate ;
dcl src NavSrc ;
dcl wDefault KeyVector ;
dcl wExclude Boolean ;
dcl wKeyMap KeyMapArray ;
Filter_Idle
ResetMac
RxMpdu (pdu, endRx,
startTs,rxRate, wExclude,wDefault, wKeyMap)
/* This process filters valid received frames by destination address, and BssId for group destination addresses. This process also maintains received pdu counters and the tuple cache for detecting duplicated unicast frames.
Data and management frames which need acknowledgment cause a NeedAck signal to Protocol_Control as well as an RxMpdu to Defragment. Piggybacked CfAcks cause RxCfack signals, and CfPolls cause RxCfpoll signals to Protocol_Control. If an RxCfPoll is sent for a Data+CfPoll or Data+CfPoll+CfAck, the NeedAck has to reach TxCoord during the Sifs. (The data frame report cannot serve this purpose because the payload may be a nonfinal fragment.)
Duration and Cfp duration remaining are reported to Channel_State, and power save mode is reported to Mlme. */
dAck:= if (moreFrag (pdu) = 1) and
(durID(pdu) > 32767) then dUsec(durId(pdu)) else 0 fi
PsIndicate (addr2(pdu), pwrmgt(pdu))
Gather Power management info from all valid frames.
dNav:=dUsec (durId(pdu)), src:= misc
import(
mActing_ AsAp)
(true)
(false)
ap_ addr
AP, check all frames, 2 pages ahead.
toDs(pdu)
(=1)
(=0)
sta_ addr
Non-AP, toDS=0 to accept frame, next page.
durId(pdu)
else
(1:32767)
SetNav (endRx, dNav, src)
Filter_Idle
Frames with toDs=1 ignored by non-APs, except Duration/Id field for Nav update.

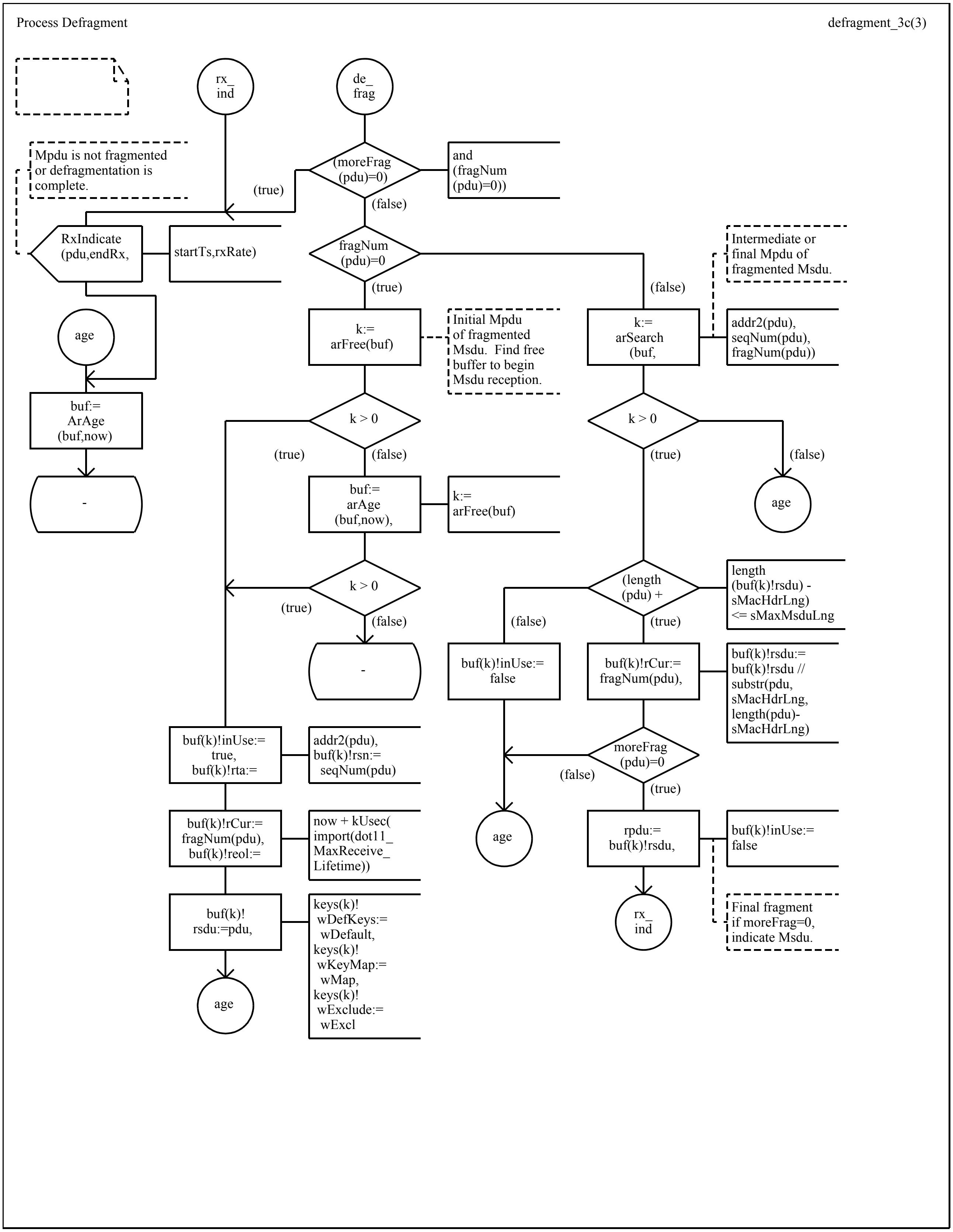

Process Filter_MPDU
filter_sta_2b(4)
sta_
addr
sCfPollable
(false)
(true)
ftype(pdu)
else
(cfack,
data_ack,
cfpoll_ack,
data_poll_ack,
cfend_ack)
RxCfAck
(addr2(pdu))
Unsolicited
RxCfAck
signals are
ignored.
import
(mIbss)
Report probe
requests only
when in Ibss.
(false)
(true)
Filter_Idle
multi_
cast
(probe_req)
other_
bss
ftype
(pdu)
(beacon)
(cfend,
cfend_ack)
else
and ssid(pdu) =
import(mSsid)
and capB(pdu)=cIbss
import
(mIbss)
(true)
(false)
startTs,
rxRate,
wExclude,
wDefault,
wKeyMap)
RxIndicate
(pdu,
endRx,
Filter_Idle
cfDur_
Rem(pdu)
(>0)
else
ClearNav
(cfend_
Other)
dNav:=
dUsec(cfDur_
Rem(pdu))
SetNav
(endRx,dNav,
cfpOther)
Filter_Idle
Report Ibss beacon
with own SsId but
other BssId, for use
based on TSF age.
If addr1 (RA) is
group address,
check the BssId.
isGroup(
addr1(pdu))
(true)
(false)
import
(mBssId)
= (if frDs(pdu)=1
then addr2(pdu)
else addr3(pdu) fi)
(false)
(true)
other_
bss
addr1(pdu)
in (
BcAddr or
import(aGro_
upAddresses)
(false)
(true)
multi_
cast
Check for
beacon from
another BSS.
Filter_Idle
Rx broadcast
or multicast,
2 pages ahead.
Ignore frames
addressed to
other groups
in this BSS.
(false)
src:= if rts=
ftype(pdu) then
rts else src fi
durId(pdu)
(1:
32767)
(49153:
51159)
else
dNav:=
dPsp
Update Nav
for PS-Poll
Ack Dur/Id
field is AID.
SetNav
(endRx,
dNav, src)
Filter_Idle
Update Nav using
Duration/ID value
from frames to all
other stations.
Else case is for
DurId=32768
in the CF period.
addr1(pdu)
=
import
(dot11Mac_
Address)
(true)
addr3
(pdu) =
import
(mBssId)
(false)
(true)
sCfPollable
Ensure that
Cfpoll comes
from own Bss.
(false)
(true)
ftype(pdu)
else
(cfpoll,
cfpoll_ack,
data_poll,
data_poll_
_ack)
RxCfPoll
(endRx,
rxRate)
Only report
cfpoll from
unicast to sta.
uni_
cast
Rx directed
frame to sta,
2 pages ahead.

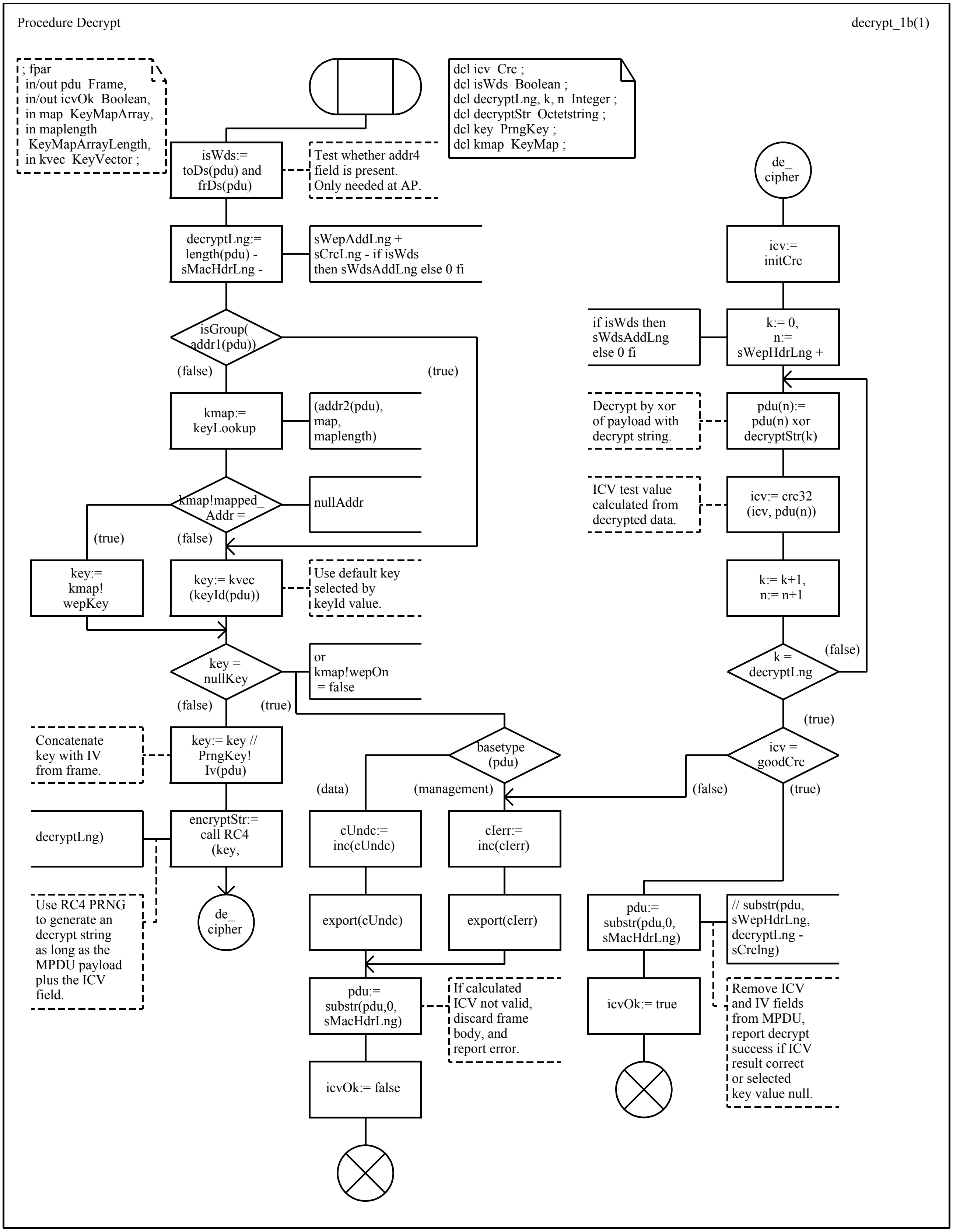

Process Filter_MPDU
filter_ap_3a(4)
ap_
addr
Receive probe
request at AP
the same as at
Ibss station.
All frames to
AP are directed
except probe_req.
isGroup(
addr1(pdu))
(false)
(true)
addr1(pdu)
=
import
(dot11_
MacAddress)
(false)
(true)
ftype
(pdu)
(probe_req)
(beacon)
(cfend,
cfend_ack)
else
import
(mCfp)
(false)
(true)
multi_
cast
cfDur_
Rem(pdu)
ClearNav
(cfend_
Other)
(>0)
else
ftype(pdu)
else
dNav:=
dUsec(cfDur_
Rem(pdu))
(data_ack,
cfend_ack,
cfack)
SetNav
(endRx,dNav,
cfpOther)
RxCfAck
(addr2(pdu))
Unsolicited
RxCfAck signals
should not occur.
Filter_Idle
src:= if rts=
ftype(pdu) then
rts else src fi
uni_
cast
Rx directed
frame to AP,
next page.
durId(pdu)
(1:
32767)
(49153:
51159)
else
Nav to cover
PS-Poll Ack
when DurID
field is SID.
dNav:=
dPsp
Update Nav
using value
in Duration/ID
field of frames
directed to all
other stations.
Else case is for
DurId=32768
in the CF period.
SetNav
(endRx,
dNav, src)
Filter_Idle


Process Filter_MPDU
report_rx_4a(4)
uni_ cast
Report incoming directed frames, including all received frames accepted at AP.
Count all valid directed frames to this sta, even those that will be discarded as duplicates or due to WEP.
cRx:= inc(cRx)
export(cRx)
retryBit (pdu)
(=1)
(=0)
dup:=
searchTupleCache (cache, addr2(pdu), seq(pdu), frag(pdu))
dup
(true)
(false)
cDup:= inc(cDup)
export(cDup)
RxMpdu (pdu, endRx,
startTs, rxRate, wExclude, wDefault, wKeyMap)
Ps-Poll is on else path (as control frame) to allow ack or data as the response from protocol ctl.
basetype (pdu)
else
(data, management)
NeedAck (addr2(pdu), endRx,dAck)
Directed Atim frames must be acknowledged, but may be omitted from cache, see 9.2.9.
New cache entry if (addr2,seq) is not cached. If entry exists for (addr2,seq), update time and fragment number of entry.
cache:=
updateTupleCache (cache, addr2(pdu), seq(pdu),frag(pdu), endRx)
Filter_Idle
multi_ cast
Report incoming group-addressed frames at station.
cRx:= inc(cRx), cMc:= inc(cMc)
Count all valid broadcast and multicast frames to this sta, even those that will be discarded due to WEP.
export (cRx, cMc)
ftype(pdu)
(beacon)
else
(cfend, cfend_ack)
cfDurRem (pdu)
(>0)
ClearNav (cfendBss)
dNav:= dUsec(cfDur_ Rem(pdu))
SetNav (endRx,dNav, cfpBss)
RxMpdu (pdu, endRx,
startTs, rxRate, wExclude, wDefault, wKeyMap)
Filter_Idle


Process Defragment
wep_filter_1b(3)
Decrypt
dcl exported cIerr as dot11WepIcvErrorCount,
cUndc as dot11WepUndecryptableCount,
cExcl as dot11WepExcludedCount Counter32:= 0 ;
dLife:= dUsec(import
(dot11Max_ Receive_ Lifetime))
imported mCfp Boolean ;
imported dot11MaxReceiveLifetime Kusec ;
imported procedure RC4 ; fpar PrngKey, Integer ;
returns Octetstring ;
export(cIerr, cUndc, cExcl)
dcl buf DefragArray ;
dcl dLife Duration ;
dcl endRx, startTs Time ;
dcl icvOk Boolean ;
dcl k DefragIndex ;
dcl keys DefragKeysArray ;
dcl pri CfPriority ;
dcl pdu, sdu Frame ;
dcl wExcl Boolean ;
dcl wDefault KeyVector ;
dcl wMap KeyMapArray ;
buf:= ArAge(buf, now+dLife+1)
Defragmentation buffers forced empty using the aging function.
Defrag_ Inactive
not import (mDisable)
mDisable=false when started or joined Bss.
RxMpdu (pdu, endRx,
startTs,rxRate, wExcl,wDefault, wMap)
Defrag_ Idle
ftype (pdu)
When not in Bss only pass beacon and probe_rsp.
else
(beacon, probe_rsp)
import (mDisable)
RxMpdu (pdu,endRx, startTs,rxRate,
wExcl,wDefault, wMap)
RxIndicate (pdu,endRx,
startTs,rxRate)
-
basetype (pdu)
(control)
(management)
else
wepBit (pdu)
(=0)
(=1)
wepBit (pdu)
(=1)
(=0)
rx_ ind
import(
dot11Privacy_ Option_ Implemented)
wExcl
(false)
(false)
(true)
(true)
(false)
auth) and authSeqNum (pdu)=3) and import(dot11Privacy_ Option_ Implemented)
(ftype (pdu)=
cUndc:= inc(cUndc)
de_ crypt
cExcl:= inc(cExcl)
de_ frag
(true)
de_ crypt
export(cUndc)
export(cExcl)
-
-


Process Defragment
wep_decrypt_2b(3)
de_
crypt
Decrypt
(pdu,
icvOk,
wMap, sKey_
MappingLength,
wDefault)
icvOk
Icv errors and
certain undecryptable
errors counted in
Decrypt procedure.
(true)
(false)
de_
frag
ftype
(pdu)
else
(auth)
RxIndicate
(pdu,endRx,
startTs,rxRate)
Do not report
receipt of
data frames
with Icv errors.
-
Authentication
challenge resposnes
with Icv errors
are reported, but
Decrypt removes
payload so Auth
service is able
to distinguish
a bad key from
a nonresponse.


Process Defragment
defragment_3c(3)
rx_
ind
de_
frag
Mpdu is not fragmented
or defragmentation is
complete.
(moreFrag
(pdu)=0)
and
(fragNum
(pdu)=0))
(true)
(false)
RxIndicate
(pdu,endRx,
startTs,rxRate)
fragNum
(pdu)=0
(true)
(false)
Intermediate or
final Mpdu of
fragmented Msdu.
age
k:=
arFree(buf)
Initial Mpdu
of fragmented
Msdu. Find free
buffer to begin
Msdu reception.
k:=
arSearch
(buf,
addr2(pdu),
seqNum(pdu),
fragNum(pdu))
buf:=
ArAge
(buf,now)
k > 0
(true)
(false)
k > 0
(true)
(false)
age
-
buf:=
arAge
(buf,now),
k:=
arFree(buf)
k > 0
(true)
(false)
-
(length
(pdu) +
length
(buf(k)!rsdu) -
sMacHdrLng)
<= sMaxMsduLng
(false)
(true)
buf(k)!inUse:=
false
buf(k)!rCur:=
fragNum(pdu),
buf(k)!rsdu:=
buf(k)!rsdu //
substr(pdu,
sMacHdrLng,
length(pdu)-
sMacHdrLng)
buf(k)!inUse:=
true,
buf(k)!rta:=
addr2(pdu),
buf(k)!rsn:=
seqNum(pdu)
moreFrag
(pdu)=0
(false)
(true)
buf(k)!rCur:=
fragNum(pdu),
buf(k)!reol:=
now + kUsec(
import(dot11_
MaxReceive_
Lifetime))
age
rpdu:=
buf(k)!rsdu,
buf(k)!inUse:=
false
buf(k)!
rsdu:=pdu,
keys(k)!
wDefKeys:=
wDefault,
keys(k)!
wKeyMap:=
wMap,
keys(k)!
wExclude:=
wExcl
rx_
ind
Final fragment
if moreFrag=0,
indicate Msdu.
age


Procedure Decrypt
decrypt_1b(1)
; fpar
in/out pdu Frame,
in/out icvOk Boolean,
in map KeyMapArray,
in maplength
KeyMapArrayLength,
in kvec KeyVector ;
dcl icv Crc ;
dcl isWds Boolean ;
dcl decryptLng, k, n Integer ;
dcl decryptStr Octetstring ;
dcl key PrngKey ;
dcl kmap KeyMap ;
isWds:=
toDs(pdu) and
frDs(pdu)
Test whether addr4
field is present.
Only needed at AP.
decryptLng:=
length(pdu) -
sMacHdrLng -
sWepAddLng +
sCrcLng - if isWds
then sWdsAddLng else 0 fi
isGroup(
addr1(pdu))
(false)
(true)
kmap:=
keyLookup
(addr2(pdu),
map,
maplength)
kmap!mapped_
Addr =
nullAddr
(true)
(false)
key:=
kmap!
wepKey
key:= kvec
(keyId(pdu))
Use default key
selected by
keyId value.
key =
nullKey
or
kmap!wepOn
= false
(false)
(true)
Concatenate
key with IV
from frame.
key:= key //
PrngKey!
Iv(pdu)
decryptLng)
encryptStr:=
call RC4
(key,
Use RC4 PRNG
to generate an
decrypt string
as long as the
MPDU payload
plus the ICV
field.
de_
cipher
basetype
(pdu)
(data)
(management)
cUndc:=
inc(cUndc)
cIerr:=
inc(cIerr)
export(cUndc)
export(cIerr)
pdu:=
substr(pdu,0,
sMacHdrLng)
If calculated
ICV not valid,
discard frame
body, and
report error.
icvOk:= false
de_
cipher
icv:=
initCrc
if isWds then
sWdsAddLng
else 0 fi
k:= 0,
n:=
sWepHdrLng +
Decrypt by xor
of payload with
decrypt string.
pdu(n):=
pdu(n) xor
decryptStr(k)
ICV test value
calculated from
decrypted data.
icv:= crc32
(icv, pdu(n))
k:= k+1,
n:= n+1
k =
decryptLng
(false)
(true)
icv =
goodCrc
(false)
(true)
pdu:=
substr(pdu,0,
sMacHdrLng)
// substr(pdu,
sWepHdrLng,
decryptLng -
sCrclng)
Remove ICV
and IV fields
from MPDU,
report decrypt
success if ICV
result correct
or selected
key value null.
icvOk:= true

Annex K

(informative)

High Rate PHY/FH interoperability

K.1 Status of this Annex

The mechanisms described in this annex are obsolete. Consequently, this annex may be removed in a future revision of the standard.

K.2 General

The Channel Agility option described in 17.4.6.8 provides for IEEE 802.11 FH PHY interoperability with the High Rate PHY. The FH patterns, as defined within this annex, enable synchronization with an FH-PHY-compliant BSS in North America and most of Europe. In addition, CCA requirements on a High Rate STA using this mode provide for CCA detection of 1 MHz wide FH signals within the wideband DS channel selected. FH PHY STAs operating in mixed mode FH/DS environments are advised to use similar cross PHY CCA mechanisms. The FH (Channel Agility) and cross CCA mechanisms provide the basic mechanisms to enable coexistence and interoperability.

The MAC elements include both DS and FH elements in Beacon frames and Probe Response frames when the Channel Agility option is turned on. Added capability fields indicate the ability to support the Channel Agility option and to indicate whether the option is turned on. These fields allow synchronization to the hopping sequence and timing, identification of what modes are being used within a BSS when joining on either High Rate or FHSS sides, and rejection of an association request in some cases.

Interoperability within an infrastructure BSS can be achieved, as an example, using a virtual dual AP. A virtual dual AP is defined, for purposes of discussion, as two logically separate APs that exist within a single physical AP with a single radio (one transmit and one receive path). Both FHSS and High Rate logical APs send out their own Beacon frames, DTIMs, and other nondirected packets. The two sides interact in the sharing of the medium and the AP’s processor and radio. Addressing and association issues may be handled in one of several ways and are left as an implementation choice.

Minimal interoperability with a nonhopping High Rate or legacy DSSS is provided by the use of a channel at least 1/7 or more of the time. While throughput would be significantly reduced by having a channel only 1/7 of the time, connection and minimal throughput can be provided.

When the FH option is utilized, the HR/DSSS PHY should provide the CCA capability to detect 1 MHz wide FH PHY signals operating within the wideband DS channel at levels 10 dB higher than that specified in 17.4.8.5 for wideband HR/DSSS signals. This is in addition to the primary CCA requirements in 17.4.8.5. A timeout mechanism to avoid excessive deferral to constant CW or other non-IEEE-802.11 type signals is allowed.

FH PHY STAs operating in mixed environments should provide similar CCA mechanisms to detect wideband DSSS signals at levels specified in 17.4.8.5, but measured within a 1 MHz bandwidth. Signal levels measured in a full DSSS channel are generally 10 dB or higher.

Annex L

(informative)

Examples of encoding a frame for OFDM PHYs

L.1 Example 1 - BCC encoding

L.1.1 Introduction

The purpose of this annex is to show an example of encoding a frame for the OFDM PHY, as described in Clause 18. This example covers all the encoding details defined by this standard.

The encoding illustration goes through the following stages:

a) Generating the short training sequence section of the preamble;
b) Generating the long preamble sequence section of the preamble;
c) Generating the SIGNAL field bits;
d) Coding and interleaving the SIGNAL field bits;
e) Mapping the SIGNAL field into frequency domain;
f) Pilot insertion;
g) Transforming into time domain;
h) Delineating the data octet stream into a bit stream;
i) Prepending the SERVICE field and adding the pad bits, thus forming the DATA;
j) Scrambling and zeroing the tail bits;
k) Encoding the DATA with a convolutional encoder and puncturing;
l) Mapping into complex 16-QAM symbols;
m) Pilot insertion;
n) Transforming from frequency to time and adding a circular prefix;
o) Concatenating the OFDM symbols into a single, time-domain signal.

In the description of time domain waveforms, a complex baseband signal at 20 Msample/s shall be used.

This example uses the 36 Mb/s data rate and a message of 100 octets. These parameters are chosen in order to illustrate as many nontrivial aspects of the processing as possible:

— Use of several bits per symbol (4 in this case);
— Puncturing of the convolutional code;
— Interleaving, which uses the LSB–MSB swapping stage;
— Scrambling of the pilot subcarriers.

In each Annex L table that has "Binary Val" columns, the bit positions of the binary values are specified in the header of the table.

L.1.2 The message for the BCC example

The message being encoded consists of the first 72 characters (shown in **bold** and including line breaks) of the well-known “Ode to Joy” by F. Schiller:

Joy, bright spark of divinity,
Daughter of Elysium,
Fire-insired we tread
Thy sanctuary.
Thy magic power re-unites
All that custom has divided,
All men become brothers
Under the sway of thy gentle wings.

The message is converted to ASCII; then it is prepended with an appropriate MAC header and a CRC32 is added. The resulting 100 octets PSDU is shown in Table L-1.

Table L-1—The message for the BCC example

Octet ##	Val	Val	Val	Val	Val
1...5	0x04	0x02	0x00	0x2E	0x00
6...10	0x60	0x08	0xCD	0x37	0xA6
11...15	0x00	0x20	0xD6	0x01	0x3C
16...20	0xF1	0x00	0x60	0x08	0xAD
21...25	0x3B	0xAF	0x00	0x00	0x4A
26...30	0x6F	0x79	0x2C	0x20	0x62
31...35	0x72	0x69	0x67	0x68	0x74
36...40	0x20	0x73	0x70	0x61	0x72
41...45	0x6B	0x20	0x6F	0x66	0x20
46...50	0x64	0x69	0x76	0x69	0x6E
51...55	0x69	0x74	0x79	0x2C	0x0A
56...60	0x44	0x61	0x75	0x67	0x68
61...65	0x74	0x65	0x72	0x20	0x6F
66...70	0x66	0x20	0x45	0x6C	0x79
71...75	0x73	0x69	0x75	0x6D	0x2C
76...80	0x0A	0x46	0x69	0x72	0x65
81...85	0x2D	0x69	0x6E	0x73	0x69
86...90	0x72	0x65	0x64	0x20	0x77
91...95	0x65	0x20	0x74	0x72	0x65
96...100	0x61	0x67	0x33	0x21	0xB6

L.1.3 Generation of the preamble

L.1.3.1 Generation of the short sequences

The short sequences section of the preamble is described by its frequency domain representation, given in Table L-2.

Table L-2—Frequency domain representation of the short sequences

##	Re	Im	##	Re	Im	##	Re	Im	##	Re	Im
–32	0.0	0.0	–16	1.472	1.472	0	0.0	0.0	16	1.472	1.472
–31	0.0	0.0	–15	0.0	0.0	1	0.0	0.0	17	0.0	0.0
–30	0.0	0.0	–14	0.0	0.0	2	0.0	0.0	18	0.0	0.0
–29	0.0	0.0	–13	0.0	0.0	3	0.0	0.0	19	0.0	0.0
–28	0.0	0.0	–12	–1.472	–1.472	4	–1.472	–1.472	20	1.472	1.472
–27	0.0	0.0	–11	0.0	0.0	5	0.0	0.0	21	0.0	0.0
–26	0.0	0.0	–10	0.0	0.0	6	0.0	0.0	22	0.0	0.0
–25	0.0	0.0	–9	0.0	0.0	7	0.0	0.0	23	0.0	0.0
–24	1.472	1.472	–8	–1.472	–1.472	8	–1.472	–1.472	24	1.472	1.472
–23	0.0	0.0	–7	0.0	0.0	9	0.0	0.0	25	0.0	0.0
–22	0.0	0.0	–6	0.0	0.0	10	0.0	0.0	26	0.0	0.0
–21	0.0	0.0	–5	0.0	0.0	11	0.0	0.0	27	0.0	0.0
–20	–1.472	–1.472	–4	1.472	1.472	12	1.472	1.472	28	0.0	0.0
–19	0.0	0.0	–3	0.0	0.0	13	0.0	0.0	29	0.0	0.0
–18	0.0	0.0	–2	0.0	0.0	14	0.0	0.0	30	0.0	0.0
–17	0.0	0.0	–1	0.0	0.0	15	0.0	0.0	31	0.0	0.0

One period of the IFFT on the contents of Table L-2 is given in Table L-3.

Table L-3—One period of IFFT of the short sequences

##	Re	Im	##	Re	Im	##	Re	Im	##	Re	Im
0	0.046	0.046	1	–0.132	0.002	2	–0.013	–0.079	3	0.143	–0.013
4	0.092	0.000	5	0.143	–0.013	6	–0.013	–0.079	7	–0.132	0.002
8	0.046	0.046	9	0.002	–0.132	10	–0.079	–0.013	11	–0.013	0.143
12	0.000	0.092	13	–0.013	0.143	14	–0.079	–0.013	15	0.002	–0.132
16	0.046	0.046	17	–0.132	0.002	18	–0.013	–0.079	19	0.143	–0.013
20	0.092	0.000	21	0.143	–0.013	22	–0.013	–0.079	23	–0.132	0.002

Table L-3—One period of IFFT of the short sequences *(continued)*

##	Re	Im	##	Re	Im	##	Re	Im	##	Re	Im
24	0.046	0.046	25	0.002	–0.132	26	–0.079	–0.013	27	–0.013	0.143
28	0.000	0.092	29	–0.013	0.143	30	–0.079	–0.013	31	0.002	–0.132
32	0.046	0.046	33	–0.132	0.002	34	–0.013	–0.079	35	0.143	–0.013
36	0.092	0.000	37	0.143	–0.013	38	–0.013	–0.079	39	–0.132	0.002
40	0.046	0.046	41	0.002	–0.132	42	–0.079	–0.013	43	–0.013	0.143
44	0.000	0.092	45	–0.013	0.143	46	–0.079	–0.013	47	0.002	–0.132
48	0.046	0.046	49	–0.132	0.002	50	–0.013	–0.079	51	0.143	–0.013
52	0.092	0.000	53	0.143	–0.013	54	–0.013	–0.079	55	–0.132	0.002
56	0.046	0.046	57	0.002	–0.132	58	–0.079	–0.013	59	–0.013	0.143
60	0.000	0.092	61	–0.013	0.143	62	–0.079	–0.013	63	0.002	–0.132

The single period of the short training sequence is extended periodically for 161 samples (about 8 µs), and then multiplied by the window function:

$$W(k) = \begin{bmatrix} 0.5 & \mathrm{k} = 0 \\ 1 & 1 \le k \le 159 \\ 0.5 & \mathrm{k} = 160 \end{bmatrix}$$

The last sample serves as an overlap with the following OFDM symbol. The 161 samples vector is shown in Table L-4. The time-windowing feature illustrated here is not part of the normative specifications.

Table L-4—Time domain representation of the short sequence

##	Re	Im	##	Re	Im	##	Re	Im	##	Re	Im
0	0.023	0.023	1	–0.132	0.002	2	–0.013	–0.079	3	0.143	–0.013
4	0.092	0.000	5	0.143	–0.013	6	–0.013	–0.079	7	–0.132	0.002
8	0.046	0.046	9	0.002	–0.132	10	–0.079	–0.013	11	–0.013	0.143
12	0.000	0.092	13	–0.013	0.143	14	–0.079	–0.013	15	0.002	–0.132
16	0.046	0.046	17	–0.132	0.002	18	–0.013	–0.079	19	0.143	–0.013
20	0.092	0.000	21	0.143	–0.013	22	–0.013	–0.079	23	–0.132	0.002
24	0.046	0.046	25	0.002	–0.132	26	–0.079	–0.013	27	–0.013	0.143
28	0.000	0.092	29	–0.013	0.143	30	–0.079	–0.013	31	0.002	–0.132
32	0.046	0.046	33	–0.132	0.002	34	–0.013	–0.079	35	0.143	–0.013
36	0.092	0.000	37	0.143	–0.013	38	–0.013	–0.079	39	–0.132	0.002
40	0.046	0.046	41	0.002	–0.132	42	–0.079	–0.013	43	–0.013	0.143

Table L-4—Time domain representation of the short sequence *(continued)*

##	Re	Im	##	Re	Im	##	Re	Im	##	Re	Im
44	0.000	0.092	45	–0.013	0.143	46	–0.079	–0.013	47	0.002	–0.132
48	0.046	0.046	49	–0.132	0.002	50	–0.013	–0.079	51	0.143	–0.013
52	0.092	0.000	53	0.143	–0.013	54	–0.013	–0.079	55	–0.132	0.002
56	0.046	0.046	57	0.002	–0.132	58	–0.079	–0.013	59	–0.013	0.143
60	0.000	0.092	61	–0.013	0.143	62	–0.079	–0.013	63	0.002	–0.132
64	0.046	0.046	65	–0.132	0.002	66	–0.013	–0.079	67	0.143	–0.013
68	0.092	0.000	69	0.143	–0.013	70	–0.013	–0.079	71	–0.132	0.002
72	0.046	0.046	73	0.002	–0.132	74	–0.079	–0.013	75	–0.013	0.143
76	0.000	0.092	77	–0.013	0.143	78	–0.079	–0.013	79	0.002	–0.132
80	0.046	0.046	81	–0.132	0.002	82	–0.013	–0.079	83	0.143	–0.013
84	0.092	0.000	85	0.143	–0.013	86	–0.013	–0.079	87	–0.132	0.002
88	0.046	0.046	89	0.002	–0.132	90	–0.079	–0.013	91	–0.013	0.143
92	0.000	0.092	93	–0.013	0.143	94	–0.079	–0.013	95	0.002	–0.132
96	0.046	0.046	97	–0.132	0.002	98	–0.013	–0.079	99	0.143	–0.013
100	0.092	0.000	101	0.143	–0.013	102	–0.013	–0.079	103	–0.132	0.002
104	0.046	0.046	105	0.002	–0.132	106	–0.079	–0.013	107	–0.013	0.143
108	0.000	0.092	109	–0.013	0.143	110	–0.079	–0.013	111	0.002	–0.132
112	0.046	0.046	113	–0.132	0.002	114	–0.013	–0.079	115	0.143	–0.013
116	0.092	0.000	117	0.143	–0.013	118	–0.013	–0.079	119	–0.132	0.002
120	0.046	0.046	121	0.002	–0.132	122	–0.079	–0.013	123	–0.013	0.143
124	0.000	0.092	125	–0.013	0.143	126	–0.079	–0.013	127	0.002	–0.132
128	0.046	0.046	129	–0.132	0.002	130	–0.013	–0.079	131	0.143	–0.013
132	0.092	0.000	133	0.143	–0.013	134	–0.013	–0.079	135	–0.132	0.002
136	0.046	0.046	137	0.002	–0.132	138	–0.079	–0.013	139	–0.013	0.143
140	0.000	0.092	141	–0.013	0.143	142	–0.079	–0.013	143	0.002	–0.132
144	0.046	0.046	145	–0.132	0.002	146	–0.013	–0.079	147	0.143	–0.013
148	0.092	0.000	149	0.143	–0.013	150	–0.013	–0.079	151	–0.132	0.002
152	0.046	0.046	153	0.002	–0.132	154	–0.079	–0.013	155	–0.013	0.143
156	0.000	0.092	157	–0.013	0.143	158	–0.079	–0.013	159	0.002	–0.132
160	0.023	0.023									

L.1.3.2 Generation of the long sequences

The frequency domain representation of the long training sequence part of the preamble is given in Table L-5.

Table L-5—Frequency domain representation of the long sequences

##	Re	Im		##	Re	Im		##	Re	Im		##	Re	Im
–32	0.000	0.000		–16	1.000	0.000		0	0.000	0.000		16	1.000	0.000
–31	0.000	0.000		–15	1.000	0.000		1	1.000	0.000		17	–1.000	0.000
–30	0.000	0.000		–14	1.000	0.000		2	–1.000	0.000		18	–1.000	0.000
–29	0.000	0.000		–13	1.000	0.000		3	–1.000	0.000		19	1.000	0.000
–28	0.000	0.000		–12	1.000	0.000		4	1.000	0.000		20	–1.000	0.000
–27	0.000	0.000		–11	–1.000	0.000		5	1.000	0.000		21	1.000	0.000
–26	1.000	0.000		–10	–1.000	0.000		6	–1.000	0.000		22	–1.000	0.000
–25	1.000	0.000		–9	1.000	0.000		7	1.000	0.000		23	1.000	0.000
–24	–1.000	0.000		–8	1.000	0.000		8	–1.000	0.000		24	1.000	0.000
–23	–1.000	0.000		–7	–1.000	0.000		9	1.000	0.000		25	1.000	0.000
–22	1.000	0.000		–6	1.000	0.000		10	–1.000	0.000		26	1.000	0.000
–21	1.000	0.000		–5	–1.000	0.000		11	–1.000	0.000		27	0.000	0.000
–20	–1.000	0.000		–4	1.000	0.000		12	–1.000	0.000		28	0.000	0.000
–19	1.000	0.000		–3	1.000	0.000		13	–1.000	0.000		29	0.000	0.000
–18	–1.000	0.000		–2	1.000	0.000		14	–1.000	0.000		30	0.000	0.000
–17	1.000	0.000		–1	1.000	0.000		15	1.000	0.000		31	0.000	0.000

The time domain representation is derived by performing IFFT on the contents of Table L-5, cyclically extending the result to get the cyclic prefix, and then multiplying with the window function given in L.1.3.1. The resulting 161 points vector is shown in Table L-6. The samples are appended to the short sequence section by overlapping and adding element 160 of Table L-4 to element 0 of Table L-6.

Table L-6—Time domain representation of the long sequence

##	Re	Im	##	Re	Im	##	Re	Im	##	Re	Im
0	–0.078	0.000	1	0.012	–0.098	2	0.092	–0.106	3	–0.092	–0.115
4	–0.003	–0.054	5	0.075	0.074	6	–0.127	0.021	7	–0.122	0.017
8	–0.035	0.151	9	–0.056	0.022	10	–0.060	–0.081	11	0.070	–0.014
12	0.082	–0.092	13	–0.131	–0.065	14	–0.057	–0.039	15	0.037	–0.098
16	0.062	0.062	17	0.119	0.004	18	–0.022	–0.161	19	0.059	0.015

Table L-6—Time domain representation of the long sequence *(continued)*

##	Re	Im	##	Re	Im	##	Re	Im	##	Re	Im
20	0.024	0.059	21	–0.137	0.047	22	0.001	0.115	23	0.053	–0.004
24	0.098	0.026	25	–0.038	0.106	26	–0.115	0.055	27	0.060	0.088
28	0.021	–0.028	29	0.097	–0.083	30	0.040	0.111	31	–0.005	0.120
32	0.156	0.000	33	–0.005	–0.120	34	0.040	–0.111	35	0.097	0.083
36	0.021	0.028	37	0.060	–0.088	38	–0.115	–0.055	39	–0.038	–0.106
40	0.098	–0.026	41	0.053	0.004	42	0.001	–0.115	43	–0.137	–0.047
44	0.024	–0.059	45	0.059	–0.015	46	–0.022	0.161	47	0.119	–0.004
48	0.062	–0.062	49	0.037	0.098	50	–0.057	0.039	51	–0.131	0.065
52	0.082	0.092	53	0.070	0.014	54	–0.060	0.081	55	–0.056	–0.022
56	–0.035	–0.151	57	–0.122	–0.017	58	–0.127	–0.021	59	0.075	–0.074
60	–0.003	0.054	61	–0.092	0.115	62	0.092	0.106	63	0.012	0.098
64	–0.156	0.000	65	0.012	–0.098	66	0.092	–0.106	67	–0.092	–0.115
68	–0.003	–0.054	69	0.075	0.074	70	–0.127	0.021	71	–0.122	0.017
72	–0.035	0.151	73	–0.056	0.022	74	–0.060	–0.081	75	0.070	–0.014
76	0.082	–0.092	77	–0.131	–0.065	78	–0.057	–0.039	79	0.037	–0.098
80	0.062	0.062	81	0.119	0.004	82	–0.022	–0.161	83	0.059	0.015
84	0.024	0.059	85	–0.137	0.047	86	0.001	0.115	87	0.053	–0.004
88	0.098	0.026	89	–0.038	0.106	90	–0.115	0.055	91	0.060	0.088
92	0.021	–0.028	93	0.097	–0.083	94	0.040	0.111	95	–0.005	0.120
96	0.156	0.000	97	–0.005	–0.120	98	0.040	–0.111	99	0.097	0.083
100	0.021	0.028	101	0.060	–0.088	102	–0.115	–0.055	103	–0.038	–0.106
104	0.098	–0.026	105	0.053	0.004	106	0.001	–0.115	107	–0.137	–0.047
108	0.024	–0.059	109	0.059	–0.015	110	–0.022	0.161	111	0.119	–0.004
112	0.062	–0.062	113	0.037	0.098	114	–0.057	0.039	115	–0.131	0.065
116	0.082	0.092	117	0.070	0.014	118	–0.060	0.081	119	–0.056	–0.022
120	–0.035	–0.151	121	–0.122	–0.017	122	–0.127	–0.021	123	0.075	–0.074
124	–0.003	0.054	125	–0.092	0.115	126	0.092	0.106	127	0.012	0.098
128	–0.156	0.000	129	0.012	–0.098	130	0.092	–0.106	131	–0.092	–0.115
132	–0.003	–0.054	133	0.075	0.074	134	–0.127	0.021	135	–0.122	0.017
136	–0.035	0.151	137	–0.056	0.022	138	–0.060	–0.081	139	0.070	–0.014
140	0.082	–0.092	141	–0.131	–0.065	142	–0.057	–0.039	143	0.037	–0.098
144	0.062	0.062	145	0.119	0.004	146	–0.022	–0.161	147	0.059	0.015
148	0.024	0.059	149	–0.137	0.047	150	0.001	0.115	151	0.053	–0.004

Table L-6—Time domain representation of the long sequence *(continued)*

##	Re	Im	##	Re	Im	##	Re	Im	##	Re	Im
152	0.098	0.026	153	–0.038	0.106	154	–0.115	0.055	155	0.060	0.088
156	0.021	–0.028	157	0.097	–0.083	158	0.040	0.111	159	–0.005	0.120
160	0.078	0									

L.1.4 Generation of the SIGNAL field

L.1.4.1 SIGNAL field bit assignment

The SIGNAL field bit assignment follows 18.3.4 and Figure 18-5. The transmitted bits are shown in Table L-7, where bit 0 is transmitted first.

Table L-7—Bit assignment for SIGNAL field

##	Bit	Meaning		##	Bit	Meaning
0	1	RATE: R1		12	0	—
1	0	RATE: R2		13	0	—
2	1	RATE: R3		14	0	—
3	1	RATE: R4		15	0	—
4	0	Reserved		16	0	LENGTH (MSB)
5	0	LENGTH (LSB)		17	0	Parity
6	0	—		18	0	SIGNAL TAIL
7	1	—		19	0	SIGNAL TAIL
8	0	—		20	0	SIGNAL TAIL
9	0	—		21	0	SIGNAL TAIL
10	1	—		22	0	SIGNAL TAIL
11	1	—		23	0	SIGNAL TAIL

L.1.4.2 Coding the SIGNAL field bits

The bits are encoded by the rate 1/2 convolutional encoder to yield the 48 bits given in Table L-8.

Table L-8—SIGNAL field bits after encoding

##	Bit		##	Bit		##	Bit		##	Bit		##	Bit		##	Bit
0	1		8	1		16	0		24	0		32	0		40	0
1	1		9	0		17	0		25	0		33	1		41	0
2	0		10	1		18	0		26	1		34	1		42	0
3	1		11	0		19	0		27	1		35	1		43	0
4	0		12	0		20	0		28	1		36	0		44	0
5	0		13	0		21	0		29	1		37	0		45	0
6	0		14	0		22	1		30	1		38	0		46	0
7	1		15	1		23	0		31	0		39	0		47	0

L.1.4.3 Interleaving the SIGNAL field bits

The encoded bits are interleaved according to the interleaver of 18.3.5.7. A detailed breakdown of the interleaving operation is described in L.1.6.2. The interleaved SIGNAL field bits are shown in Table L-9.

Table L-9—SIGNAL field bits after interleaving

##	Bit		##	Bit		##	Bit		##	Bit		##	Bit		##	Bit
0	1		8	1		16	0		24	1		32	0		40	1
1	0		9	1		17	0		25	0		33	0		41	0
2	0		10	0		18	0		26	0		34	1		42	0
3	1		11	1		19	1		27	0		35	0		43	1
4	0		12	0		20	0		28	0		36	0		44	0
5	1		13	0		21	1		29	0		37	1		45	1
6	0		14	0		22	0		30	1		38	0		46	0
7	0		15	0		23	0		31	1		39	0		47	0

L.1.4.4 SIGNAL field frequency domain

The encoded and interleaved bits are BPSK modulated to yield the frequency domain representation given in Table L-10. Locations –21, –7, 7, and 21 are skipped and are used for pilot insertion.

Table L-10—Frequency domain representation of SIGNAL field

##	Re	Im		##	Re	Im		##	Re	Im		##	Re	Im
–32	0.000	0.000		–16	1.000	0.000		0	0.000	0.000		16	–1.000	0.000
–31	0.000	0.000		–15	–1.000	0.000		1	1.000	0.000		17	–1.000	0.000
–30	0.000	0.000		–14	1.000	0.000		2	–1.000	0.000		18	1.000	0.000
–29	0.000	0.000		–13	–1.000	0.000		3	–1.000	0.000		19	–1.000	0.000
–28	0.000	0.000		–12	–1.000	0.000		4	–1.000	0.000		20	–1.000	0.000
–27	0.000	0.000		–11	–1.000	0.000		5	–1.000	0.000		21	X	X
–26	1.000	0.000		–10	–1.000	0.000		6	–1.000	0.000		22	1.000	0.000
–25	–1.000	0.000		–9	–1.000	0.000		7	X	X		23	–1.000	0.000
–24	–1.000	0.000		–8	–1.000	0.000		8	1.000	0.000		24	1.000	0.000
–23	1.000	0.000		–7	X	X		9	1.000	0.000		25	–1.000	0.000
–22	–1.000	0.000		–6	–1.000	0.000		10	–1.000	0.000		26	–1.000	0.000
–21	X	X		–5	1.000	0.000		11	–1.000	0.000		27	0.000	0.000
–20	1.000	0.000		–4	–1.000	0.000		12	1.000	0.000		28	0.000	0.000
–19	–1.000	0.000		–3	1.000	0.000		13	–1.000	0.000		29	0.000	0.000
–18	–1.000	0.000		–2	–1.000	0.000		14	–1.000	0.000		30	0.000	0.000
–17	1.000	0.000		–1	–1.000	0.000		15	1.000	0.000		31	0.000	0.000

Four pilot subcarriers are added by taking the values {1.0,1.0,1.0,–1.0}, multiplying them by the first element of sequence $p_{0\ldots126}$, given in Equation (18-22) (in 18.3.5.10), and inserting them into location {–21, –7,7,21}, respectively. The resulting frequency domain values are given in Table L-11.

Table L-11—Frequency domain representation of SIGNAL field with pilots inserted

##	Re	Im		##	Re	Im		##	Re	Im		##	Re	Im
–32	0.000	0.000		–16	1.000	0.000		0	0.000	0.000		16	–1.000	0.000
–31	0.000	0.000		–15	–1.000	0.000		1	1.000	0.000		17	–1.000	0.000
–30	0.000	0.000		–14	1.000	0.000		2	–1.000	0.000		18	1.000	0.000
–29	0.000	0.000		–13	–1.000	0.000		3	–1.000	0.000		19	–1.000	0.000
–28	0.000	0.000		–12	–1.000	0.000		4	–1.000	0.000		20	–1.000	0.000
–27	0.000	0.000		–11	–1.000	0.000		5	–1.000	0.000		21	–1.000	0.000
–26	1.000	0.000		–10	–1.000	0.000		6	–1.000	0.000		22	1.000	0.000
–25	–1.000	0.000		–9	–1.000	0.000		7	1.000	0.000		23	–1.000	0.000
–24	–1.000	0.000		–8	–1.000	0.000		8	1.000	0.000		24	1.000	0.000

Table L-11—Frequency domain representation of SIGNAL field with pilots inserted *(continued)*

##	Re	Im		##	Re	Im		##	Re	Im		##	Re	Im
–23	1.000	0.000		–7	1.000	0.000		9	1.000	0.000		25	–1.000	0.000
–22	–1.000	0.000		–6	–1.000	0.000		10	–1.000	0.000		26	–1.000	0.000
–21	1.000	0.000		–5	1.000	0.000		11	–1.000	0.000		27	0.000	0.000
–20	1.000	0.000		–4	–1.000	0.000		12	1.000	0.000		28	0.000	0.000
–19	–1.000	0.000		–3	1.000	0.000		13	–1.000	0.000		29	0.000	0.000
–18	–1.000	0.000		–2	–1.000	0.000		14	–1.000	0.000		30	0.000	0.000
–17	1.000	0.000		–1	–1.000	0.000		15	1.000	0.000		31	0.000	0.000

L.1.4.5 SIGNAL field time domain

The time domain representation is derived by performing IFFT on the contents of Table L-11, extending cyclically, and multiplying by the window function

$$W(k) = \begin{bmatrix} 0.5 & \mathrm{k} = 0 \\ 1 & 1 \le k \le 79 \\ 0.5 & \mathrm{k} = 80 \end{bmatrix}$$

The resulting 81 samples vector is shown in Table L-12. Note that the time-windowing feature illustrated here is not a part of the normative specifications.

Table L-12—Time domain representation of SIGNAL field

##	Re	Im	##	Re	Im	##	Re	Im	##	Re	Im
0	0.031	0.000	1	0.033	–0.044	2	–0.002	–0.038	3	–0.081	0.084
4	0.007	–0.100	5	–0.001	–0.113	6	–0.021	–0.005	7	0.136	–0.105
8	0.098	–0.044	9	0.011	–0.002	10	–0.033	0.044	11	–0.060	0.124
12	0.010	0.097	13	0.000	–0.008	14	0.018	–0.083	15	–0.069	0.027
16	–0.219	0.000	17	–0.069	–0.027	18	0.018	0.083	19	0.000	0.008
20	0.010	–0.097	21	–0.060	–0.124	22	–0.033	–0.044	23	0.011	0.002
24	0.098	0.044	25	0.136	0.105	26	–0.021	0.005	27	–0.001	0.113
28	0.007	0.100	29	–0.081	–0.084	30	–0.002	0.038	31	0.033	0.044
32	0.062	0.000	33	0.057	0.052	34	0.016	0.174	35	0.035	0.116
36	–0.051	–0.202	37	0.011	0.036	38	0.089	0.209	39	–0.049	–0.008
40	–0.035	0.044	41	0.017	–0.059	42	0.053	–0.017	43	0.099	0.100
44	0.034	–0.148	45	–0.003	–0.094	46	–0.120	0.042	47	–0.136	–0.070

Table L-12—Time domain representation of SIGNAL field *(continued)*

##	Re	Im	##	Re	Im	##	Re	Im	##	Re	Im
48	–0.031	0.000	49	–0.136	0.070	50	–0.120	–0.042	51	–0.003	0.094
52	0.034	0.148	53	0.099	–0.100	54	0.053	0.017	55	0.017	0.059
56	–0.035	–0.044	57	–0.049	0.008	58	0.089	–0.209	59	0.011	–0.036
60	–0.051	0.202	61	0.035	–0.116	62	0.016	–0.174	63	0.057	–0.052
64	0.062	0.000	65	0.033	–0.044	66	–0.002	–0.038	67	–0.081	0.084
68	0.007	–0.100	69	–0.001	–0.113	70	–0.021	–0.005	71	0.136	–0.105
72	0.098	–0.044	73	0.011	–0.002	74	–0.033	0.044	75	–0.060	0.124
76	0.010	0.097	77	0.000	–0.008	78	0.018	–0.083	79	–0.069	0.027
80	–0.109	0.000									

The SIGNAL field samples are appended with one sample overlap to the preamble, given in Table L-6.

L.1.5 Generating the DATA bits for the BCC example

L.1.5.1 Delineating, SERVICE field prepending, and zero padding

The transmitted message shown in Table L-1 contains 100 octets or, equivalently, 800 bits. The bits are prepended by the 16 SERVICE field bits and are appended by 6 tail bits. The resulting 822 bits are appended by some number of bits with value 0 to yield an integral number of OFDM symbols. For the 36 Mb/s mode, there are 144 data bits per OFDM symbol; the overall number of bits is Ceiling (822/144) × 144 = 864. Hence, 864 – 822 = 42 zero bits are appended.

The DATA bits are shown in Table L-13.

Table L-13—The DATA bits before scrambling

Bit ##	Binary Val b7 b0	Binary Val b15 b8	Binary Val b23 b16	Hex Val	Hex Val	Hex Val
000-023	00000000	00000000	00000100	0x00	0x00	0x04
024-047	00000010	00000000	00101110	0x02	0x00	0x2E
048-071	00000000	01100000	00001000	0x00	0x60	0x08
072-095	11001101	00110111	10100110	0xCD	0x37	0xA6
096-119	00000000	00100000	11010110	0x00	0x20	0xD6
120-143	00000001	00111100	11110001	0x01	0x3C	0xF1
144-167	00000000	01100000	00001000	0x00	0x60	0x08
168-191	10101101	00111011	10101111	0xAD	0x3B	0xAF
192-215	00000000	00000000	01001010	0x00	0x00	0x4A

Table L-13—The DATA bits before scrambling *(continued)*

Bit ##	Binary Val b7 b0	Binary Val b15 b8	Binary Val b23 b16	Hex Val	Hex Val	Hex Val
216-239	01101111	01111001	00101100	0x6F	0x79	0x2C
240-263	00100000	01100010	01110010	0x20	0x62	0x72
264-287	01101001	01100111	01101000	0x69	0x67	0x68
288-311	01110100	00100000	01110011	0x74	0x20	0x73
312-335	01110000	01100001	01110010	0x70	0x61	0x72
336-359	01101011	00100000	01101111	0x6B	0x20	0x6F
360-383	01100110	00100000	01100100	0x66	0x20	0x64
384-407	01101001	01110110	01101001	0x69	0x76	0x69
408-431	01101110	01101001	01110100	0x6E	0x69	0x74
432-455	01111001	00101100	00001010	0x79	0x2C	0x0A
456-479	01000100	01100001	01110101	0x44	0x61	0x75
480-503	01100111	01101000	01110100	0x67	0x68	0x74
504-527	01100101	01110010	00100000	0x65	0x72	0x20
528-551	01101111	01100110	00100000	0x6F	0x66	0x20
552-575	01000101	01101100	01111001	0x45	0x6C	0x79
576-599	01110011	01101001	01110101	0x73	0x69	0x75
600-623	01101101	00101100	00001010	0x6D	0x2C	0x0A
624-647	01000110	01101001	01110010	0x46	0x69	0x72
648-671	01100101	00101101	01101001	0x65	0x2D	0x69
672-695	01101110	01110011	01101001	0x6E	0x73	0x69
696-719	01110010	01100101	01100100	0x72	0x65	0x64
720-743	00100000	01110111	01100101	0x20	0x77	0x65
744-767	00100000	01110100	01110010	0x20	0x74	0x72
768-791	01100101	01100001	01100111	0x65	0x61	0x67
792-815	00110011	00100001	10110110	0x33	0x21	0xB6
816-839	00000000	00000000	00000000	0x00	0x00	0x00
840-863	00000000	00000000	00000000	0x00	0x00	0x00

L.1.5.2 Scrambling the BCC example

The 864 bits are scrambled by the scrambler defined in 18.3.5.5. The initial state of the scrambler is the state 1011101. The generated scrambling sequence is given in Table L-14.

Table L-14—Scrambling sequence for seed 1011101

##	Bit	##	Bit	##	Bit	##	Bit	##	Bit	##	Bit	##	Bit	##	Bit
0	0	16	1	32	0	48	1	64	0	80	0	96	0	112	1
1	1	17	0	33	1	49	1	65	1	81	0	97	0	113	0
2	1	18	1	34	1	50	1	66	1	82	1	98	1	114	0
3	0	19	0	35	0	51	1	67	1	83	1	99	0	115	1
4	1	20	1	36	1	52	0	68	0	84	1	100	0	116	1
5	1	21	0	37	0	53	1	69	0	85	0	101	1	117	0
6	0	22	0	38	0	54	0	70	0	86	1	102	0	118	0
7	0	23	1	39	0	55	0	71	1	87	1	103	0	119	0
8	0	24	1	40	0	56	1	72	1	88	1	104	0	120	1
9	0	25	1	41	1	57	0	73	1	89	1	105	0	121	0
10	0	26	0	42	0	58	1	74	1	90	0	106	0	122	1
11	1	27	0	43	1	59	0	75	1	91	0	107	0	123	1
12	1	28	1	44	0	60	0	76	1	92	1	108	1	124	1
13	0	29	1	45	1	61	0	77	1	93	0	109	0	125	0
14	0	30	1	46	0	62	1	78	0	94	1	110	0	126	1
15	1	31	1	47	1	63	1	79	0	95	1	111	0		

After scrambling, the 6 bits in location 816 (i.e., bit 817) to 821 (i.e., bit 822) are set to 0. The scrambled DATA bits are shown in Table L-15.

Table L-15—The DATA bits after scrambling

Bit ##	Binary Val b0 b7	Binary Val b8 b15	Binary Val b16 b23	Hex Val	Hex Val	Hex Val
000-023	01101100	00011001	10001001	0x6C	0x19	0x89
024-047	10001111	01101000	00100001	0x8F	0x68	0x21
048-071	11110100	10100101	01100001	0xF4	0xA5	0x61
072-095	01001111	11010111	10101110	0x4F	0xD7	0xAE
096-119	00100100	00001100	11110011	0x24	0x0C	0xF3
120-143	00111010	11100100	10111100	0x3A	0xE4	0xBC
144-167	01010011	10011000	11000000	0x53	0x98	0xC0
168-191	00011110	00110101	10110011	0x1E	0x35	0xB3
192-215	11100011	11111000	00100101	0xE3	0xF8	0x25
216-239	01100000	11010110	00100101	0x60	0xD6	0x25

Table L-15—The DATA bits after scrambling *(continued)*

Bit ##	Binary Val b0 b7	Binary Val b8 b15	Binary Val b16 b23	Hex Val	Hex Val	Hex Val
240-263	00110101	00110011	11111110	0x35	0x33	0xFE
264-287	11110000	01000001	00101011	0xF0	0x41	0x2B
288-311	10001111	01010011	00011100	0x8F	0x53	0x1C
312-335	10000011	01000001	10111110	0x83	0x41	0xBE
336-359	00111001	00101000	01100110	0x39	0x28	0x66
360-383	01000100	01100110	11001101	0x44	0x66	0xCD
384-407	11110110	10100011	11011000	0xF6	0xA3	0xD8
408-431	00001101	11010100	10000001	0x0D	0xD4	0x81
432-455	00111011	00101111	11011111	0x3B	0x2F	0xDF
456-479	11000011	01011000	11110111	0xC3	0x58	0xF7
480-503	11000110	01010010	11101011	0xC6	0x52	0xEB
504-527	01110000	10001111	10011110	0x70	0x8F	0x9E
528-551	01101010	10010000	10000001	0x6A	0x90	0x81
552-575	11111101	01111100	10101001	0xFD	0x7C	0xA9
576-599	11010001	01010101	00010010	0xD1	0x55	0x12
600-623	00000100	01110100	11011001	0x04	0x74	0xD9
624-647	11101001	00111011	11001101	0xE9	0x3B	0xCD
648-671	10010011	10001101	01111011	0x93	0x8D	0x7B
672-695	01111100	01110000	00000010	0x7C	0x70	0x02
696-719	00100000	10011001	10100001	0x20	0x99	0xA1
720-743	01111101	10001010	00100111	0x7D	0x8A	0x27
744-767	00010111	00111001	00010101	0x17	0x39	0x15
768-791	10100000	11101100	10010101	0xA0	0xEC	0x95
792-815	00010110	10010001	00010000	0x16	0x91	0x10
816-839	00000000	11011100	01111111	0x00	0xDC	0x7F
840-863	00001110	11110010	11001001	0x0E	0xF2	0xC9

L.1.6 Generating the first DATA symbol for the BCC example

L.1.6.1 Coding the DATA bits

The scrambled bits are coded with a rate 3/4 convolutional code. The DATA encoded bits are shown in Table L-16.

Table L-16—The BCC encoded DATA bits

Bit ##	Binary Val b0 b7	Binary Val b8 b15	Binary Val b16 b23	Binary Val b24 b31	Hex Val	Hex Val	Hex Val	Hex Val
0000-0031	00101011	00001000	10100001	11110000	0x2B	0x08	0xA1	0xF0
0032-0063	10011101	10110101	10011010	00011101	0x9D	0xB5	0x9A	0x1D
0064-0095	01001010	11111011	11101000	11000010	0x4A	0xFB	0xE8	0xC2
0096-0127	10001111	11000000	11001000	01110011	0x8F	0xC0	0xC8	0x73
0128-0159	11000000	01000011	11100000	00011001	0xC0	0x43	0xE0	0x19
0160-0191	11100000	11010011	11101011	10110010	0xE0	0xD3	0xEB	0xB2
0192-0223	10101111	10011000	11111101	01011001	0xAF	0x98	0xFD	0x59
0224-0255	00001111	10001011	01101001	01100110	0x0F	0x8B	0x69	0x66
0256-0287	00001100	10101010	11011001	00010000	0x0C	0xAA	0xD9	0x10
0288-0319	01010110	10001011	10100110	01000000	0x56	0x8B	0xA6	0x40
0320-0351	01100100	10110011	00100001	10011110	0x64	0xB3	0x21	0x9E
0352-0383	10001110	10010001	11000001	00000101	0x8E	0x91	0xC1	0x05
0384-0415	10110111	10110111	11000101	11011000	0xB7	0xB7	0xC5	0xD8
0416-0447	10000000	00101111	10100010	11011101	0x80	0x2F	0xA2	0xDD
0448-0479	01101111	00101011	10010111	01100001	0x6F	0x2B	0x97	0x61
0480-0511	11011001	11011101	00001101	00010010	0xD9	0xDD	0x0D	0x12
0512-0543	01110110	00100111	00000010	01001100	0x76	0x27	0x02	0x4C
0544-0575	10010010	10111100	00010010	01001011	0x92	0xBC	0x12	0x4B
0576-0607	01101010	11110111	01110000	00100011	0x6A	0xF7	0x70	0x23
0608-0639	00100111	10001110	00000001	10110100	0x27	0x8E	0x01	0xB4
0640-0671	11010110	11000011	01101010	01100000	0xD6	0xC3	0x6A	0x60
0672-0703	01001101	01001011	11001011	01010001	0x4D	0x4B	0xCB	0x51
0704-0735	10011100	10110000	10000000	11101011	0x9C	0xB0	0x80	0xEB
0736-0767	10001001	00110100	00010100	01000000	0x89	0x34	0x14	0x40
0768-0799	01101100	10011110	00101100	01010001	0x6C	0x9E	0x2C	0x51
0800-0831	01001011	01111100	01101001	00010001	0x4B	0x7C	0x69	0x11

Table L-16—The BCC encoded DATA bits *(continued)*

Bit ##	Binary Val b0 b7	Binary Val b8 b15	Binary Val b16 b23	Binary Val b24 b31	Hex Val	Hex Val	Hex Val	Hex Val
0832-0863	00010101	10000110	11111101	10111110	0x15	0x86	0xFD	0xBE
0864-0895	01011110	11111001	10111110	00101000	0x5E	0xF9	0xBE	0x28
0896-0927	11101111	11001010	01010101	00000011	0xEF	0xCA	0x55	0x03
0928-0959	11111101	00100110	10010001	00111011	0xFD	0x26	0x91	0x3B
0960-0991	10010101	11101100	01011011	00100011	0x95	0xEC	0x5B	0x23
0992-1023	10011001	01011111	00101000	00111110	0x99	0x5F	0x28	0x3E
1024-1055	11010100	11101001	11110111	10111000	0xD4	0xE9	0xF7	0xB8
1056-1087	00010011	01110101	10001110	11110010	0x13	0x75	0x8E	0xF2
1088-1119	10100000	00011011	01101100	11101001	0xA0	0x1B	0x6C	0xE9
1120-1151	00000111	01011101	10110000	10111111	0x07	0x5D	0xB0	0xBF

L.1.6.2 Interleaving the DATA bits

The interleaver is defined as a two-permutation process. The index of the coded bit before the first permutation shall be denoted by *k*; *i* shall be the index after the first and before the second permutation; and *j* shall be the index after the second permutation, just prior to modulation mapping. The mapping from *k* to *i* is shown in Table L-17, and the mapping from *i* to *j* is shown in Table L-18.

As a specific example, consider the case of *k* = 17 (the 18th bit after encoding and puncturing). It is mapped by the first permutation to *i* = 13 and by the second permutation to *j* = 12 (the 13th bit before mapping).

Table L-17—First permutation

k	*i*	*k*	*i*	*k*	*i*	*k*	*i*	*k*	*i*	*k*	*i*	*k*	*i*	*k*	*i*
0	0	24	97	48	3	72	100	96	6	120	103	144	9	168	106
1	12	25	109	49	15	73	112	97	18	121	115	145	21	169	118
2	24	26	121	50	27	74	124	98	30	122	127	146	33	170	130
3	36	27	133	51	39	75	136	99	42	123	139	147	45	171	142
4	48	28	145	52	51	76	148	100	54	124	151	148	57	172	154
5	60	29	157	53	63	77	160	101	66	125	163	149	69	173	166
6	72	30	169	54	75	78	172	102	78	126	175	150	81	174	178
7	84	31	181	55	87	79	184	103	90	127	187	151	93	175	190
8	96	32	2	56	99	80	5	104	102	128	8	152	105	176	11
9	108	33	14	57	111	81	17	105	114	129	20	153	117	177	23
10	120	34	26	58	123	82	29	106	126	130	32	154	129	178	35
11	132	35	38	59	135	83	41	107	138	131	44	155	141	179	47

Table L-17—First permutation *(continued)*

k	*i*	*k*	*i*	*k*	*i*	*k*	*i*	*k*	*i*	*k*	*i*	*k*	*i*	*k*	*i*
12	144	36	50	60	147	84	53	108	150	132	56	156	153	180	59
13	156	37	62	61	159	85	65	109	162	133	68	157	165	181	71
14	168	38	74	62	171	86	77	110	174	134	80	158	177	182	83
15	180	39	86	63	183	87	89	111	186	135	92	159	189	183	95
16	1	40	98	64	4	88	101	112	7	136	104	160	10	184	107
17	13	41	110	65	16	89	113	113	19	137	116	161	22	185	119
18	25	42	122	66	28	90	125	114	31	138	128	162	34	186	131
19	37	43	134	67	40	91	137	115	43	139	140	163	46	187	143
20	49	44	146	68	52	92	149	116	55	140	152	164	58	188	155
21	61	45	158	69	64	93	161	117	67	141	164	165	70	189	167
22	73	46	170	70	76	94	173	118	79	142	176	166	82	190	179
23	85	47	182	71	88	95	185	119	91	143	188	167	94	191	191

Table L-18—Second permutation

i	*j*	*i*	*j*	*i*	*j*	*i*	*j*	*i*	*j*	*i*	*j*	*i*	*j*	*i*	*j*
0	0	24	24	48	48	72	72	96	96	120	120	144	144	168	168
1	1	25	25	49	49	73	73	97	97	121	121	145	145	169	169
2	2	26	26	50	50	74	74	98	98	122	122	146	146	170	170
3	3	27	27	51	51	75	75	99	99	123	123	147	147	171	171
4	4	28	28	52	52	76	76	100	100	124	124	148	148	172	172
5	5	29	29	53	53	77	77	101	101	125	125	149	149	173	173
6	6	30	30	54	54	78	78	102	102	126	126	150	150	174	174
7	7	31	31	55	55	79	79	103	103	127	127	151	151	175	175
8	8	32	32	56	56	80	80	104	104	128	128	152	152	176	176
9	9	33	33	57	57	81	81	105	105	129	129	153	153	177	177
10	10	34	34	58	58	82	82	106	106	130	130	154	154	178	178
11	11	35	35	59	59	83	83	107	107	131	131	155	155	179	179
12	13	36	37	60	61	84	85	108	109	132	133	156	157	180	181
13	12	37	36	61	60	85	84	109	108	133	132	157	156	181	180
14	15	38	39	62	63	86	87	110	111	134	135	158	159	182	183
15	14	39	38	63	62	87	86	111	110	135	134	159	158	183	182
16	17	40	41	64	65	88	89	112	113	136	137	160	161	184	185

Table L-18—Second permutation *(continued)*

i	*j*	*i*	*j*	*i*	*j*	*i*	*j*	*i*	*j*	*i*	*j*	*i*	*j*	*i*	*j*
17	16	41	40	65	64	89	88	113	112	137	136	161	160	185	184
18	19	42	43	66	67	90	91	114	115	138	139	162	163	186	187
19	18	43	42	67	66	91	90	115	114	139	138	163	162	187	186
20	21	44	45	68	69	92	93	116	117	140	141	164	165	188	189
21	20	45	44	69	68	93	92	117	116	141	140	165	164	189	188
22	23	46	47	70	71	94	95	118	119	142	143	166	167	190	191
23	22	47	46	71	70	95	94	119	118	143	142	167	166	191	190

The interleaved bits are shown in Table L-19.

Table L-19—Interleaved bits of first DATA symbol

##	Bit	##	Bit	##	Bit	##	Bit	##	Bit	##	Bit
0	0	32	0	64	0	96	0	128	0	160	0
1	1	33	1	65	0	97	1	129	0	161	0
2	1	34	1	66	0	98	1	130	0	162	0
3	1	35	1	67	1	99	0	131	1	163	0
4	0	36	0	68	0	100	1	132	1	164	0
5	1	37	0	69	0	101	1	133	0	165	0
6	1	38	1	70	0	102	1	134	1	166	0
7	1	39	1	71	0	103	0	135	1	167	0
8	1	40	0	72	1	104	0	136	0	168	0
9	1	41	0	73	0	105	0	137	1	169	0
10	1	42	0	74	0	106	1	138	1	170	0
11	1	43	0	75	1	107	1	139	0	171	0
12	0	44	0	76	1	108	1	140	1	172	1
13	0	45	0	77	0	109	0	141	0	173	1
14	0	46	0	78	1	110	0	142	1	174	0
15	0	47	0	79	0	111	0	143	1	175	1
16	1	48	1	80	0	112	1	144	1	176	1
17	1	49	0	81	0	113	1	145	0	177	0
18	1	50	1	82	0	114	1	146	0	178	1
19	0	51	1	83	1	115	1	147	1	179	1

Table L-19—Interleaved bits of first DATA symbol *(continued)*

##	Bit		##	Bit		##	Bit		##	Bit		##	Bit		##	Bit
20	1		52	1		84	1		116	0		148	1		180	0
21	1		53	1		85	1		117	1		149	0		181	0
22	1		54	1		86	0		118	0		150	0		182	1
23	1		55	1		87	1		119	1		151	0		183	1
24	1		56	0		88	0		120	0		152	0		184	0
25	1		57	0		89	0		121	1		153	1		185	1
26	0		58	0		90	0		122	1		154	0		186	1
27	0		59	1		91	1		123	0		155	0		187	0
28	0		60	0		92	0		124	1		156	0		188	1
29	1		61	0		93	0		125	0		157	0		189	1
30	0		62	0		94	1		126	0		158	1		190	0
31	0		63	1		95	0		127	1		159	1		191	1

L.1.6.3 Mapping into symbols

The frequency domain symbols are generated by grouping 4 coded bits and mapping into complex 16-QAM symbols according to Table 18-10 (in 18.3.5.8). For instance, the first 4 bits (0 1 1 1) are mapped to the complex value, –0.316 + 0.316j, inserted at subcarrier #26.

Four pilot subcarriers are added by taking the values {1.0,1.0,1.0,–1.0}, multiplying them by the second element of sequence p, given in Equation (18-22) (in 18.3.5.10), and inserting them into location {–21,–7,7,21}, respectively.

The frequency domain is shown in Table L-20.

Table L-20—Frequency domain of first DATA symbol

##	Re	Im		##	Re	Im		##	Re	Im		##	Re	Im
–32	0.000	0.000		–16	–0.949	0.316		0	0.000	0.000		16	–0.316	–0.949
–31	0.000	0.000		–15	–0.949	–0.949		1	–0.316	0.949		17	–0.949	0.316
–30	0.000	0.000		–14	–0.949	–0.949		2	0.316	0.949		18	–0.949	–0.949
–29	0.000	0.000		–13	0.949	0.316		3	–0.949	0.316		19	–0.949	–0.949
–28	0.000	0.000		–12	0.316	0.316		4	0.949	–0.949		20	–0.949	–0.949
–27	0.000	0.000		–11	–0.949	–0.316		5	0.316	0.316		21	–1.000	0.000
–26	–0.316	0.316		–10	–0.949	–0.316		6	–0.316	–0.316		22	0.316	–0.316
–25	–0.316	0.316		–9	–0.949	–0.316		7	1.000	0.000		23	0.949	0.316
–24	0.316	0.316		–8	–0.949	–0.949		8	–0.316	0.949		24	–0.949	0.316

Table L-20—Frequency domain of first DATA symbol *(continued)*

##	Re	Im		##	Re	Im		##	Re	Im		##	Re	Im
–23	–0.949	–0.949		–7	1.000	0.000		9	0.949	–0.316		25	–0.316	0.949
–22	0.316	0.949		–6	0.949	–0.316		10	–0.949	–0.316		26	0.316	–0.316
–21	1.000	0.000		–5	0.949	0.949		11	0.949	0.316		27	0.000	0.000
–20	0.316	0.316		–4	–0.949	–0.316		12	–0.316	0.949		28	0.000	0.000
–19	0.316	–0.949		–3	0.316	–0.316		13	0.949	0.316		29	0.000	0.000
–18	–0.316	–0.949		–2	–0.949	–0.316		14	0.949	–0.316		30	0.000	0.000
–17	–0.316	0.316		–1	–0.949	0.949		15	0.949	–0.949		31	0.000	0.000

The time domain samples are produced by performing IFFT, cyclically extending, and multiplying with the window function in the same manner as described in L.1.4.5. The time domain samples are appended with one sample overlap to the SIGNAL field symbol.

L.1.7 Generating the additional DATA symbols

The generation of the additional five data symbols follows the same procedure as described in Clause 4. Special attention should be paid to the scrambling of the pilot subcarriers. Table L-21 lists the polarity of the pilot subcarriers and the elements of the sequence $p_{0\ldots126}$ for the DATA symbols. For completeness, the pilots in the SIGNAL are included as well. The symbols are appended one after the other with a one-sample overlap.

Table L-21—Polarity of the pilot subcarriers

i	OFDM symbol	Element of p_i	Pilot at #–21	Pilot at #–7	Pilot at #7	Pilot at #21
0	SIGNAL	1	1.0 +0 j	1.0 +0 j	1.0 +0 j	–1.0 +0 j
1	DATA 1	1	1.0 +0 j	1.0 +0 j	1.0 +0 j	–1.0 +0 j
2	DATA 2	1	1.0 +0 j	1.0 +0 j	1.0 +0 j	–1.0 +0 j
3	DATA 3	1	1.0 +0 j	1.0 +0 j	1.0 +0 j	–1.0 +0 j
4	DATA 4	–1	–1.0 +0 j	–1.0 +0 j	–1.0 +0 j	1.0 +0 j
5	DATA 5	–1	–1.0 +0 j	–1.0 +0 j	–1.0 +0 j	1.0 +0 j
6	DATA 6	–1	–1.0 +0 j	–1.0 +0 j	–1.0 +0 j	1.0 +0 j

L.1.8 The entire packet for the BCC example

The packet in its entirety is shown in the tables in this subclause. These tables illustrate the short training sequence section (Table L-22), the long training sequence section (Table L-23), the SIGNAL field

(Table L-24), and the six DATA symbols (Table L-25 to Table L-30).

Table L-22—Time domain representation of the short training sequence

##	Real	Imag	##	Real	Imag	##	Real	Imag	##	Real	Imag
0	0.023	0.023	1	–0.132	0.002	2	–0.013	–0.079	3	0.143	–0.013
4	0.092	0.000	5	0.143	–0.013	6	–0.013	–0.079	7	–0.132	0.002
8	0.046	0.046	9	0.002	–0.132	10	–0.079	–0.013	11	–0.013	0.143
12	0.000	0.092	13	–0.013	0.143	14	–0.079	–0.013	15	0.002	–0.132
16	0.046	0.046	17	–0.132	0.002	18	–0.013	–0.079	19	0.143	–0.013
20	0.092	0.000	21	0.143	–0.013	22	–0.013	–0.079	23	–0.132	0.002
24	0.046	0.046	25	0.002	–0.132	26	–0.079	–0.013	27	–0.013	0.143
28	0.000	0.092	29	–0.013	0.143	30	–0.079	–0.013	31	0.002	–0.132
32	0.046	0.046	33	–0.132	0.002	34	–0.013	–0.079	35	0.143	–0.013
36	0.092	0.000	37	0.143	–0.013	38	–0.013	–0.079	39	–0.132	0.002
40	0.046	0.046	41	0.002	–0.132	42	–0.079	–0.013	43	–0.013	0.143
44	0.000	0.092	45	–0.013	0.143	46	–0.079	–0.013	47	0.002	–0.132
48	0.046	0.046	49	–0.132	0.002	50	–0.013	–0.079	51	0.143	–0.013
52	0.092	0.000	53	0.143	–0.013	54	–0.013	–0.079	55	–0.132	0.002
56	0.046	0.046	57	0.002	–0.132	58	–0.079	–0.013	59	–0.013	0.143
60	0.000	0.092	61	–0.013	0.143	62	–0.079	–0.013	63	0.002	–0.132
64	0.046	0.046	65	–0.132	0.002	66	–0.013	–0.079	67	0.143	–0.013
68	0.092	0.000	69	0.143	–0.013	70	–0.013	–0.079	71	–0.132	0.002
72	0.046	0.046	73	0.002	–0.132	74	–0.079	–0.013	75	–0.013	0.143
76	0.000	0.092	77	–0.013	0.143	78	–0.079	–0.013	79	0.002	–0.132
80	0.046	0.046	81	–0.132	0.002	82	–0.013	–0.079	83	0.143	–0.013
84	0.092	0.000	85	0.143	–0.013	86	–0.013	–0.079	87	–0.132	0.002
88	0.046	0.046	89	0.002	–0.132	90	–0.079	–0.013	91	–0.013	0.143
92	0.000	0.092	93	–0.013	0.143	94	–0.079	–0.013	95	0.002	–0.132
96	0.046	0.046	97	–0.132	0.002	98	–0.013	–0.079	99	0.143	–0.013
100	0.092	0.000	101	0.143	–0.013	102	–0.013	–0.079	103	–0.132	0.002
104	0.046	0.046	105	0.002	–0.132	106	–0.079	–0.013	107	–0.013	0.143
108	0.000	0.092	109	–0.013	0.143	110	–0.079	–0.013	111	0.002	–0.132
112	0.046	0.046	113	–0.132	0.002	114	–0.013	–0.079	115	0.143	–0.013
116	0.092	0.000	117	0.143	–0.013	118	–0.013	–0.079	119	–0.132	0.002

Table L-22—Time domain representation of the short training sequence *(continued)*

##	Real	Imag	##	Real	Imag	##	Real	Imag	##	Real	Imag
120	0.046	0.046	121	0.002	–0.132	122	–0.079	–0.013	123	–0.013	0.143
124	0.000	0.092	125	–0.013	0.143	126	–0.079	–0.013	127	0.002	–0.132
128	0.046	0.046	129	–0.132	0.002	130	–0.013	–0.079	131	0.143	–0.013
132	0.092	0.000	133	0.143	–0.013	134	–0.013	–0.079	135	–0.132	0.002
136	0.046	0.046	137	0.002	–0.132	138	–0.079	–0.013	139	–0.013	0.143
140	0.000	0.092	141	–0.013	0.143	142	–0.079	–0.013	143	0.002	–0.132
144	0.046	0.046	145	–0.132	0.002	146	–0.013	–0.079	147	0.143	–0.013
148	0.092	0.000	149	0.143	–0.013	150	–0.013	–0.079	151	–0.132	0.002
152	0.046	0.046	153	0.002	–0.132	154	–0.079	–0.013	155	–0.013	0.143
156	0.000	0.092	157	–0.013	0.143	158	–0.079	–0.013	159	0.002	–0.132

Table L-23—Time domain representation of the long training sequence

##	Real	Imag	##	Real	Imag	##	Real	Imag	##	Real	Imag
160	–0.055	0.023	161	0.012	–0.098	162	0.092	–0.106	163	–0.092	–0.115
164	–0.003	–0.054	165	0.075	0.074	166	–0.127	0.021	167	–0.122	0.017
168	–0.035	0.151	169	–0.056	0.022	170	–0.060	–0.081	171	0.070	–0.014
172	0.082	–0.092	173	–0.131	–0.065	174	–0.057	–0.039	175	0.037	–0.098
176	0.062	0.062	177	0.119	0.004	178	–0.022	–0.161	179	0.059	0.015
180	0.024	0.059	181	–0.137	0.047	182	0.001	0.115	183	0.053	–0.004
184	0.098	0.026	185	–0.038	0.106	186	–0.115	0.055	187	0.060	0.088
188	0.021	–0.028	189	0.097	–0.083	190	0.040	0.111	191	–0.005	0.120
192	0.156	0.000	193	–0.005	–0.120	194	0.040	–0.111	195	0.097	0.083
196	0.021	0.028	197	0.060	–0.088	198	–0.115	–0.055	199	–0.038	–0.106
200	0.098	–0.026	201	0.053	0.004	202	0.001	–0.115	203	–0.137	–0.047
204	0.024	–0.059	205	0.059	–0.015	206	–0.022	0.161	207	0.119	–0.004
208	0.062	–0.062	209	0.037	0.098	210	–0.057	0.039	211	–0.131	0.065
212	0.082	0.092	213	0.070	0.014	214	–0.060	0.081	215	–0.056	–0.022
216	–0.035	–0.151	217	–0.122	–0.017	218	–0.127	–0.021	219	0.075	–0.074
220	–0.003	0.054	221	–0.092	0.115	222	0.092	0.106	223	0.012	0.098
224	–0.156	0.000	225	0.012	–0.098	226	0.092	–0.106	227	–0.092	–0.115
228	–0.003	–0.054	229	0.075	0.074	230	–0.127	0.021	231	–0.122	0.017
232	–0.035	0.151	233	–0.056	0.022	234	–0.060	–0.081	235	0.070	–0.014

Table L-23—Time domain representation of the long training sequence *(continued)*

##	Real	Imag	##	Real	Imag	##	Real	Imag	##	Real	Imag
236	0.082	–0.092	237	–0.131	–0.065	238	–0.057	–0.039	239	0.037	–0.098
240	0.062	0.062	241	0.119	0.004	242	–0.022	–0.161	243	0.059	0.015
244	0.024	0.059	245	–0.137	0.047	246	0.001	0.115	247	0.053	–0.004
248	0.098	0.026	249	–0.038	0.106	250	–0.115	0.055	251	0.060	0.088
252	0.021	–0.028	253	0.097	–0.083	254	0.040	0.111	255	–0.005	0.120
256	0.156	0.000	257	–0.005	–0.120	258	0.040	–0.111	259	0.097	0.083
260	0.021	0.028	261	0.060	–0.088	262	–0.115	–0.055	263	–0.038	–0.106
264	0.098	–0.026	265	0.053	0.004	266	0.001	–0.115	267	–0.137	–0.047
268	0.024	–0.059	269	0.059	–0.015	270	–0.022	0.161	271	0.119	–0.004
272	0.062	–0.062	273	0.037	0.098	274	–0.057	0.039	275	–0.131	0.065
276	0.082	0.092	277	0.070	0.014	278	–0.060	0.081	279	–0.056	–0.022
280	–0.035	–0.151	281	–0.122	–0.017	282	–0.127	–0.021	283	0.075	–0.074
284	–0.003	0.054	285	–0.092	0.115	286	0.092	0.106	287	0.012	0.098
288	–0.156	0.000	289	0.012	–0.098	290	0.092	–0.106	291	–0.092	–0.115
292	–0.003	–0.054	293	0.075	0.074	294	–0.127	0.021	295	–0.122	0.017
296	–0.035	0.151	297	–0.056	0.022	298	–0.060	–0.081	299	0.070	–0.014
300	0.082	–0.092	301	–0.131	–0.065	302	–0.057	–0.039	303	0.037	–0.098
304	0.062	0.062	305	0.119	0.004	306	–0.022	–0.161	307	0.059	0.015
308	0.024	0.059	309	–0.137	0.047	310	0.001	0.115	311	0.053	–0.004
312	0.098	0.026	313	–0.038	0.106	314	–0.115	0.055	315	0.060	0.088
316	0.021	–0.028	317	0.097	–0.083	318	0.040	0.111	319	–0.005	0.120

Table L-24—Time domain representation of the SIGNAL field (1 symbol)

##	Real	Imag	##	Real	Imag	##	Real	Imag	##	Real	Imag
320	0.109	0.000	321	0.033	–0.044	322	–0.002	–0.038	323	–0.081	0.084
324	0.007	–0.100	325	–0.001	–0.113	326	–0.021	–0.005	327	0.136	–0.105
328	0.098	–0.044	329	0.011	–0.002	330	–0.033	0.044	331	–0.060	0.124
332	0.010	0.097	333	0.000	–0.008	334	0.018	–0.083	335	–0.069	0.027
336	–0.219	0.000	337	–0.069	–0.027	338	0.018	0.083	339	0.000	0.008
340	0.010	–0.097	341	–0.060	–0.124	342	–0.033	–0.044	343	0.011	0.002
344	0.098	0.044	345	0.136	0.105	346	–0.021	0.005	347	–0.001	0.113
348	0.007	0.100	349	–0.081	–0.084	350	–0.002	0.038	351	0.033	0.044

Table L-24—Time domain representation of the SIGNAL field (1 symbol) *(continued)*

##	Real	Imag	##	Real	Imag	##	Real	Imag	##	Real	Imag
352	0.062	0.000	353	0.057	0.052	354	0.016	0.174	355	0.035	0.116
356	–0.051	–0.202	357	0.011	0.036	358	0.089	0.209	359	–0.049	–0.008
360	–0.035	0.044	361	0.017	–0.059	362	0.053	–0.017	363	0.099	0.100
364	0.034	–0.148	365	–0.003	–0.094	366	–0.120	0.042	367	–0.136	–0.070
368	–0.031	0.000	369	–0.136	0.070	370	–0.120	–0.042	371	–0.003	0.094
372	0.034	0.148	373	0.099	–0.100	374	0.053	0.017	375	0.017	0.059
376	–0.035	–0.044	377	–0.049	0.008	378	0.089	–0.209	379	0.011	–0.036
380	–0.051	0.202	381	0.035	–0.116	382	0.016	–0.174	383	0.057	–0.052
384	0.062	0.000	385	0.033	–0.044	386	–0.002	–0.038	387	–0.081	0.084
388	0.007	–0.100	389	–0.001	–0.113	390	–0.021	–0.005	391	0.136	–0.105
392	0.098	–0.044	393	0.011	–0.002	394	–0.033	0.044	395	–0.060	0.124
396	0.010	0.097	397	0.000	–0.008	398	0.018	–0.083	399	–0.069	0.027

Table L-25—Time domain representation of the DATA field: symbol 1of 6

##	Real	Imag	##	Real	Imag	##	Real	Imag	##	Real	Imag
400	–0.139	0.050	401	0.004	0.014	402	0.011	–0.100	403	–0.097	–0.020
404	0.062	0.081	405	0.124	0.139	406	0.104	–0.015	407	0.173	–0.140
408	–0.040	0.006	409	–0.133	0.009	410	–0.002	–0.043	411	–0.047	0.092
412	–0.109	0.082	413	–0.024	0.010	414	0.096	0.019	415	0.019	–0.023
416	–0.087	–0.049	417	0.002	0.058	418	–0.021	0.228	419	–0.103	0.023
420	–0.019	–0.175	421	0.018	0.132	422	–0.071	0.160	423	–0.153	–0.062
424	–0.107	0.028	425	0.055	0.140	426	0.070	0.103	427	–0.056	0.025
428	–0.043	0.002	429	0.016	–0.118	430	0.026	–0.071	431	0.033	0.177
432	0.020	–0.021	433	0.035	–0.088	434	–0.008	0.101	435	–0.035	–0.010
436	0.065	0.030	437	0.092	–0.034	438	0.032	–0.123	439	–0.018	0.092
440	0.000	–0.006	441	–0.006	–0.056	442	–0.019	0.040	443	0.053	–0.131
444	0.022	–0.133	445	0.104	–0.032	446	0.163	–0.045	447	–0.105	–0.030
448	–0.110	–0.069	449	–0.008	–0.092	450	–0.049	–0.043	451	0.085	–0.017
452	0.090	0.063	453	0.015	0.153	454	0.049	0.094	455	0.011	0.034
456	–0.012	0.012	457	–0.015	–0.017	458	–0.061	0.031	459	–0.070	–0.040
460	0.011	–0.109	461	0.037	–0.060	462	–0.003	–0.178	463	–0.007	–0.128
464	–0.059	0.100	465	0.004	0.014	466	0.011	–0.100	467	–0.097	–0.020

Table L-25—Time domain representation of the DATA field: symbol 1of 6 *(continued)*

##	Real	Imag	##	Real	Imag	##	Real	Imag	##	Real	Imag
468	0.062	0.081	469	0.124	0.139	470	0.104	–0.015	471	0.173	–0.140
472	–0.040	0.006	473	–0.133	0.009	474	–0.002	–0.043	475	–0.047	0.092
476	–0.109	0.082	477	–0.024	0.010	478	0.096	0.019	479	0.019	–0.023

Table L-26—Time domain representation of the DATA field: symbol 2 of 6

##	Real	Imag	##	Real	Imag	##	Real	Imag	##	Real	Imag
480	–0.058	0.016	481	–0.096	–0.045	482	–0.110	0.003	483	–0.070	0.216
484	–0.040	0.059	485	0.010	–0.056	486	0.034	0.065	487	0.117	0.033
488	0.078	–0.133	489	–0.043	–0.146	490	0.158	–0.071	491	0.254	–0.021
492	0.068	0.117	493	–0.044	0.114	494	–0.035	0.041	495	0.085	0.070
496	0.120	0.010	497	0.057	0.055	498	0.063	0.188	499	0.091	0.149
500	–0.017	–0.039	501	–0.078	–0.075	502	0.049	0.079	503	–0.014	–0.007
504	0.030	–0.027	505	0.080	0.054	506	–0.186	–0.067	507	–0.039	–0.027
508	0.043	–0.072	509	–0.092	–0.089	510	0.029	0.105	511	–0.144	0.003
512	–0.069	–0.041	513	0.132	0.057	514	–0.126	0.070	515	–0.031	0.109
516	0.161	–0.009	517	0.056	–0.046	518	–0.004	0.028	519	–0.049	0.000
520	–0.078	–0.005	521	0.015	–0.087	522	0.149	–0.104	523	–0.021	–0.051
524	–0.154	–0.106	525	0.024	0.030	526	0.046	0.123	527	–0.004	–0.098
528	–0.061	–0.128	529	–0.024	–0.038	530	0.066	–0.048	531	–0.067	0.027
532	0.054	–0.050	533	0.171	–0.049	534	–0.108	0.132	535	–0.161	–0.019
536	–0.070	–0.072	537	–0.177	0.049	538	–0.172	–0.050	539	0.051	–0.075
540	0.122	–0.057	541	0.009	–0.044	542	–0.012	–0.021	543	0.004	0.009
544	–0.030	0.081	545	–0.096	–0.045	546	–0.110	0.003	547	–0.070	0.216
548	–0.040	0.059	549	0.010	–0.056	550	0.034	0.065	551	0.117	0.033
552	0.078	–0.133	553	–0.043	–0.146	554	0.158	–0.071	555	0.254	–0.021
556	0.068	0.117	557	–0.044	0.114	558	–0.035	0.041	559	0.085	0.070

Table L-27—Time domain representation of the DATA field: symbol 3 of 6

##	Real	Imag	##	Real	Imag	##	Real	Imag	##	Real	Imag
560	0.001	0.011	561	–0.099	–0.048	562	0.054	–0.196	563	0.124	0.035
564	0.092	0.045	565	–0.037	–0.066	566	–0.021	–0.004	567	0.042	–0.065
568	0.061	0.048	569	0.046	0.004	570	–0.063	–0.045	571	–0.102	0.152
572	–0.039	–0.019	573	–0.005	–0.106	574	0.083	0.031	575	0.226	0.028
576	0.140	–0.010	577	–0.132	–0.033	578	–0.116	0.088	579	0.023	0.052
580	–0.171	–0.080	581	–0.246	–0.025	582	–0.062	–0.038	583	–0.055	–0.062
584	–0.004	–0.060	585	0.034	0.000	586	–0.030	0.021	587	0.075	–0.122
588	0.043	–0.080	589	–0.022	0.041	590	0.026	0.013	591	–0.031	–0.018
592	0.059	0.008	593	0.109	0.078	594	0.002	0.101	595	–0.016	0.054
596	–0.059	0.070	597	0.017	0.114	598	0.104	–0.034	599	–0.024	–0.059
600	–0.081	0.051	601	–0.040	–0.069	602	–0.069	0.058	603	–0.067	0.117
604	0.007	–0.131	605	0.009	0.028	606	0.075	0.117	607	0.118	0.030
608	–0.041	0.148	609	0.005	0.098	610	0.026	0.002	611	–0.116	0.045
612	–0.020	0.084	613	0.101	0.006	614	0.205	–0.064	615	0.073	–0.063
616	–0.174	–0.118	617	–0.024	0.026	618	–0.041	0.129	619	–0.042	–0.053
620	0.148	–0.126	621	–0.030	–0.049	622	–0.015	–0.021	623	0.089	–0.069
624	–0.119	0.011	625	–0.099	–0.048	626	0.054	–0.196	627	0.124	0.035
628	0.092	0.045	629	–0.037	–0.066	630	–0.021	–0.004	631	0.042	–0.065
632	0.061	0.048	633	0.046	0.004	634	–0.063	–0.045	635	–0.102	0.152
636	–0.039	–0.019	637	–0.005	–0.106	638	0.083	0.031	639	0.226	0.028

Table L-28—Time domain representation of the DATA field: symbol 4 of 6

##	Real	Imag	##	Real	Imag	##	Real	Imag	##	Real	Imag
640	0.085	–0.065	641	0.034	–0.142	642	0.004	–0.012	643	0.126	–0.043
644	0.055	0.068	645	–0.020	0.077	646	0.008	–0.056	647	–0.034	0.046
648	–0.040	–0.134	649	–0.056	–0.131	650	0.014	0.097	651	0.045	–0.009
652	–0.113	–0.170	653	–0.065	–0.230	654	0.065	–0.011	655	0.011	0.048
656	–0.091	–0.059	657	–0.110	0.024	658	0.074	–0.034	659	0.124	0.022
660	–0.037	0.071	661	0.015	0.002	662	0.028	0.099	663	–0.062	0.068
664	0.064	0.016	665	0.078	0.156	666	0.009	0.219	667	0.147	0.024
668	0.106	0.030	669	–0.080	0.143	670	–0.049	–0.100	671	–0.036	–0.082
672	–0.089	0.021	673	–0.070	–0.029	674	–0.086	0.048	675	–0.066	–0.015

Table L-28—Time domain representation of the DATA field: symbol 4 of 6 *(continued)*

##	Real	Imag	##	Real	Imag	##	Real	Imag	##	Real	Imag
676	–0.024	0.002	677	–0.030	–0.023	678	–0.032	0.020	679	–0.002	0.212
680	0.158	–0.024	681	0.141	–0.119	682	–0.146	0.058	683	–0.155	0.083
684	–0.002	–0.030	685	0.018	–0.129	686	0.012	–0.018	687	–0.008	–0.037
688	0.031	0.040	689	0.023	0.097	690	0.014	–0.039	691	0.050	0.019
692	–0.072	–0.141	693	–0.023	–0.051	694	0.024	0.099	695	–0.127	–0.116
696	0.094	0.102	697	0.183	0.098	698	–0.040	–0.020	699	0.065	0.077
700	0.088	–0.147	701	–0.039	–0.059	702	–0.057	0.124	703	–0.077	0.020
704	0.030	–0.120	705	0.034	–0.142	706	0.004	–0.012	707	0.126	–0.043
708	0.055	0.068	709	–0.020	0.077	710	0.008	–0.056	711	–0.034	0.046
712	–0.040	–0.134	713	–0.056	–0.131	714	0.014	0.097	715	0.045	–0.009
716	–0.113	–0.170	717	–0.065	–0.230	718	0.065	–0.011	719	0.011	0.048
720	–0.026	–0.021	721	–0.002	0.041	722	0.001	0.071	723	–0.037	–0.117

Table L-29—Time domain representation of the DATA field: symbol 5 of 6

##	Real	Imag	##	Real	Imag	##	Real	Imag	##	Real	Imag
724	–0.106	–0.062	725	0.002	0.057	726	–0.008	–0.011	727	0.019	0.072
728	0.016	0.059	729	–0.065	–0.077	730	0.142	–0.062	731	0.087	0.025
732	–0.003	–0.103	733	0.107	–0.152	734	–0.054	0.036	735	–0.030	–0.003
736	0.058	–0.020	737	–0.028	0.007	738	–0.027	–0.099	739	0.049	–0.075
740	0.174	0.031	741	0.134	0.156	742	0.060	0.077	743	–0.010	–0.022
744	–0.084	0.040	745	–0.074	0.011	746	–0.163	0.054	747	–0.052	–0.008
748	0.076	–0.042	749	0.043	0.101	750	0.058	–0.018	751	0.003	–0.090
752	0.059	–0.018	753	0.023	–0.031	754	0.007	–0.017	755	0.066	–0.017
756	–0.135	–0.098	757	–0.056	–0.081	758	0.089	0.154	759	0.120	0.122
760	0.102	0.001	761	–0.141	0.102	762	0.006	–0.011	763	0.057	–0.039
764	–0.059	0.066	765	0.132	0.111	766	0.012	0.114	767	0.047	–0.106
768	0.160	–0.099	769	–0.076	0.084	770	–0.049	0.073	771	0.005	–0.086
772	–0.052	–0.108	773	–0.073	0.129	774	–0.129	–0.034	775	–0.153	–0.111
776	–0.193	0.098	777	–0.107	–0.068	778	0.004	–0.009	779	–0.039	0.024

Table L-29—Time domain representation of the DATA field: symbol 5 of 6 *(continued)*

##	Real	Imag	##	Real	Imag	##	Real	Imag	##	Real	Imag
780	–0.054	–0.079	781	0.024	0.084	782	0.052	–0.002	783	0.028	–0.044
784	0.040	0.018	785	–0.002	0.041	786	0.001	0.071	787	–0.037	–0.117
788	–0.106	–0.062	789	0.002	0.057	790	–0.008	–0.011	791	0.019	0.072
792	0.016	0.059	793	–0.065	–0.077	794	0.142	–0.062	795	0.087	0.025
796	–0.003	–0.103	797	0.107	–0.152	798	–0.054	0.036	799	–0.030	–0.003

Table L-30—Time domain representation of the DATA field: symbol 6 of 6

##	Real	Imag	##	Real	Imag	##	Real	Imag	##	Real	Imag
800	0.029	–0.026	801	–0.047	0.077	802	–0.007	–0.002	803	0.050	–0.021
804	0.046	–0.040	805	–0.061	–0.099	806	–0.121	0.008	807	0.014	0.050
808	0.145	0.034	809	0.001	–0.046	810	–0.058	–0.121	811	0.040	0.001
812	–0.029	0.041	813	0.002	–0.066	814	0.015	–0.054	815	0.010	–0.029
816	0.008	–0.119	817	–0.134	0.002	818	0.064	0.079	819	0.095	–0.102
820	–0.069	–0.014	821	0.156	0.037	822	0.047	–0.008	823	–0.076	0.025
824	0.117	–0.143	825	0.056	–0.042	826	0.002	0.075	827	–0.039	–0.058
828	–0.092	0.014	829	–0.041	0.047	830	–0.058	0.092	831	0.012	0.154
832	0.079	0.091	833	–0.067	0.017	834	–0.102	–0.032	835	0.039	0.084
836	–0.036	0.014	837	–0.001	–0.046	838	0.195	0.131	839	0.039	0.067
840	–0.007	0.045	841	0.051	0.008	842	–0.074	–0.109	843	–0.033	0.070
844	–0.028	0.176	845	–0.041	0.045	846	0.014	–0.084	847	0.054	–0.040
848	0.110	–0.020	849	0.014	–0.021	850	0.006	0.139	851	0.008	0.011
852	–0.060	–0.040	853	0.008	0.179	854	0.008	0.020	855	0.044	–0.114
856	0.021	–0.015	857	–0.008	–0.052	858	0.091	–0.109	859	–0.025	–0.040
860	–0.049	0.006	861	–0.043	–0.041	862	–0.178	–0.026	863	–0.073	–0.057
864	0.000	–0.031	865	–0.047	0.077	866	–0.007	–0.002	867	0.050	–0.021
868	0.046	–0.040	869	–0.061	–0.099	870	–0.121	0.008	871	0.014	0.050
872	0.145	0.034	873	0.001	–0.046	874	–0.058	–0.121	875	0.040	0.001
876	–0.029	0.041	877	0.002	–0.066	878	0.015	–0.054	879	0.010	–0.029
880	0.004	–0.059									

L.2 Generating encoded DATA bits—LDPC example 1

LDPC example 1 is similar to the BCC example. This example illustrates LDPC shortening, encoding, and puncturing of a single codeword.

Input TXVECTOR parameters for LDPC example 1:

- FEC_CODING = LDPC_CODING = 1 (LDPC encoder; not BCC)
- CH_BANDWIDTH = HT_CBW20 = 0 (CH_BANDWIDTH = 0 => 20 MHz)
- MCS = 4 (MCS = 4; QAM 16; Coding rate = 3/4)
- Coding rate R = 3/4
- LENGTH = 100 octets (with 16-bit SERVICE field becomes 102 Octets = 816 bits to scramble and encode)
- STBC = 0 (STBC = 0 => OFF; m_STBC=1)

L.2.1 The message for LDPC example 1

The message being encoded consists of the first 72 characters (shown in **bold** and including line breaks) of the well-known “Ode to Joy” by F. Schiller:

Joy, bright spark of divinity,
Daughter of Elysium,
Fire-insired we tread
Thy sanctuary.
Thy magic power re-unites
All that custom has divided,
All men become brothers
Under the sway of thy gentle wings.

The message is converted to ASCII; then it is prepended with an appropriate MAC header, and a CRC32 is added. The resulting 100 octets PSDU is shown in Table L-31.

NOTE 1—The message for LDPC example 1 is identical to the message for the BCC example; in other words, the FCS field (octets 97–100) has the same CRC 32 value.

NOTE 2—The DurationID field (i.e., octets 3 and 4) remains 0x02E = 46 µs.

Table L-31—Message for LDPC example 1

Octet ##	Value	Value	Value	Value	Value
1...5	0x04	0x02	0x00	0x2E	0x00
6...10	0x60	0x08	0xCD	0x37	0xA6
11...15	0x00	0x20	0xD6	0x01	0x3C
16...20	0xF1	0x00	0x60	0x08	0xAD
21...25	0x3B	0xAF	0x00	0x00	0x4A
26...30	0x6F	0x79	0x2C	0x20	0x62
31...35	0x72	0x69	0x67	0x68	0x74
36...40	0x20	0x73	0x70	0x61	0x72
41...45	0x6B	0x20	0x6F	0x66	0x20

Table L-31—Message for LDPC example 1 *(continued)*

Octet ##	Value	Value	Value	Value	Value
46...50	0x64	0x69	0x76	0x69	0x6E
51...55	0x69	0x74	0x79	0x2C	0x0A
56...60	0x44	0x61	0x75	0x67	0x68
61...65	0x74	0x65	0x72	0x20	0x6F
66...70	0x66	0x20	0x45	0x6C	0x79
71...75	0x73	0x69	0x75	0x6D	0x2C
76...80	0x0A	0x46	0x69	0x72	0x65
81...85	0x2D	0x69	0x6E	0x73	0x69
86...90	0x72	0x65	0x64	0x20	0x77
91...95	0x65	0x20	0x74	0x72	0x65
96...100	0x61	0x67	0x33	0x21	0xB6

L.2.2 Prepending the SERVICE field for LDPC example 1

The transmitted message shown in Table L-31 contains 100 octets, or equivalently, 800 bits. The bits are prepended by the 16 SERVICE field bits (bits 0–15 in Table L-32), as defined in 20.3.11.2, but tail bits and padding bits are not appended as in the BCC example. The resulting 816 bits are shown in Table L-32.

Table L-32—DATA bits for LDPC example 1 before scrambling

Bit ##	Binary value b7 b0	Binary value b15 b8	Binary value b23 b16	Hex value	Hex value	Hex value
000–023	00000000	00000000	00000100	0x00	0x00	0x04
024–047	00000010	00000000	00101110	0x02	0x00	0x2E
048–071	00000000	01100000	00001000	0x00	0x60	0x08
072–095	11001101	00110111	10100110	0xCD	0x37	0xA6
096–119	00000000	00100000	11010110	0x00	0x20	0xD6
120–143	00000001	00111100	11110001	0x01	0x3C	0xF1
144–167	00000000	01100000	00001000	0x00	0x60	0x08
168–191	10101101	00111011	10101111	0xAD	0x3B	0xAF
192–215	00000000	00000000	01001010	0x00	0x00	0x4A
216–239	01101111	01111001	00101100	0x6F	0x79	0x2C
240–263	00100000	01100010	01110010	0x20	0x62	0x72
264–287	01101001	01100111	01101000	0x69	0x67	0x68
288–311	01110100	00100000	01110011	0x74	0x20	0x73

Table L-32—DATA bits for LDPC example 1 before scrambling *(continued)*

Bit ##	Binary value b7 b0	Binary value b15 b8	Binary value b23 b16	Hex value	Hex value	Hex value
312–335	01110000	01100001	01110010	0x70	0x61	0x72
336–359	01101011	00100000	01101111	0x6B	0x20	0x6F
360–383	01100110	00100000	01100100	0x66	0x20	0x64
384–407	01101001	01110110	01101001	0x69	0x76	0x69
408–431	01101110	01101001	01110100	0x6E	0x69	0x74
432–455	01111001	00101100	00001010	0x79	0x2C	0x0A
456–479	01000100	01100001	01110101	0x44	0x61	0x75
480–503	01100111	01101000	01110100	0x67	0x68	0x74
504–527	01100101	01110010	00100000	0x65	0x72	0x20
528–551	01101111	01100110	00100000	0x6F	0x66	0x20
552–575	01000101	01101100	01111001	0x45	0x6C	0x79
576–599	01110011	01101001	01110101	0x73	0x69	0x75
600–623	01101101	00101100	00001010	0x6D	0x2C	0x0A
624–647	01000110	01101001	01110010	0x46	0x69	0x72
648–671	01100101	00101101	01101001	0x65	0x2D	0x69
672–695	01101110	01110011	01101001	0x6E	0x73	0x69
696–719	01110010	01100101	01100100	0x72	0x65	0x64
720–743	00100000	01110111	01100101	0x20	0x77	0x65
744–767	00100000	01110100	01110010	0x20	0x74	0x72
768–791	01100101	01100001	01100111	0x65	0x61	0x67
792–815	00110011	00100001	10110110	0x33	0x21	0xB6

L.2.3 Scrambling LDPC example 1

The 816 bits are scrambled by the scrambler defined in 18.3.5.5. The initial state of the scrambler is the state 1011101 binary (0x5D hexadecimal). The scrambled sequence is given in Table L-33.

NOTE—The scrambled entries for the correct CRC32 value are given in bits 784–815.

Table L-33—DATA bits for LDPC example 1 after scrambling

Bit ##	Binary value b0 b7	Binary value b8 b15	Binary value b16 b23	Hex value	Hex value	Hex value
000–023	01101100	00011001	10001001	0x6C	0x19	0x89
024–047	10001111	01101000	00100001	0x8F	0x68	0x21
048–071	11110100	10100101	01100001	0xF4	0xA5	0x61
072–095	01001111	11010111	10101110	0x4F	0xD7	0xAE
096–119	00100100	00001100	11110011	0x24	0x0C	0xF3
120–143	00111010	11100100	10111100	0x3A	0xE4	0xBC
144–167	01010011	10011000	11000000	0x53	0x98	0xC0
168–191	00011110	00110101	10110011	0x1E	0x35	0xB3
192–215	11100011	11111000	00100101	0xE3	0xF8	0x25
216–239	01100000	11010110	00100101	0x60	0xD6	0x25
240–263	00110101	00110011	11111110	0x35	0x33	0xFE
264–287	11110000	01000001	00101011	0xF0	0x41	0x2B
288–311	10001111	01010011	00011100	0x8F	0x53	0x1C
312–335	10000011	01000001	10111110	0x83	0x41	0xBE
336–359	00111001	00101000	01100110	0x39	0x28	0x66
360–383	01000100	01100110	11001101	0x44	0x66	0xCD
384–407	11110110	10100011	11011000	0xF6	0xA3	0xD8
408–431	00001101	11010100	10000001	0x0D	0xD4	0x81
432–455	00111011	00101111	11011111	0x3B	0x2F	0xDF
456–479	11000011	01011000	11110111	0xC3	0x58	0xF7
480–503	11000110	01010010	11101011	0xC6	0x52	0xEB
504–527	01110000	10001111	10011110	0x70	0x8F	0x9E
528–551	01101010	10010000	10000001	0x6A	0x90	0x81
552–575	11111101	01111100	10101001	0xFD	0x7C	0xA9
576–599	11010001	01010101	00010010	0xD1	0x55	0x12
600–623	00000100	01110100	11011001	0x04	0x74	0xD9
624–647	11101001	00111011	11001101	0xE9	0x3B	0xCD
648–671	10010011	10001101	01111011	0x93	0x8D	0x7B

Table L-33—DATA bits for LDPC example 1 after scrambling *(continued)*

Bit ##	Binary value b0 b7	Binary value b8 b15	Binary value b16 b23	Hex value	Hex value	Hex value
672–695	01111100	01110000	00000010	0x7C	0x70	0x02
696–719	00100000	10011001	10100001	0x20	0x99	0xA1
720–743	01111101	10001010	00100111	0x7D	0x8A	0x27
744–767	00010111	00111001	00010101	0x17	0x39	0x15
768–791	10100000	11101100	10010101	0xA0	0xEC	0x95
792–815	00010110	10010001	00010000	0x16	0x91	0x10

L.2.4 Inserting shortening bits for LDPC example 1

The equations of 20.3.11.7.5 are solved to calculate the following derived parameters for LDPC example 1 from the input TXVECTOR parameters:

— $N_{CW} = 1$ (number of codewords)
— $L_{LDPC} = 1944$ (size of codeword)
— $N_{CBPS} = 208$ (number of coded bits per symbol)
— $N_{avbits} = 1248$ (number of available bits)
— $N_{shrt} = 642$ (number of bits to be shortened)
— $N_{punc} = 54$ (number of bits to be punctured)
— $N_{SYM} = 6$ (number of OFDM symbols)
— $N_{rep} = 0$ (number of bits to be repeated)

The results of applying shortening bits, as prescribed in paragraph (c) of 20.3.11.7.5, is given in Table L-34.

NOTE—$N_{shrt} = 642$ shortening bits have been inserted as 0s in bits 816–1457.

Table L-34—DATA bits for LDPC example 1 after insertion of shortening bits

Bit ##	Binary value b0 b7	Binary value b8 b15	Binary value b16 b23	Hex value	Hex value	Hex value
0000–0023	01101100	00011001	10001001	0x6C	0x19	0x89
0024–0047	10001111	01101000	00100001	0x8F	0x68	0x21
0048–0071	11110100	10100101	01100001	0xF4	0xA5	0x61
0072–0095	01001111	11010111	10101110	0x4F	0xD7	0xAE
0096–0119	00100100	00001100	11110011	0x24	0x0C	0xF3
0120–0143	00111010	11100100	10111100	0x3A	0xE4	0xBC
0144–0167	01010011	10011000	11000000	0x53	0x98	0xC0
0168–0191	00011110	00110101	10110011	0x1E	0x35	0xB3
0192–0215	11100011	11111000	00100101	0xE3	0xF8	0x25

Table L-34—DATA bits for LDPC example 1 after insertion of shortening bits *(continued)*

Bit ##	Binary value b0 b7	Binary value b8 b15	Binary value b16 b23	Hex value	Hex value	Hex value
0216–0239	01100000	11010110	00100101	0x60	0xD6	0x25
0240–0263	00110101	00110011	11111110	0x35	0x33	0xFE
0264–0287	11110000	01000001	00101011	0xF0	0x41	0x2B
0288–0311	10001111	01010011	00011100	0x8F	0x53	0x1C
0312–0335	10000011	01000001	10111110	0x83	0x41	0xBE
0336–0359	00111001	00101000	01100110	0x39	0x28	0x66
0360–0383	01000100	01100110	11001101	0x44	0x66	0xCD
0384–0407	11110110	10100011	11011000	0xF6	0xA3	0xD8
0408–0431	00001101	11010100	10000001	0x0D	0xD4	0x81
0432–0455	00111011	00101111	11011111	0x3B	0x2F	0xDF
0456–0479	11000011	01011000	11110111	0xC3	0x58	0xF7
0480–0503	11000110	01010010	11101011	0xC6	0x52	0xEB
0504–0527	01110000	10001111	10011110	0x70	0x8F	0x9E
0528–0551	01101010	10010000	10000001	0x6A	0x90	0x81
0552–0575	11111101	01111100	10101001	0xFD	0x7C	0xA9
0576–0599	11010001	01010101	00010010	0xD1	0x55	0x12
0600–0623	00000100	01110100	11011001	0x04	0x74	0xD9
0624–0647	11101001	00111011	11001101	0xE9	0x3B	0xCD
0648–0671	10010011	10001101	01111011	0x93	0x8D	0x7B
0672–0695	01111100	01110000	00000010	0x7C	0x70	0x02
0696–0719	00100000	10011001	10100001	0x20	0x99	0xA1
0720–0743	01111101	10001010	00100111	0x7D	0x8A	0x27
0744–0767	00010111	00111001	00010101	0x17	0x39	0x15
0768–0791	10100000	11101100	10010101	0xA0	0xEC	0x95
0792–0815	00010110	10010001	00010000	0x16	0x91	0x10
0816–0839	00000000	00000000	00000000	0x00	0x00	0x00
0840–0863	00000000	00000000	00000000	0x00	0x00	0x00
0864–0887	00000000	00000000	00000000	0x00	0x00	0x00
0888–0911	00000000	00000000	00000000	0x00	0x00	0x00
0912–0935	00000000	00000000	00000000	0x00	0x00	0x00
0936–0959	00000000	00000000	00000000	0x00	0x00	0x00
0960–0983	00000000	00000000	00000000	0x00	0x00	0x00

Table L-34—DATA bits for LDPC example 1 after insertion of shortening bits *(continued)*

Bit ##	Binary value b0 b7	Binary value b8 b15	Binary value b16 b23	Hex value	Hex value	Hex value
0984–1007	00000000	00000000	00000000	0x00	0x00	0x00
1008–1031	00000000	00000000	00000000	0x00	0x00	0x00
1032–1055	00000000	00000000	00000000	0x00	0x00	0x00
1056–1079	00000000	00000000	00000000	0x00	0x00	0x00
1080–1103	00000000	00000000	00000000	0x00	0x00	0x00
1104–1127	00000000	00000000	00000000	0x00	0x00	0x00
1128–1151	00000000	00000000	00000000	0x00	0x00	0x00
1152–1175	00000000	00000000	00000000	0x00	0x00	0x00
1176–1199	00000000	00000000	00000000	0x00	0x00	0x00
1200–1223	00000000	00000000	00000000	0x00	0x00	0x00
1224–1247	00000000	00000000	00000000	0x00	0x00	0x00
1248–1271	00000000	00000000	00000000	0x00	0x00	0x00
1272–1295	00000000	00000000	00000000	0x00	0x00	0x00
1296–1319	00000000	00000000	00000000	0x00	0x00	0x00
1320–1343	00000000	00000000	00000000	0x00	0x00	0x00
1344–1367	00000000	00000000	00000000	0x00	0x00	0x00
1368–1391	00000000	00000000	00000000	0x00	0x00	0x00
1392–1415	00000000	00000000	00000000	0x00	0x00	0x00
1416–1439	00000000	00000000	00000000	0x00	0x00	0x00
1440–1457	00000000	00000000	00 - - - - - -	0x00	0x00	0x0-

L.2.5 Encoding data for LDPC example 1

The DATA with shortening bits are LDPC encoded as a single (N_{CW}=1) codeword (L_{LDPC}=944; R=3/4) as prescribed by paragraph (c) of 20.3.11.7.5. The results are given in Table L-35.

NOTE—The LDPC encoder appends 486 bits (i.e., bits 1458–1943) after the shortening bits.

Table L-35—DATA bits for LDPC example 1 after LDPC encoding

Bit ##	Binary value b0 b7	Binary value b8 b15	Binary value b16 b23	Hex value	Hex value	Hex value
0000–0023	01101100	00011001	10001001	0x6C	0x19	0x89
0024–0047	10001111	01101000	00100001	0x8F	0x68	0x21
0048–0071	11110100	10100101	01100001	0xF4	0xA5	0x61

Table L-35—DATA bits for LDPC example 1 after LDPC encoding *(continued)*

Bit ##	Binary value b0 b7	Binary value b8 b15	Binary value b16 b23	Hex value	Hex value	Hex value
0072–0095	01001111	11010111	10101110	0x4F	0xD7	0xAE
0096–0119	00100100	00001100	11110011	0x24	0x0C	0xF3
0120–0143	00111010	11100100	10111100	0x3A	0xE4	0xBC
0144–0167	01010011	10011000	11000000	0x53	0x98	0xC0
0168–0191	00011110	00110101	10110011	0x1E	0x35	0xB3
0192–0215	11100011	11111000	00100101	0xE3	0xF8	0x25
0216–0239	01100000	11010110	00100101	0x60	0xD6	0x25
0240–0263	00110101	00110011	11111110	0x35	0x33	0xFE
0264–0287	11110000	01000001	00101011	0xF0	0x41	0x2B
0288–0311	10001111	01010011	00011100	0x8F	0x53	0x1C
0312–0335	10000011	01000001	10111110	0x83	0x41	0xBE
0336–0359	00111001	00101000	01100110	0x39	0x28	0x66
0360–0383	01000100	01100110	11001101	0x44	0x66	0xCD
0384–0407	11110110	10100011	11011000	0xF6	0xA3	0xD8
0408–0431	00001101	11010100	10000001	0x0D	0xD4	0x81
0432–0455	00111011	00101111	11011111	0x3B	0x2F	0xDF
0456–0479	11000011	01011000	11110111	0xC3	0x58	0xF7
0480–0503	11000110	01010010	11101011	0xC6	0x52	0xEB
0504–0527	01110000	10001111	10011110	0x70	0x8F	0x9E
0528–0551	01101010	10010000	10000001	0x6A	0x90	0x81
0552–0575	11111101	01111100	10101001	0xFD	0x7C	0xA9
0576–0599	11010001	01010101	00010010	0xD1	0x55	0x12
0600–0623	00000100	01110100	11011001	0x04	0x74	0xD9
0624–0647	11101001	00111011	11001101	0xE9	0x3B	0xCD
0648–0671	10010011	10001101	01111011	0x93	0x8D	0x7B
0672–0695	01111100	01110000	00000010	0x7C	0x70	0x02
0696–0719	00100000	10011001	10100001	0x20	0x99	0xA1
0720–0743	01111101	10001010	00100111	0x7D	0x8A	0x27
0744–0767	00010111	00111001	00010101	0x17	0x39	0x15
0768–0791	10100000	11101100	10010101	0xA0	0xEC	0x95
0792–0815	00010110	10010001	00010000	0x16	0x91	0x10
0816–0839	00000000	00000000	00000000	0x00	0x00	0x00
0840–0863	00000000	00000000	00000000	0x00	0x00	0x00

Table L-35—DATA bits for LDPC example 1 after LDPC encoding *(continued)*

Bit ##	Binary value b0 b7	Binary value b8 b15	Binary value b16 b23	Hex value	Hex value	Hex value
0864–0887	00000000	00000000	00000000	0x00	0x00	0x00
0888–0911	00000000	00000000	00000000	0x00	0x00	0x00
0912–0935	00000000	00000000	00000000	0x00	0x00	0x00
0936–0959	00000000	00000000	00000000	0x00	0x00	0x00
0960–0983	00000000	00000000	00000000	0x00	0x00	0x00
0984–1007	00000000	00000000	00000000	0x00	0x00	0x00
1008–1031	00000000	00000000	00000000	0x00	0x00	0x00
1032–1055	00000000	00000000	00000000	0x00	0x00	0x00
1056–1079	00000000	00000000	00000000	0x00	0x00	0x00
1080–1103	00000000	00000000	00000000	0x00	0x00	0x00
1104–1127	00000000	00000000	00000000	0x00	0x00	0x00
1128–1151	00000000	00000000	00000000	0x00	0x00	0x00
1152–1175	00000000	00000000	00000000	0x00	0x00	0x00
1176–1199	00000000	00000000	00000000	0x00	0x00	0x00
1200–1223	00000000	00000000	00000000	0x00	0x00	0x00
1224–1247	00000000	00000000	00000000	0x00	0x00	0x00
1248–1271	00000000	00000000	00000000	0x00	0x00	0x00
1272–1295	00000000	00000000	00000000	0x00	0x00	0x00
1296–1319	00000000	00000000	00000000	0x00	0x00	0x00
1320–1343	00000000	00000000	00000000	0x00	0x00	0x00
1344–1367	00000000	00000000	00000000	0x00	0x00	0x00
1368–1391	00000000	00000000	00000000	0x00	0x00	0x00
1392–1415	00000000	00000000	00000000	0x00	0x00	0x00
1416–1439	00000000	00000000	00000000	0x00	0x00	0x00
1440–1463	00000000	00000000	00100110	0x00	0x00	0x26
1464–1487	00111101	10101001	10011100	0x3D	0xA9	0x9C
1488–1511	01000000	11010111	10110010	0x40	0xD7	0xB2
1512–1535	10000110	11100011	10111111	0x86	0xE3	0xBF
1536–1559	01000011	10100101	11011001	0x43	0xA5	0xD9
1560–1583	00001101	00000110	11010110	0x0D	0x06	0xD6
1584–1607	01100000	11110100	00011111	0x60	0xF4	0x1F
1608–1631	00110001	00001100	00010011	0x31	0x0C	0x13
1632–1655	01110110	00001111	10011111	0x76	0x0F	0x9F

Table L-35—DATA bits for LDPC example 1 after LDPC encoding *(continued)*

Bit ##	Binary value b0 b7	Binary value b8 b15	Binary value b16 b23	Hex value	Hex value	Hex value
1656–1679	11011010	10011111	10101001	0xDA	0x9F	0xA9
1680–1703	01110100	01011001	11011100	0x74	0x59	0xDC
1704–1727	10001001	11110010	11100010	0x89	0xF2	0xE2
1728–1751	11011000	01101000	10100001	0xD8	0x68	0xA1
1752–1775	01100011	00011101	10100101	0x63	0x1D	0xA5
1776–1799	10100110	10000000	11010001	0xA6	0x80	0xD1
1800–1823	10001001	01010111	11011100	0x89	0x57	0xDC
1824–1847	10110011	01011101	00110011	0xB3	0x5D	0x33
1848–1871	01110000	11011100	10110010	0x70	0xDC	0xB2
1872–1895	11110110	00111001	00111101	0xF6	0x39	0x3D
1896–1919	00100011	10011011	00110110	0x23	0x9B	0x36
1920–1943	00111110	00010101	00010001	0x3E	0x15	0x11

L.2.6 Removing shortening bits and puncturing for LDPC example 1

The shortening bits, applied before LDPC encoding, are now removed as prescribed in paragraph (c) of 20.3.11.7.5. Finally, either puncturing is applied as described in paragraph (d) of the same subclause, or the copying of repeated bits is applied as described in paragraph (e) of the same subclause. In LDPC example 1, because N_{punc} = 54 is nonzero and N_{rep} = 0, puncturing is prescribed and completes the LDPC encoding process.

The results are given in Table L-36.

NOTE—The N_{shrt} = 642 shortening bits have been removed, and the N_{punc} = 54 bits have been punctured from Table L-35 to produce bits 816–1247 of Table L-36.

Table L-36—DATA bits after puncturing and removal of shortening bits

Bit ##	Binary value b0 b7	Binary value b8 b15	Binary value b16 b23	Hex value	Hex value	Hex value
0000–0023	01101100	00011001	10001001	0x6C	0x19	0x89
0024–0047	10001111	01101000	00100001	0x8F	0x68	0x21
0048–0071	11110100	10100101	01100001	0xF4	0xA5	0x61
0072–0095	01001111	11010111	10101110	0x4F	0xD7	0xAE
0096–0119	00100100	00001100	11110011	0x24	0x0C	0xF3
0120–0143	00111010	11100100	10111100	0x3A	0xE4	0xBC
0144–0167	01010011	10011000	11000000	0x53	0x98	0xC0
0168–0191	00011110	00110101	10110011	0x1E	0x35	0xB3

Table L-36—DATA bits after puncturing and removal of shortening bits *(continued)*

Bit ##	Binary value b0 b7	Binary value b8 b15	Binary value b16 b23	Hex value	Hex value	Hex value
0192–0215	11100011	11111000	00100101	0xE3	0xF8	0x25
0216–0239	01100000	11010110	00100101	0x60	0xD6	0x25
0240–0263	00110101	00110011	11111110	0x35	0x33	0xFE
0264–0287	11110000	01000001	00101011	0xF0	0x41	0x2B
0288–0311	10001111	01010011	00011100	0x8F	0x53	0x1C
0312–0335	10000011	01000001	10111110	0x83	0x41	0xBE
0336–0359	00111001	00101000	01100110	0x39	0x28	0x66
0360–0383	01000100	01100110	11001101	0x44	0x66	0xCD
0384–0407	11110110	10100011	11011000	0xF6	0xA3	0xD8
0408–0431	00001101	11010100	10000001	0x0D	0xD4	0x81
0432–0455	00111011	00101111	11011111	0x3B	0x2F	0xDF
0456–0479	11000011	01011000	11110111	0xC3	0x58	0xF7
0480–0503	11000110	01010010	11101011	0xC6	0x52	0xEB
0504–0527	01110000	10001111	10011110	0x70	0x8F	0x9E
0528–0551	01101010	10010000	10000001	0x6A	0x90	0x81
0552–0575	11111101	01111100	10101001	0xFD	0x7C	0xA9
0576–0599	11010001	01010101	00010010	0xD1	0x55	0x12
0600–0623	00000100	01110100	11011001	0x04	0x74	0xD9
0624–0647	11101001	00111011	11001101	0xE9	0x3B	0xCD
0648–0671	10010011	10001101	01111011	0x93	0x8D	0x7B
0672–0695	01111100	01110000	00000010	0x7C	0x70	0x02
0696–0719	00100000	10011001	10100001	0x20	0x99	0xA1
0720–0743	01111101	10001010	00100111	0x7D	0x8A	0x27
0744–0767	00010111	00111001	00010101	0x17	0x39	0x15
0768–0791	10100000	11101100	10010101	0xA0	0xEC	0x95
0792–0815	00010110	10010001	00010000	0x16	0x91	0x10
0816–0839	10011000	11110110	10100110	0x98	0xF6	0xA6
0840–0863	01110001	00000011	01011110	0x71	0x03	0x5E
0864–0887	11001010	00011011	10001110	0xCA	0x1B	0x8E
0888–0911	11111101	00001110	10010111	0xFD	0x0E	0x97
0912–0935	01100100	00110100	00011011	0x64	0x34	0x1B
0936–0959	01011001	10000011	11010000	0x59	0x83	0xD0
0960–0983	01111100	11000100	00110000	0x7C	0xC4	0x30

Table L-36—DATA bits after puncturing and removal of shortening bits *(continued)*

Bit ##	Binary value b0 b7	Binary value b8 b15	Binary value b16 b23	Hex value	Hex value	Hex value
0984–1007	01001101	11011000	00111110	0x4D	0xD8	0x3E
1008–1031	01111111	01101010	01111110	0x7F	0x6A	0x7E
1032–1055	10100101	11010001	01100111	0xA5	0xD1	0x67
1056–1079	01110010	00100111	11001011	0x72	0x27	0xCB
1080–1103	10001011	01100001	10100010	0x8B	0x61	0xA2
1104–1127	10000101	10001100	01110110	0x85	0x8C	0x76
1128–1151	10010110	10011010	00000011	0x96	0x9A	0x03
1152–1175	01000110	00100101	01011111	0x46	0x25	0x5F
1176–1199	01110010	11001101	01110100	0x72	0xCD	0x74
1200–1223	11001101	11000011	01110010	0xCD	0xC3	0x72
1224–1247	11001011	11011000	11100100	0xCB	0xD8	0xE4

L.3 Generating encoded DATA bits—LDPC example 2

LDPC example 2 exercises the alternative branches of the LDPC encoding procedure not exercised in LDPC example 1. Example 2 also exhibits LDPC shortening, encoding, and padding by repetition and employs multiple codewords and diversifies the TXVECTOR parameters—all without making the length of this example cumbersome.

The length of the text of the message is increased by 40 octets, from 72 characters to 112 characters, in order to illustrate padding (rather than puncturing) and encoding of multiple codewords.

Input TXVECTOR parameters for LDPC example 2:

— FEC_CODING = LDPC_CODING = 1	(LDPC encoder; not BCC)
— CH_BANDWIDTH = HT_CBW40 = 1	(CH_BANDWIDTH = 1 => 40 MHz)
— MCS = 1	(MCS = 1; QPSK; coding rate = 1/2)
— Coding rate R = 1/2	
— LENGTH = 140 octets	(with 16-bit SERVICE field becomes 142 octets = 1136 bits to scramble and encode)
— STBC = 1	(STBC = 1 => ON; m_STBC = 2)

L.3.1 The message for LDPC example 2

The message being encoded consists of the first 112 characters (shown in **bold** and including line breaks) of the well-known “Ode to Joy” by F. Schiller:

Joy, bright spark of divinity,
Daughter of Elysium,
Fire-insired we tread
Thy sanctuary.

Thy magic power re-unites
All that custom has divided,
All men become brothers
Under the sway of thy gentle wings.

The message is converted to ASCII; then it is prepended with an appropriate MAC header and a CRC32 is added. The resulting 140 octets PSDU is shown in Table L-37.

Because of the additional 40 characters, note that the message for LDPC example 2 has a different FCS field (octets 137–140) than the previous examples and that the DurationID field (i.e., octets 3 and 4) changes to 0x036 = 54 µs.

Table L-37—Message for LDPC example 2

Octet ##	Value	Value	Value	Value	Value
1...5	0x04	0x02	0x00	0x36	0x00
6...10	0x60	0x08	0xCD	0x37	0xA6
11...15	0x00	0x20	0xD6	0x01	0x3C
16...20	0xF1	0x00	0x60	0x08	0xAD
21...25	0x3B	0xAF	0x00	0x00	0x4A
26...30	0x6F	0x79	0x2C	0x20	0x62
31...35	0x72	0x69	0x67	0x68	0x74
36...40	0x20	0x73	0x70	0x61	0x72
41...45	0x6B	0x20	0x6F	0x66	0x20
46...50	0x64	0x69	0x76	0x69	0x6E
51...55	0x69	0x74	0x79	0x2C	0x0A
56...60	0x44	0x61	0x75	0x67	0x68
61...65	0x74	0x65	0x72	0x20	0x6F
66...70	0x66	0x20	0x45	0x6C	0x79
71...75	0x73	0x69	0x75	0x6D	0x2C
76...80	0x0A	0x46	0x69	0x72	0x65
81...85	0x2D	0x69	0x6E	0x73	0x69
86...90	0x72	0x65	0x64	0x20	0x77
91...95	0x65	0x20	0x74	0x72	0x65
96...100	0x61	0x64	0x0A	0x54	0x68
101...105	0x79	0x20	0x73	0x61	0x6E
106...110	0x63	0x74	0x75	0x61	0x72
111...115	0x79	0x2E	0x0A	0x54	0x68
116...120	0x79	0x20	0x6D	0x61	0x67
121...125	0x69	0x63	0x20	0x70	0x6F

Table L-37—Message for LDPC example 2 *(continued)*

Octet ##	Value	Value	Value	Value	Value
126...130	0x77	0x65	0x72	0x20	0x72
131...135	0x65	0x2D	0x75	0x6E	0x69
136...140	0x74	0x3B	0xDB	0xB5	0x22

L.3.2 Prepending the SERVICE field for LDPC example 2

The transmitted message shown in Table L-37 contains 140 octets, or equivalently, 1120 bits. The bits are prepended by the 16 SERVICE field bits (bits 0–15 in Table L-38), as defined by 20.3.11.2, but tail bits and padding bits are not appended as in the BCC example. The resulting 1136 bits are shown in Table L-38.

Table L-38—DATA bits for LDPC example 2 before scrambling

Bit ##	Binary value b7 b0	Binary value b15 b8	Binary value b23 b16	Hex value	Hex value	Hex value
0000–0023	00000000	00000000	00000100	0x00	0x00	0x04
0024–0047	00000010	00000000	00110110	0x02	0x00	0x36
0048–0071	00000000	01100000	00001000	0x00	0x60	0x08
0072–0095	11001101	00110111	10100110	0xCD	0x37	0xA6
0096–0119	00000000	00100000	11010110	0x00	0x20	0xD6
0120–0143	00000001	00111100	11110001	0x01	0x3C	0xF1
0144–0167	00000000	01100000	00001000	0x00	0x60	0x08
0168–0191	10101101	00111011	10101111	0xAD	0x3B	0xAF
0192–0215	00000000	00000000	01001010	0x00	0x00	0x4A
0216–0239	01101111	01111001	00101100	0x6F	0x79	0x2C
0240–0263	00100000	01100010	01110010	0x20	0x62	0x72
0264–0287	01101001	01100111	01101000	0x69	0x67	0x68
0288–0311	01110100	00100000	01110011	0x74	0x20	0x73
0312–0335	01110000	01100001	01110010	0x70	0x61	0x72
0336–0359	01101011	00100000	01101111	0x6B	0x20	0x6F
0360–0383	01100110	00100000	01100100	0x66	0x20	0x64
0384–0407	01101001	01110110	01101001	0x69	0x76	0x69
0408–0431	01101110	01101001	01110100	0x6E	0x69	0x74
0432–0455	01111001	00101100	00001010	0x79	0x2C	0x0A
0456–0479	01000100	01100001	01110101	0x44	0x61	0x75
0480–0503	01100111	01101000	01110100	0x67	0x68	0x74

Table L-38—DATA bits for LDPC example 2 before scrambling *(continued)*

Bit ##	Binary value b7 b0	Binary value b15 b8	Binary value b23 b16	Hex value	Hex value	Hex value
0504–0527	01100101	01110010	00100000	0x65	0x72	0x20
0528–0551	01101111	01100110	00100000	0x6F	0x66	0x20
0552–0575	01000101	01101100	01111001	0x45	0x6C	0x79
0576–0599	01110011	01101001	01110101	0x73	0x69	0x75
0600–0623	01101101	00101100	00001010	0x6D	0x2C	0x0A
0624–0647	01000110	01101001	01110010	0x46	0x69	0x72
0648–0671	01100101	00101101	01101001	0x65	0x2D	0x69
0672–0695	01101110	01110011	01101001	0x6E	0x73	0x69
0696–0719	01110010	01100101	01100100	0x72	0x65	0x64
0720–0743	00100000	01110111	01100101	0x20	0x77	0x65
0744–0767	00100000	01110100	01110010	0x20	0x74	0x72
0768–0791	01100101	01100001	01100100	0x65	0x61	0x64
0792–0815	00001010	01010100	01101000	0x0A	0x54	0x68
0816–0839	01111001	00100000	01110011	0x79	0x20	0x73
0840–0863	01100001	01101110	01100011	0x61	0x6E	0x63
0864–0887	01110100	01110101	01100001	0x74	0x75	0x61
0888–0911	01110010	01111001	00101110	0x72	0x79	0x2E
0912–0935	00001010	01010100	01101000	0x0A	0x54	0x68
0936–0959	01111001	00100000	01101101	0x79	0x20	0x6D
0960–0983	01100001	01100111	01101001	0x61	0x67	0x69
0984–1007	01100011	00100000	01110000	0x63	0x20	0x70
1008–1031	01101111	01110111	01100101	0x6F	0x77	0x65
1032–1055	01110010	00100000	01110010	0x72	0x20	0x72
1056–1079	01100101	00101101	01110101	0x65	0x2D	0x75
1080–1103	01101110	01101001	01110100	0x6E	0x69	0x74
1104–1127	00111011	11011011	10110101	0x3B	0xDB	0xB5
1128–1135	00100010	--------	--------	0x22	----	----

L.3.3 Scrambling LDPC example 2

The 1136 bits are scrambled by the scrambler defined in 18.3.5.5. The initial state of the scrambler is the state 1011101 binary (0x5D hexadecimal). The scrambled sequence is given in Table L-39.

Table L-39—DATA bits for LDPC example 2 after scrambling

Bit ##	Binary value b0 b7	Binary value b8 b15	Binary value b16 b23	Hex value	Hex value	Hex value
0000–0023	01101100	00011001	10001001	0x6C	0x19	0x89
0024–0047	10001111	01101000	00111001	0x8F	0x68	0x39
0048–0071	11110100	10100101	01100001	0xF4	0xA5	0x61
0072–0095	01001111	11010111	10101110	0x4F	0xD7	0xAE
0096–0119	00100100	00001100	11110011	0x24	0x0C	0xF3
0120–0143	00111010	11100100	10111100	0x3A	0xE4	0xBC
0144–0167	01010011	10011000	11000000	0x53	0x98	0xC0
0168–0191	00011110	00110101	10110011	0x1E	0x35	0xB3
0192–0215	11100011	11111000	00100101	0xE3	0xF8	0x25
0216–0239	01100000	11010110	00100101	0x60	0xD6	0x25
0240–0263	00110101	00110011	11111110	0x35	0x33	0xFE
0264–0287	11110000	01000001	00101011	0xF0	0x41	0x2B
0288–0311	10001111	01010011	00011100	0x8F	0x53	0x1C
0312–0335	10000011	01000001	10111110	0x83	0x41	0xBE
0336–0359	00111001	00101000	01100110	0x39	0x28	0x66
0360–0383	01000100	01100110	11001101	0x44	0x66	0xCD
0384–0407	11110110	10100011	11011000	0xF6	0xA3	0xD8
0408–0431	00001101	11010100	10000001	0x0D	0xD4	0x81
0432–0455	00111011	00101111	11011111	0x3B	0x2F	0xDF
0456–0479	11000011	01011000	11110111	0xC3	0x58	0xF7
0480–0503	11000110	01010010	11101011	0xC6	0x52	0xEB
0504–0527	01110000	10001111	10011110	0x70	0x8F	0x9E
0528–0551	01101010	10010000	10000001	0x6A	0x90	0x81
0552–0575	11111101	01111100	10101001	0xFD	0x7C	0xA9
0576–0599	11010001	01010101	00010010	0xD1	0x55	0x12
0600–0623	00000100	01110100	11011001	0x04	0x74	0xD9
0624–0647	11101001	00111011	11001101	0xE9	0x3B	0xCD
0648–0671	10010011	10001101	01111011	0x93	0x8D	0x7B
0672–0695	01111100	01110000	00000010	0x7C	0x70	0x02
0696–0719	00100000	10011001	10100001	0x20	0x99	0xA1
0720–0743	01111101	10001010	00100111	0x7D	0x8A	0x27
0744–0767	00010111	00111001	00010101	0x17	0x39	0x15
0768–0791	10100000	11101100	01010101	0xA0	0xEC	0x55

Table L-39—DATA bits for LDPC example 2 after scrambling *(continued)*

Bit ##	Binary value b0 b7	Binary value b8 b15	Binary value b16 b23	Hex value	Hex value	Hex value
0792–0815	10001010	00111111	01101011	0x8A	0x3F	0x6B
0816–0839	10110110	11011000	10110001	0xB6	0xD8	0xB1
0840–0863	10001000	10000100	00001111	0x88	0x84	0x0F
0864–0887	00101100	10001000	10101000	0x2C	0x88	0xA8
0888–0911	11111000	10010010	10100000	0xF8	0x92	0xA0
0912–0935	10110111	10011110	00111100	0xB7	0x9E	0x3C
0936–0959	01100100	01010101	00001110	0x64	0x55	0x0E
0960–0983	01111000	11111011	01110011	0x78	0xFB	0x73
0984–1007	01010100	00000000	01000010	0x54	0x00	0x42
1008–1031	10101011	10000010	10111111	0xAB	0x82	0xBF
1032–1055	11100111	11001011	00100110	0xE7	0xCB	0x26
1056–1079	11110011	01000000	00001101	0xF3	0x40	0x0D
1080–1103	00000111	01101010	00010101	0x07	0x6A	0x15
1104–1127	00010111	11111111	10100101	0x17	0xFF	0xA5
1128–1135	11011100	--------	--------	0xDC	----	----

L.3.4 Inserting the shortening bits for LDPC example 2

The equations of 20.3.11.7.5 are solved to calculate the following derived parameters for LDPC example 2 from the input TXVECTOR parameters:

- $N_{CW} = 2$ (number of codewords)
- $L_{LDPC} = 1296$ (size of codeword)
- $N_{CBPS} = 216$ (number of coded bits per symbol)
- $N_{avbits} = 2592$ (number of available bits)
- $N_{shrt} = 160$ (number of bits to be shortened)
- $N_{punc} = 0$ (number of bits to be punctured)
- $N_{SYM} = 12$ (number of OFDM symbols)
- $N_{rep} = 160$ (number of bits to be repeated)

The results of applying shortening bits, as prescribed in paragraph (c) of 20.3.11.7.5, is given in Table L-40.

NOTE—N_{shrt} = 160 shortening bits have been inserted as 0s: 80 zeros at bits 568–647 and 80 zeros at bits 1216–1295; this order equally distributes the shortening bits across the N_{CW} = 2 codewords.

Table L-40—DATA bits for LDPC example 2 after insertion of shortening bits

Bit ##	Binary value b0 b7	Binary value b8 b15	Binary value b16 b23	Hex value	Hex value	Hex value
0000–0023	01101100	00011001	10001001	0x6C	0x19	0x89
0024–0047	10001111	01101000	00111001	0x8F	0x68	0x39
0048–0071	11110100	10100101	01100001	0xF4	0xA5	0x61
0072–0095	01001111	11010111	10101110	0x4F	0xD7	0xAE
0096–0119	00100100	00001100	11110011	0x24	0x0C	0xF3
0120–0143	00111010	11100100	10111100	0x3A	0xE4	0xBC
0144–0167	01010011	10011000	11000000	0x53	0x98	0xC0
0168–0191	00011110	00110101	10110011	0x1E	0x35	0xB3
0192–0215	11100011	11111000	00100101	0xE3	0xF8	0x25
0216–0239	01100000	11010110	00100101	0x60	0xD6	0x25
0240–0263	00110101	00110011	11111110	0x35	0x33	0xFE
0264–0287	11110000	01000001	00101011	0xF0	0x41	0x2B
0288–0311	10001111	01010011	00011100	0x8F	0x53	0x1C
0312–0335	10000011	01000001	10111110	0x83	0x41	0xBE
0336–0359	00111001	00101000	01100110	0x39	0x28	0x66
0360–0383	01000100	01100110	11001101	0x44	0x66	0xCD
0384–0407	11110110	10100011	11011000	0xF6	0xA3	0xD8
0408–0431	00001101	11010100	10000001	0x0D	0xD4	0x81
0432–0455	00111011	00101111	11011111	0x3B	0x2F	0xDF
0456–0479	11000011	01011000	11110111	0xC3	0x58	0xF7
0480–0503	11000110	01010010	11101011	0xC6	0x52	0xEB
0504–0527	01110000	10001111	10011110	0x70	0x8F	0x9E
0528–0551	01101010	10010000	10000001	0x6A	0x90	0x81
0552–0575	11111101	01111100	00000000	0xFD	0x7C	0x00
0576–0599	00000000	00000000	00000000	0x00	0x00	0x00
0600–0623	00000000	00000000	00000000	0x00	0x00	0x00
0624–0647	00000000	00000000	00000000	0x00	0x00	0x00
0648–0671	10101001	11010001	01010101	0xA9	0xD1	0x55
0672–0695	00010010	00000100	01110100	0x12	0x04	0x74

Table L-40—DATA bits for LDPC example 2 after insertion of shortening bits *(continued)*

Bit ##	Binary value b0 b7	Binary value b8 b15	Binary value b16 b23	Hex value	Hex value	Hex value
0696–0719	11011001	11101001	00111011	0xD9	0xE9	0x3B
0720–0743	11001101	10010011	10001101	0xCD	0x93	0x8D
0744–0767	01111011	01111100	01110000	0x7B	0x7C	0x70
0768–0791	00000010	00100000	10011001	0x02	0x20	0x99
0792–0815	10100001	01111101	10001010	0xA1	0x7D	0x8A
0816–0839	00100111	00010111	00111001	0x27	0x17	0x39
0840–0863	00010101	10100000	11101100	0x15	0xA0	0xEC
0864–0887	01010101	10001010	00111111	0x55	0x8A	0x3F
0888–0911	01101011	10110110	11011000	0x6B	0xB6	0xD8
0912–0935	10110001	10001000	10000100	0xB1	0x88	0x84
0936–0959	00001111	00101100	10001000	0x0F	0x2C	0x88
0960–0983	10101000	11111000	10010010	0xA8	0xF8	0x92
0984–1007	10100000	10110111	10011110	0xA0	0xB7	0x9E
1008–1031	00111100	01100100	01010101	0x3C	0x64	0x55
1032–1055	00001110	01111000	11111011	0x0E	0x78	0xFB
1056–1079	01110011	01010100	00000000	0x73	0x54	0x00
1080–1103	01000010	10101011	10000010	0x42	0xAB	0x82
1104–1127	10111111	11100111	11001011	0xBF	0xE7	0xCB
1128–1151	00100110	11110011	01000000	0x26	0xF3	0x40
1152–1175	00001101	00000111	01101010	0x0D	0x07	0x6A
1176–1199	00010101	00010111	11111111	0x15	0x17	0xFF
1200–1223	10100101	11011100	00000000	0xA5	0xDC	0x00
1224–1247	00000000	00000000	00000000	0x00	0x00	0x00
1248–1271	00000000	00000000	00000000	0x00	0x00	0x00
1272–1295	00000000	00000000	00000000	0x00	0x00	0x00

L.3.5 Encoding the data for LDPC example 2

The DATA with shortening bits are LDPC encoded as two ($N_{CW} = 2$) codewords ($L_{LDPC} = 1296$; R = 1/2) as prescribed by paragraph (c) of 20.3.11.7.5. The results are given in Table L-41.

NOTE—The LDPC encoder appends 648 bits as follows: bits 648–1295 after the first shortened codeword and another 648 bits (bits 1944–2591) after the second shortened codeword.

Table L-41—DATA bits for LDPC example 2 after LDPC encoding

Bit ##	Binary value b0 b7	Binary value b8 b15	Binary value b16 b23	Hex value	Hex value	Hex value
0000–0023	01101100	00011001	10001001	0x6C	0x19	0x89
0024–0047	10001111	01101000	00111001	0x8F	0x68	0x39
0048–0071	11110100	10100101	01100001	0xF4	0xA5	0x61
0072–0095	01001111	11010111	10101110	0x4F	0xD7	0xAE
0096–0119	00100100	00001100	11110011	0x24	0x0C	0xF3
0120–0143	00111010	11100100	10111100	0x3A	0xE4	0xBC
0144–0167	01010011	10011000	11000000	0x53	0x98	0xC0
0168–0191	00011110	00110101	10110011	0x1E	0x35	0xB3
0192–0215	11100011	11111000	00100101	0xE3	0xF8	0x25
0216–0239	01100000	11010110	00100101	0x60	0xD6	0x25
0240–0263	00110101	00110011	11111110	0x35	0x33	0xFE
0264–0287	11110000	01000001	00101011	0xF0	0x41	0x2B
0288–0311	10001111	01010011	00011100	0x8F	0x53	0x1C
0312–0335	10000011	01000001	10111110	0x83	0x41	0xBE
0336–0359	00111001	00101000	01100110	0x39	0x28	0x66
0360–0383	01000100	01100110	11001101	0x44	0x66	0xCD
0384–0407	11110110	10100011	11011000	0xF6	0xA3	0xD8
0408–0431	00001101	11010100	10000001	0x0D	0xD4	0x81
0432–0455	00111011	00101111	11011111	0x3B	0x2F	0xDF
0456–0479	11000011	01011000	11110111	0xC3	0x58	0xF7
0480–0503	11000110	01010010	11101011	0xC6	0x52	0xEB
0504–0527	01110000	10001111	10011110	0x70	0x8F	0x9E
0528–0551	01101010	10010000	10000001	0x6A	0x90	0x81
0552–0575	11111101	01111100	00000000	0xFD	0x7C	0x00
0576–0599	00000000	00000000	00000000	0x00	0x00	0x00
0600–0623	00000000	00000000	00000000	0x00	0x00	0x00
0624–0647	00000000	00000000	00000000	0x00	0x00	0x00
0648–0671	00001001	11000001	11111011	0x09	0xC1	0xFB
0672–0695	01101000	11001101	00000101	0x68	0xCD	0x05
0696–0719	10110110	11000111	01100101	0xB6	0xC7	0x65
0720–0743	10100101	10011001	11100000	0xA5	0x99	0xE0
0744–0767	01110011	01110000	01101101	0x73	0x70	0x6D
0768–0791	01011110	01111001	11100011	0x5E	0x79	0xE3

Table L-41—DATA bits for LDPC example 2 after LDPC encoding *(continued)*

Bit ##	Binary value b0 b7	Binary value b8 b15	Binary value b16 b23	Hex value	Hex value	Hex value
0792–0815	01100111	00100111	01011110	0x67	0x27	0x5E
0816–0839	10010101	10101000	11110110	0x95	0xA8	0xF6
0840–0863	00110101	01001000	10100111	0x35	0x48	0xA7
0864–0887	00100110	00101001	00110001	0x26	0x29	0x31
0888–0911	00101110	00011001	11110100	0x2E	0x19	0xF4
0912–0935	00110100	01101111	01010000	0x34	0x6F	0x50
0936–0959	01010000	11101001	11000100	0x50	0xE9	0xC4
0960–0983	00000110	11011001	11101110	0x06	0xD9	0xEE
0984–1007	11111000	00011011	11011001	0xF8	0x1B	0xD9
1008–1031	01101100	10000110	11010011	0x6C	0x86	0xD3
1032–1055	11101001	01100100	11001000	0xE9	0x64	0xC8
1056–1079	11110001	10100001	00001011	0xF1	0xA1	0x0B
1080–1103	11000010	01000100	01010100	0xC2	0x44	0x54
1104–1127	10100000	10001100	10111011	0xA0	0x8C	0xBB
1128–1151	10100011	11100100	10101001	0xA3	0xE4	0xA9
1152–1175	10101011	01010000	11100010	0xAB	0x50	0xE2
1176–1199	01110000	00101000	00110110	0x70	0x28	0x36
1200–1223	11111100	00110000	00110100	0xFC	0x30	0x34
1224–1247	01101010	01001001	00100010	0x6A	0x49	0x22
1248–1271	11010101	00000111	11001111	0xD5	0x07	0xCF
1272–1295	00110101	00111010	10001110	0x35	0x3A	0x8E
1296–1319	10101001	11010001	01010101	0xA9	0xD1	0x55
1320–1343	00010010	00000100	01110100	0x12	0x04	0x74
1344–1367	11011001	11101001	00111011	0xD9	0xE9	0x3B
1368–1391	11001101	10010011	10001101	0xCD	0x93	0x8D
1392–1415	01111011	01111100	01110000	0x7B	0x7C	0x70
1416–1439	00000010	00100000	10011001	0x02	0x20	0x99
1440–1463	10100001	01111101	10001010	0xA1	0x7D	0x8A
1464–1487	00100111	00010111	00111001	0x27	0x17	0x39
1488–1511	00010101	10100000	11101100	0x15	0xA0	0xEC
1512–1535	01010101	10001010	00111111	0x55	0x8A	0x3F
1536–1559	01101011	10110110	11011000	0x6B	0xB6	0xD8
1560–1583	10110001	10001000	10000100	0xB1	0x88	0x84

Table L-41—DATA bits for LDPC example 2 after LDPC encoding *(continued)*

Bit ##	Binary value b0 b7	Binary value b8 b15	Binary value b16 b23	Hex value	Hex value	Hex value
1584–1607	00001111	00101100	10001000	0x0F	0x2C	0x88
1608–1631	10101000	11111000	10010010	0xA8	0xF8	0x92
1632–1655	10100000	10110111	10011110	0xA0	0xB7	0x9E
1656–1679	00111100	01100100	01010101	0x3C	0x64	0x55
1680–1703	00001110	01111000	11111011	0x0E	0x78	0xFB
1704–1727	01110011	01010100	00000000	0x73	0x54	0x00
1728–1751	01000010	10101011	10000010	0x42	0xAB	0x82
1752–1775	10111111	11100111	11001011	0xBF	0xE7	0xCB
1776–1799	00100110	11110011	01000000	0x26	0xF3	0x40
1800–1823	00001101	00000111	01101010	0x0D	0x07	0x6A
1824–1847	00010101	00010111	11111111	0x15	0x17	0xFF
1848–1871	10100101	11011100	00000000	0xA5	0xDC	0x00
1872–1895	00000000	00000000	00000000	0x00	0x00	0x00
1896–1919	00000000	00000000	00000000	0x00	0x00	0x00
1920–1943	00000000	00000000	00000000	0x00	0x00	0x00
1944–1967	01100100	10110110	01010100	0x64	0xB6	0x54
1968–1991	00110001	00000001	01100001	0x31	0x01	0x61
1992–2015	00101001	00010011	01110000	0x29	0x13	0x70
2016–2039	01010000	10000000	11001110	0x50	0x80	0xCE
2040–2063	01000101	11000000	10101000	0x45	0xC0	0xA8
2064–2087	11001101	11111000	01111100	0xCD	0xF8	0x7C
2088–2111	01010011	01010001	01001110	0x53	0x51	0x4E
2112–2135	11010011	10101110	00010011	0xD3	0xAE	0x13
2136–2159	11110000	11101101	10111111	0xF0	0xED	0xBF
2160–2183	10001110	10010100	00110100	0x8E	0x94	0x34
2184–2207	11111011	00010000	11011001	0xFB	0x10	0xD9
2208–2231	10111110	00110001	10011111	0xBE	0x31	0x9F
2232–2255	01100000	00011100	10100110	0x60	0x1C	0xA6
2256–2279	01010101	11111001	10100110	0x55	0xF9	0xA6
2280–2303	10101010	00111000	01110001	0xAA	0x38	0x71
2304–2327	01111010	10101100	10110010	0x7A	0xAC	0xB2
2328–2351	11110101	11010001	10000001	0xF5	0xD1	0x81
2352–2375	01010000	11110001	00001011	0x50	0xF1	0x0B

Table L-41—DATA bits for LDPC example 2 after LDPC encoding *(continued)*

Bit ##	Binary value b0 b7	Binary value b8 b15	Binary value b16 b23	Hex value	Hex value	Hex value
2376–2399	10111101	10010011	10001011	0xBD	0x93	0x8B
2400–2423	10100010	10010110	00100101	0xA2	0x96	0x25
2424–2447	11100011	01101100	11000111	0xE3	0x6C	0xC7
2448–2471	00000101	00011000	00101000	0x05	0x18	0x28
2472–2495	11110011	00111001	11011000	0xF3	0x39	0xD8
2496–2519	00010001	01110101	00010111	0x11	0x75	0x17
2520–2543	11011101	11111011	11010010	0xDD	0xFB	0xD2
2544–2567	10101010	11101011	10100110	0xAA	0xEB	0xA6
2568–2591	10000101	10110011	01011000	0x85	0xB3	0x58

L.3.6 Removing shortening bits and repetition for LDPC example 2

The shortening bits, applied before LDPC encoding, are now removed as prescribed in paragraph (c) of 20.3.11.7.5. Finally, either puncturing is applied as described in paragraph (d) of the same subclause, or the copying of repeated bits is applied as described in paragraph (e) of the same subclause. In LDPC example 2, because N_{punc} = 0 and N_{rep} = 160 are nonzero, repetition is prescribed and completes the LDPC encoding process.

The results are given in Table L-42.

NOTE—The first 80 shortening bits (bits 568–647 from Table L-41) have been removed from the first codeword between bits 567 and 568 of Table L-42, and the second 80 shortening bits (bits 1864–1943 of Table L-41) have been removed between bits 1215 and 1216 of Table L-42. Also, 80 bits have been repeated from the beginning of the first codeword (bits 0–79) to the end of the first codeword (bits 1216–1295), and 80 bits have been repeated from the beginning of the second codeword (bits 1296–1375) to end of the second codeword (bits 2512–2591) in Table L-42.

Table L-42—DATA bits after removal of shortening bits and copying of repetition bits

Bit ##	Binary value b0 b7	Binary value b8 b15	Binary value b16 b23	Hex value	Hex value	Hex value
0000–0023	01101100	00011001	10001001	0x6C	0x19	0x89
0024–0047	10001111	01101000	00111001	0x8F	0x68	0x39
0048–0071	11110100	10100101	01100001	0xF4	0xA5	0x61
0072–0095	01001111	11010111	10101110	0x4F	0xD7	0xAE
0096–0119	00100100	00001100	11110011	0x24	0x0C	0xF3
0120–0143	00111010	11100100	10111100	0x3A	0xE4	0xBC
0144–0167	01010011	10011000	11000000	0x53	0x98	0xC0
0168–0191	00011110	00110101	10110011	0x1E	0x35	0xB3

Table L-42—DATA bits after removal of shortening bits and copying of repetition bits *(continued)*

Bit ##	Binary value b0 b7	Binary value b8 b15	Binary value b16 b23	Hex value	Hex value	Hex value
0192–0215	11100011	11111000	00100101	0xE3	0xF8	0x25
0216–0239	01100000	11010110	00100101	0x60	0xD6	0x25
0240–0263	00110101	00110011	11111110	0x35	0x33	0xFE
0264–0287	11110000	01000001	00101011	0xF0	0x41	0x2B
0288–0311	10001111	01010011	00011100	0x8F	0x53	0x1C
0312–0335	10000011	01000001	10111110	0x83	0x41	0xBE
0336–0359	00111001	00101000	01100110	0x39	0x28	0x66
0360–0383	01000100	01100110	11001101	0x44	0x66	0xCD
0384–0407	11110110	10100011	11011000	0xF6	0xA3	0xD8
0408–0431	00001101	11010100	10000001	0x0D	0xD4	0x81
0432–0455	00111011	00101111	11011111	0x3B	0x2F	0xDF
0456–0479	11000011	01011000	11110111	0xC3	0x58	0xF7
0480–0503	11000110	01010010	11101011	0xC6	0x52	0xEB
0504–0527	01110000	10001111	10011110	0x70	0x8F	0x9E
0528–0551	01101010	10010000	10000001	0x6A	0x90	0x81
0552–0575	11111101	01111100	00001001	0xFD	0x7C	0x09
0576–0599	11000001	11111011	01101000	0xC1	0xFB	0x68
0600–0623	11001101	00000101	10110110	0xCD	0x05	0xB6
0624–0647	11000111	01100101	10100101	0xC7	0x65	0xA5
0648–0671	10011001	11100000	01110011	0x99	0xE0	0x73
0672–0695	01110000	01101101	01011110	0x70	0x6D	0x5E
0696–0719	01111001	11100011	01100111	0x79	0xE3	0x67
0720–0743	00100111	01011110	10010101	0x27	0x5E	0x95
0744–0767	10101000	11110110	00110101	0xA8	0xF6	0x35
0768–0791	01001000	10100111	00100110	0x48	0xA7	0x26
0792–0815	00101001	00110001	00101110	0x29	0x31	0x2E
0816–0839	00011001	11110100	00110100	0x19	0xF4	0x34
0840–0863	01101111	01010000	01010000	0x6F	0x50	0x50
0864–0887	11101001	11000100	00000110	0xE9	0xC4	0x06
0888–0911	11011001	11101110	11111000	0xD9	0xEE	0xF8
0912–0935	00011011	11011001	01101100	0x1B	0xD9	0x6C
0936–0959	10000110	11010011	11101001	0x86	0xD3	0xE9

Table L-42—DATA bits after removal of shortening bits and copying of repetition bits *(continued)*

Bit ##	Binary value b0 b7	Binary value b8 b15	Binary value b16 b23	Hex value	Hex value	Hex value
0960–0983	01100100	11001000	11110001	0x64	0xC8	0xF1
0984–1007	10100001	00001011	11000010	0xA1	0x0B	0xC2
1008–1031	01000100	01010100	10100000	0x44	0x54	0xA0
1032–1055	10001100	10111011	10100011	0x8C	0xBB	0xA3
1056–1079	11100100	10101001	10101011	0xE4	0xA9	0xAB
1080–1103	01010000	11100010	01110000	0x50	0xE2	0x70
1104–1127	00101000	00110110	11111100	0x28	0x36	0xFC
1128–1151	00110000	00110100	01101010	0x30	0x34	0x6A
1152–1175	01001001	00100010	11010101	0x49	0x22	0xD5
1176–1199	00000111	11001111	00110101	0x07	0xCF	0x35
1200–1223	00111010	10001110	01101100	0x3A	0x8E	0x6C
1224–1247	00011001	10001001	10001111	0x19	0x89	0x8F
1248–1271	01101000	00111001	11110100	0x68	0x39	0xF4
1272–1295	10100101	01100001	01001111	0xA5	0x61	0x4F
1296–1319	10101001	11010001	01010101	0xA9	0xD1	0x55
1320–1343	00010010	00000100	01110100	0x12	0x04	0x74
1344–1367	11011001	11101001	00111011	0xD9	0xE9	0x3B
1368–1391	11001101	10010011	10001101	0xCD	0x93	0x8D
1392–1415	01111011	01111100	01110000	0x7B	0x7C	0x70
1416–1439	00000010	00100000	10011001	0x02	0x20	0x99
1440–1463	10100001	01111101	10001010	0xA1	0x7D	0x8A
1464–1487	00100111	00010111	00111001	0x27	0x17	0x39
1488–1511	00010101	10100000	11101100	0x15	0xA0	0xEC
1512–1535	01010101	10001010	00111111	0x55	0x8A	0x3F
1536–1559	01101011	10110110	11011000	0x6B	0xB6	0xD8
1560–1583	10110001	10001000	10000100	0xB1	0x88	0x84
1584–1607	00001111	00101100	10001000	0x0F	0x2C	0x88
1608–1631	10101000	11111000	10010010	0xA8	0xF8	0x92
1632–1655	10100000	10110111	10011110	0xA0	0xB7	0x9E
1656–1679	00111100	01100100	01010101	0x3C	0x64	0x55
1680–1703	00001110	01111000	11111011	0x0E	0x78	0xFB
1704–1727	01110011	01010100	00000000	0x73	0x54	0x00

Table L-42—DATA bits after removal of shortening bits and copying of repetition bits *(continued)*

Bit ##	Binary value b0 b7	Binary value b8 b15	Binary value b16 b23	Hex value	Hex value	Hex value
1728–1751	01000010	10101011	10000010	0x42	0xAB	0x82
1752–1775	10111111	11100111	11001011	0xBF	0xE7	0xCB
1776–1799	00100110	11110011	01000000	0x26	0xF3	0x40
1800–1823	00001101	00000111	01101010	0x0D	0x07	0x6A
1824–1847	00010101	00010111	11111111	0x15	0x17	0xFF
1848–1871	10100101	11011100	01100100	0xA5	0xDC	0x64
1872–1895	10110110	01010100	00110001	0xB6	0x54	0x31
1896–1919	00000001	01100001	00101001	0x01	0x61	0x29
1920–1943	00010011	01110000	01010000	0x13	0x70	0x50
1944–1967	10000000	11001110	01000101	0x80	0xCE	0x45
1968–1991	11000000	10101000	11001101	0xC0	0xA8	0xCD
1992–2015	11111000	01111100	01010011	0xF8	0x7C	0x53
2016–2039	01010001	01001110	11010011	0x51	0x4E	0xD3
2040–2063	10101110	00010011	11110000	0xAE	0x13	0xF0
2064–2087	11101101	10111111	10001110	0xED	0xBF	0x8E
2088–2111	10010100	00110100	11111011	0x94	0x34	0xFB
2112–2135	00010000	11011001	10111110	0x10	0xD9	0xBE
2136–2159	00110001	10011111	01100000	0x31	0x9F	0x60
2160–2183	00011100	10100110	01010101	0x1C	0xA6	0x55
2184–2207	11111001	10100110	10101010	0xF9	0xA6	0xAA
2208–2231	00111000	01110001	01111010	0x38	0x71	0x7A
2232–2255	10101100	10110010	11110101	0xAC	0xB2	0xF5
2256–2279	11010001	10000001	01010000	0xD1	0x81	0x50
2280–2303	11110001	00001011	10111101	0xF1	0x0B	0xBD
2304–2327	10010011	10001011	10100010	0x93	0x8B	0xA2
2328–2351	10010110	00100101	11100011	0x96	0x25	0xE3
2352–2375	01101100	11000111	00000101	0x6C	0xC7	0x05
2376–2399	00011000	00101000	11110011	0x18	0x28	0xF3
2400–2423	00111001	11011000	00010001	0x39	0xD8	0x11
2424–2447	01110101	00010111	11011101	0x75	0x17	0xDD
2448–2471	11111011	11010010	10101010	0xFB	0xD2	0xAA
2472–2495	11101011	10100110	10000101	0xEB	0xA6	0x85

Table L-42—DATA bits after removal of shortening bits and copying of repetition bits *(continued)*

Bit ##	Binary value b0 b7	Binary value b8 b15	Binary value b16 b23	Hex value	Hex value	Hex value
2496–2519	10110011	01011000	10101001	0xB3	0x58	0xA9
2520–2543	11010001	01010101	00010010	0xD1	0x55	0x12
2544–2567	00000100	01110100	11011001	0x04	0x74	0xD9
2568–2591	11101001	00111011	11001101	0xE9	0x3B	0xCD

Annex M

(informative)

RSNA reference implementations and test vectors

M.1 TKIP temporal key mixing function reference implementation and test vector

M.1.1 TKIP temporal key mixing function reference implementation

This clause provides a C-language reference implementation of the temporal key mixing function.

```
/***********************************************************************

   Contents:    Generate IEEE 802.11 per-frame ARC4 key hash test vectors
   Date:        April 19, 2002
   Notes:
   This code is written for pedagogical purposes, NOT for performance.

************************************************************************/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <assert.h>
#include <time.h>

typedef unsigned char  byte;   /*  8-bit byte (octet)  */
typedef unsigned short u16b;   /* 16-bit unsigned word */
typedef unsigned long  u32b;   /* 32-bit unsigned word */

/* macros for extraction/creation of byte/u16b values  */
#define RotR1(v16)   ((((v16) >> 1) & 0x7FFF) ^ (((v16) & 1) << 15))
#define   Lo8(v16)   ((byte)( (v16)       & 0x00FF))
#define   Hi8(v16)   ((byte)(((v16) >> 8) & 0x00FF))
#define  Lo16(v32)   ((u16b)( (v32)       & 0xFFFF))
#define  Hi16(v32)   ((u16b)(((v32) >>16) & 0xFFFF))
#define  Mk16(hi,lo) ((lo) ^ (((u16b)(hi)) << 8))

/* select the Nth 16-bit word of the Temporal Key byte array TK[]   */
#define  TK16(N)     Mk16(TK[2*(N)+1],TK[2*(N)])

/* S-box lookup: 16 bits --> 16 bits */
#define _S_(v16)     (Sbox[0][Lo8(v16)] ^ Sbox[1][Hi8(v16)])
```

```
/* fixed algorithm "parameters" */
#define PHASE1_LOOP_CNT   8    /* this needs to be "big enough"     */
#define TA_SIZE           6    /*  48-bit transmitter address       */
#define TK_SIZE          16    /* 128-bit Temporal Key              */
#define P1K_SIZE         10    /*  80-bit Phase1 key                */
#define ARC4_KEY_SIZE    16    /* 128-bit ARC4KEY (104 bits unknown)*/

/* configuration settings */
#define DO_SANITY_CHECK   1    /* validate properties of S-box?     */

/* 2-byte by 2-byte subset of the full AES S-box table */
const u16b Sbox[2][256]=       /* Sbox for hash (can be in ROM)     */
{ {
   0xC6A5,0xF884,0xEE99,0xF68D,0xFF0D,0xD6BD,0xDEB1,0x9154,
   0x6050,0x0203,0xCEA9,0x567D,0xE719,0xB562,0x4DE6,0xEC9A,
   0x8F45,0x1F9D,0x8940,0xFA87,0xEF15,0xB2EB,0x8EC9,0xFB0B,
   0x41EC,0xB367,0x5FFD,0x45EA,0x23BF,0x53F7,0xE496,0x9B5B,
   0x75C2,0xE11C,0x3DAE,0x4C6A,0x6C5A,0x7E41,0xF502,0x834F,
   0x685C,0x51F4,0xD134,0xF908,0xE293,0xAB73,0x6253,0x2A3F,
   0x080C,0x9552,0x4665,0x9D5E,0x3028,0x37A1,0x0A0F,0x2FB5,
   0x0E09,0x2436,0x1B9B,0xDF3D,0xCD26,0x4E69,0x7FCD,0xEA9F,
   0x121B,0x1D9E,0x5874,0x342E,0x362D,0xDCB2,0xB4EE,0x5BFB,
   0xA4F6,0x764D,0xB761,0x7DCE,0x527B,0xDD3E,0x5E71,0x1397,
   0xA6F5,0xB968,0x0000,0xC12C,0x4060,0xE31F,0x79C8,0xB6ED,
   0xD4BE,0x8D46,0x67D9,0x724B,0x94DE,0x98D4,0xB0E8,0x854A,
   0xBB6B,0xC52A,0x4FE5,0xED16,0x86C5,0x9AD7,0x6655,0x1194,
   0x8ACF,0xE910,0x0406,0xFE81,0xA0F0,0x7844,0x25BA,0x4BE3,
   0xA2F3,0x5DFE,0x80C0,0x058A,0x3FAD,0x21BC,0x7048,0xF104,
   0x63DF,0x77C1,0xAF75,0x4263,0x2030,0xE51A,0xFD0E,0xBF6D,
   0x814C,0x1814,0x2635,0xC32F,0xBEE1,0x35A2,0x88CC,0x2E39,
   0x9357,0x55F2,0xFC82,0x7A47,0xC8AC,0xBAE7,0x322B,0xE695,
   0xC0A0,0x1998,0x9ED1,0xA37F,0x4466,0x547E,0x3BAB,0x0B83,
   0x8CCA,0xC729,0x6BD3,0x283C,0xA779,0xBCE2,0x161D,0xAD76,
   0xDB3B,0x6456,0x744E,0x141E,0x92DB,0x0C0A,0x486C,0xB8E4,
   0x9F5D,0xBD6E,0x43EF,0xC4A6,0x39A8,0x31A4,0xD337,0xF28B,
   0xD532,0x8B43,0x6E59,0xDAB7,0x018C,0xB164,0x9CD2,0x49E0,
   0xD8B4,0xACFA,0xF307,0xCF25,0xCAAF,0xF48E,0x47E9,0x1018,
   0x6FD5,0xF088,0x4A6F,0x5C72,0x3824,0x57F1,0x73C7,0x9751,
   0xCB23,0xA17C,0xE89C,0x3E21,0x96DD,0x61DC,0x0D86,0x0F85,
   0xE090,0x7C42,0x71C4,0xCCAA,0x90D8,0x0605,0xF701,0x1C12,
   0xC2A3,0x6A5F,0xAEF9,0x69D0,0x1791,0x9958,0x3A27,0x27B9,
   0xD938,0xEB13,0x2BB3,0x2233,0xD2BB,0xA970,0x0789,0x33A7,
   0x2DB6,0x3C22,0x1592,0xC920,0x8749,0xAAFF,0x5078,0xA57A,
   0x038F,0x59F8,0x0980,0x1A17,0x65DA,0xD731,0x84C6,0xD0B8,
   0x82C3,0x29B0,0x5A77,0x1E11,0x7BCB,0xA8FC,0x6DD6,0x2C3A,
  },
  {  /* second half of table is byte-reversed version of first! */
   0xA5C6,0x84F8,0x99EE,0x8DF6,0x0DFF,0xBDD6,0xB1DE,0x5491,
   0x5060,0x0302,0xA9CE,0x7D56,0x19E7,0x62B5,0xE64D,0x9AEC,
   0x458F,0x9D1F,0x4089,0x87FA,0x15EF,0xEBB2,0xC98E,0x0BFB,
   0xEC41,0x67B3,0xFD5F,0xEA45,0xBF23,0xF753,0x96E4,0x5B9B,
   0xC275,0x1CE1,0xAE3D,0x6A4C,0x5A6C,0x417E,0x02F5,0x4F83,
   0x5C68,0xF451,0x34D1,0x08F9,0x93E2,0x73AB,0x5362,0x3F2A,
   0x0C08,0x5295,0x6546,0x5E9D,0x2830,0xA137,0x0F0A,0xB52F,
   0x090E,0x3624,0x9B1B,0x3DDF,0x26CD,0x694E,0xCD7F,0x9FEA,
   0x1B12,0x9E1D,0x7458,0x2E34,0x2D36,0xB2DC,0xEEB4,0xFB5B,
   0xF6A4,0x4D76,0x61B7,0xCE7D,0x7B52,0x3EDD,0x715E,0x9713,
   0xF5A6,0x68B9,0x0000,0x2CC1,0x6040,0x1FE3,0xC879,0xEDB6,
   0xBED4,0x468D,0xD967,0x4B72,0xDE94,0xD498,0xE8B0,0x4A85,
   0x6BBB,0x2AC5,0xE54F,0x16ED,0xC586,0xD79A,0x5566,0x9411,
   0xCF8A,0x10E9,0x0604,0x81FE,0xF0A0,0x4478,0xBA25,0xE34B,
   0xF3A2,0xFE5D,0xC080,0x8A05,0xAD3F,0xBC21,0x4870,0x04F1,
   0xDF63,0xC177,0x75AF,0x6342,0x3020,0x1AE5,0x0EFD,0x6DBF,
   0x4C81,0x1418,0x3526,0x2FC3,0xE1BE,0xA235,0xCC88,0x392E,
   0x5793,0xF255,0x82FC,0x477A,0xACC8,0xE7BA,0x2B32,0x95E6,
   0xA0C0,0x9819,0xD19E,0x7FA3,0x6644,0x7E54,0xAB3B,0x830B,
   0xCA8C,0x29C7,0xD36B,0x3C28,0x79A7,0xE2BC,0x1D16,0x76AD,
   0x3BDB,0x5664,0x4E74,0x1E14,0xDB92,0x0A0C,0x6C48,0xE4B8,
   0x5D9F,0x6EBD,0xEF43,0xA6C4,0xA839,0xA431,0x37D3,0x8BF2,
   0x32D5,0x438B,0x596E,0xB7DA,0x8C01,0x64B1,0xD29C,0xE049,
   0xB4D8,0xFAAC,0x07F3,0x25CF,0xAFCA,0x8EF4,0xE947,0x1810,
   0xD56F,0x88F0,0x6F4A,0x725C,0x2438,0xF157,0xC773,0x5197,
   0x23CB,0x7CA1,0x9CE8,0x213E,0xDD96,0xDC61,0x860D,0x850F,
   0x90E0,0x427C,0xC471,0xAACC,0xD890,0x0506,0x01F7,0x121C,
   0xA3C2,0x5F6A,0xF9AE,0xD069,0x9117,0x5899,0x273A,0xB927,
   0x38D9,0x13EB,0xB32B,0x3322,0xBBD2,0x70A9,0x8907,0xA733,
   0xB62D,0x223C,0x9215,0x20C9,0x4987,0xFFAA,0x7850,0x7AA5,
   0x8F03,0xF859,0x8009,0x171A,0xDA65,0x31D7,0xC684,0xB8D0,
   0xC382,0xB029,0x775A,0x111E,0xCB7B,0xFCA8,0xD66D,0x3A2C,
  }
};

#if DO_SANITY_CHECK
/*
**********************************************************************
* Routine: SanityCheckTable -- verify Sbox properties
*
* Inputs:  Sbox
* Output:  None, but an assertion fails if the tables are wrong
* Notes:
*   The purpose of this routine is solely to illustrate and
*   verify the following properties of the Sbox table:
*      - the Sbox is a "2x2" subset of the AES table:
*            Sbox + affine transform + MDS.
*      - the Sbox table can be easily designed to fit in a
*            512-byte table, using a byte swap
*      - the Sbox table can be easily designed to fit in a
*            256-byte table, using some shifts and a byte swap
**********************************************************************
*/
void SanityCheckTable(void)
    {
    const static int  M_x = 0x11B;   /* AES irreducible polynomial */
    const static byte Sbox8[256] = { /* AES 8-bit Sbox */
        0x63,0x7c,0x77,0x7b,0xf2,0x6b,0x6f,0xc5,
        0x30,0x01,0x67,0x2b,0xfe,0xd7,0xab,0x76,
        0xca,0x82,0xc9,0x7d,0xfa,0x59,0x47,0xf0,
        0xad,0xd4,0xa2,0xaf,0x9c,0xa4,0x72,0xc0,
        0xb7,0xfd,0x93,0x26,0x36,0x3f,0xf7,0xcc,
        0x34,0xa5,0xe5,0xf1,0x71,0xd8,0x31,0x15,
        0x04,0xc7,0x23,0xc3,0x18,0x96,0x05,0x9a,
        0x07,0x12,0x80,0xe2,0xeb,0x27,0xb2,0x75,
        0x09,0x83,0x2c,0x1a,0x1b,0x6e,0x5a,0xa0,
        0x52,0x3b,0xd6,0xb3,0x29,0xe3,0x2f,0x84,
        0x53,0xd1,0x00,0xed,0x20,0xfc,0xb1,0x5b,
        0x6a,0xcb,0xbe,0x39,0x4a,0x4c,0x58,0xcf,
        0xd0,0xef,0xaa,0xfb,0x43,0x4d,0x33,0x85,
        0x45,0xf9,0x02,0x7f,0x50,0x3c,0x9f,0xa8,
        0x51,0xa3,0x40,0x8f,0x92,0x9d,0x38,0xf5,
        0xbc,0xb6,0xda,0x21,0x10,0xff,0xf3,0xd2,
        0xcd,0x0c,0x13,0xec,0x5f,0x97,0x44,0x17,
        0xc4,0xa7,0x7e,0x3d,0x64,0x5d,0x19,0x73,
        0x60,0x81,0x4f,0xdc,0x22,0x2a,0x90,0x88,
        0x46,0xee,0xb8,0x14,0xde,0x5e,0x0b,0xdb,
        0xe0,0x32,0x3a,0x0a,0x49,0x06,0x24,0x5c,
        0xc2,0xd3,0xac,0x62,0x91,0x95,0xe4,0x79,
        0xe7,0xc8,0x37,0x6d,0x8d,0xd5,0x4e,0xa9,
        0x6c,0x56,0xf4,0xea,0x65,0x7a,0xae,0x08,
        0xba,0x78,0x25,0x2e,0x1c,0xa6,0xb4,0xc6,
        0xe8,0xdd,0x74,0x1f,0x4b,0xbd,0x8b,0x8a,
        0x70,0x3e,0xb5,0x66,0x48,0x03,0xf6,0x0e,
        0x61,0x35,0x57,0xb9,0x86,0xc1,0x1d,0x9e,
        0xe1,0xf8,0x98,0x11,0x69,0xd9,0x8e,0x94,
        0x9b,0x1e,0x87,0xe9,0xce,0x55,0x28,0xdf,
        0x8c,0xa1,0x89,0x0d,0xbf,0xe6,0x42,0x68,
        0x41,0x99,0x2d,0x0f,0xb0,0x54,0xbb,0x16 };
    int i,k,k2,k3;
    byte bitmap[0x2000];

    /* show that smaller tables can be used, if desired */
    for (i=0;i<256;i++)
        {
        k  = Sbox8[i];
        k2 = (k << 1) ^ ((k & 0x80) ? M_x : 0);
        k3 =  k ^ k2;
        assert(Sbox[0][i] == ((k2 << 8) ^ k3));
        assert(Sbox[1][i] == ((k3 << 8) ^ k2));
        }

    /* now make sure that it's a 16-bit permutation */
    memset(bitmap,0,sizeof(bitmap));
    for (i=0;i<0x10000;i++)
        {
        k = _S_(i); /* do an S-box lookup: 16 --> 16 bits */
        assert(k < (1 << 16));
        assert((bitmap[k >> 3] & (1 << (k & 7))) == 0);
        bitmap[k >> 3] |= 1 << (k & 7);
        }
    for (i=0;i<sizeof(bitmap);i++)
        assert(bitmap[i] == 0xFF);

    /* if we reach here, the 16-bit Sbox is ok */
    printf("Table sanity check successful\n");
    }
#endif

/*
**********************************************************************
* Routine: Phase 1 -- generate P1K, given TA, TK, IV32
*
* Inputs:
*     TK[]      = Temporal Key                         [128 bits]
*     TA[]      = transmitter's MAC address            [ 48 bits]
*     IV32      = upper 32 bits of IV                  [ 32 bits]
* Output:
*     P1K[]     = Phase 1 key                          [ 80 bits]
*
* Note:
*     This function only needs to be called every 2**16 frames,
*     although in theory it could be called every frame.
*
**********************************************************************
```

```
*/
void Phase1(u16b *P1K,const byte *TK,const byte *TA,u32b IV32)
    {
    int  i;

    /* Initialize the 80 bits of P1K[] from IV32 and TA[0..5]     */
    P1K[0]      = Lo16(IV32);
    P1K[1]      = Hi16(IV32);
    P1K[2]      = Mk16(TA[1],TA[0]); /* use TA[] as little-endian */
    P1K[3]      = Mk16(TA[3],TA[2]);
    P1K[4]      = Mk16(TA[5],TA[4]);

    /* Now compute an unbalanced Feistel cipher with 80-bit block */
    /* size on the 80-bit block P1K[], using the 128-bit key TK[] */
    for (i=0; i < PHASE1_LOOP_CNT ;i++)
        {                 /* Each add operation here is mod 2**16 */
        P1K[0] += _S_(P1K[4] ^ TK16((i&1)+0));
        P1K[1] += _S_(P1K[0] ^ TK16((i&1)+2));
        P1K[2] += _S_(P1K[1] ^ TK16((i&1)+4));
        P1K[3] += _S_(P1K[2] ^ TK16((i&1)+6));
        P1K[4] += _S_(P1K[3] ^ TK16((i&1)+0));
        P1K[4] +=  i;                    /* avoid "slide attacks" */
        }
    }

/*
**********************************************************************
* Routine: Phase 2 -- generate ARC4KEY, given TK, P1K, IV16
*
* Inputs:
*     TK[]      = Temporal Key                         [128 bits]
*     P1K[]     = Phase 1 output key                   [ 80 bits]
*     IV16      = low 16 bits of IV counter            [ 16 bits]
* Output:
*     ARC4KEY[]  = the key used to encrypt the frame   [128 bits]
*
* Note:
*     The value {TA,IV32,IV16} for Phase1/Phase2 must be unique
*     across all frames using the same key TK value. Then, for a
*     given value of TK[], this TKIP48 construction guarantees that
*     the final ARC4KEY value is unique across all frames.
*
* Suggested implementation optimization: if PPK[] is "overlaid"
*     appropriately on ARC4KEY[], there is no need for the final
*     for loop below that copies the PPK[] result into ARC4KEY[].
*
```

```
**********************************************************************
*/
void Phase2(byte *ARC4KEY,const byte *TK,const u16b *P1K,u16b IV16)
    {
    int  i;
    u16b PPK[6];                        /* temporary key for mixing    */

    /* all adds in the PPK[] equations below are mod 2**16          */
    for (i=0;i<5;i++) PPK[i]=P1K[i];    /* first, copy P1K to PPK      */
    PPK[5]  =  P1K[4] + IV16;           /* next, add in IV16           */

    /* Bijective non-linear mixing of the 96 bits of PPK[0..5]         */
    PPK[0] +=    _S_(PPK[5] ^ TK16(0)); /* Mix key in each "round"     */
    PPK[1] +=    _S_(PPK[0] ^ TK16(1));
    PPK[2] +=    _S_(PPK[1] ^ TK16(2));
    PPK[3] +=    _S_(PPK[2] ^ TK16(3));
    PPK[4] +=    _S_(PPK[3] ^ TK16(4));
    PPK[5] +=    _S_(PPK[4] ^ TK16(5)); /* Total # S-box lookups == 6  */

    /* Final sweep: bijective, linear. Rotates kill LSB correlations   */
    PPK[0] +=  RotR1(PPK[5] ^ TK16(6));
    PPK[1] +=  RotR1(PPK[0] ^ TK16(7)); /* Use all of TK[] in Phase2   */
    PPK[2] +=  RotR1(PPK[1]);
    PPK[3] +=  RotR1(PPK[2]);
    PPK[4] +=  RotR1(PPK[3]);
    PPK[5] +=  RotR1(PPK[4]);
    /* At this point, for a given key TK[0..15], the 96-bit output */
    /*     value PPK[0..5] is guaranteed to be unique, as a function   */
    /*     of the 96-bit "input" value   {TA,IV32,IV16}. That is, P1K  */
    /*     is now a keyed permutation of {TA,IV32,IV16}.               */

    /* Set ARC4KEY[0..3], which includes cleartext portion of ARC4 key  */
    ARC4KEY[0] = Hi8(IV16);             /* ARC4KEY[0..2] is the WEP IV  */
    ARC4KEY[1] =(Hi8(IV16) | 0x20) & 0x7F; /* Help avoid FMS weak keys  */
    ARC4KEY[2] = Lo8(IV16);
    ARC4KEY[3] = Lo8((PPK[5] ^ TK16(0)) >> 1);

    /* Copy 96 bits of PPK[0..5] to ARC4KEY[4..15]  (little-endian)      */
    for (i=0;i<6;i++)
        {
        ARC4KEY[4+2*i] = Lo8(PPK[i]);
        ARC4KEY[5+2*i] = Hi8(PPK[i]);
        }
    }

/*
```

```
*********************************************************************
* Routine: doTestCase -- execute a test case, and print results
*********************************************************************
*/
void DoTestCase(byte *ARC4KEY,u32b IV32,u16b IV16,const byte *TA,const
byte *TK)
    {
    int  i;
    u16b P1K[P1K_SIZE/2];  /* "temp" copy of phase1 key */

    printf("\nTK    =");
    for (i=0;i<TK_SIZE;i++) printf(" %02X",TK[i]);
    printf("\nTA    =");
    for (i=0;i<TA_SIZE;i++) printf(" %02X",TA[i]);
    printf("\nIV32  = %08X   [transmitted as",IV32);  /* show byte order */
    for (i=0;i<4;i++) printf(" %02X",(IV32 >> (24-8*i)) & 0xFF);
    printf("]");
    printf("\nIV16  = %04X",IV16);

    Phase1(P1K,TK,TA,IV32);

    printf("\nP1K   =");
    for (i=0;i<P1K_SIZE/2;i++) printf(" %04X ",P1K[i] & 0xFFFF);

    Phase2(ARC4KEY,TK,P1K,IV16);

    printf("\nARC4KEY= ");
    for (i=0;i<ARC4_KEY_SIZE;i++) printf("%02X ",ARC4KEY[i]);
    }

/*
*********************************************************************
* Static (Repeatable) Test Cases
*********************************************************************
*/
void DoStaticTestCases(int testCnt)
    {
    int  i,j;
    byte TA[TA_SIZE],TK[TK_SIZE],ARC4KEY[ARC4_KEY_SIZE];
    u16b IV16=0;
    u32b IV32=0;

    /* set a fixed starting point */
    for (i=0;i<TK_SIZE;i++) TK[i]=i;
    for (i=0;i<TA_SIZE;i++) TA[i]=(i+1)*17;
    TA[0] = TA[0] & 0xFC;            /* set to 0 I/G and U/L bits in OUI */
    /* now generate tests, feeding results back into new tests */
    for (i=0; i<testCnt/2; i++)
        {
        printf("\n\nTest vector #%d:",2*i+1);
        DoTestCase(ARC4KEY,IV32,IV16,TA,TK);
        IV16++;                      /* emulate per-frame "increment" */
        if (IV16 == 0) IV32++;
        printf("\n\nTest vector #%d:",2*i+2);
        DoTestCase(ARC4KEY,IV32,IV16,TA,TK);
        /* feed results back to seed the next test input values    */
        IV16 = (i) ? Mk16(ARC4KEY[15],ARC4KEY[4]) : 0xFFFF;/* force wrap */
        IV32 =       Mk16(ARC4KEY[14],ARC4KEY[5]);
        IV32 =       Mk16(ARC4KEY[13],ARC4KEY[7]) + (IV32 << 16);
        for (j=0;j<TA_SIZE;j++) TA[j]^=ARC4KEY[12-j];
        for (j=0;j<TK_SIZE;j++) TK[j]^=ARC4KEY[(j+i+1) % ARC4_KEY_SIZE] ^
                                       ARC4KEY[(j+i+7) % ARC4_KEY_SIZE] ;
        TA[0] = TA[0] & 0xFC;        /* set to 0 I/G and U/L bits in OUI */
        }
    /* comparing the final output is a good check of correctness   */
    printf("\n");
    }

/*
**********************************************************************
* Test Cases Generated at Random
**********************************************************************
*/
void DoRandomTestCases(int testCnt)
    {
    int  i,j;
    u16b IV16;
    u32b IV32;
    byte TA[TA_SIZE],ARC4KEY[ARC4_KEY_SIZE],TK[TK_SIZE];

    printf("Random tests:\n");
    /* now generate tests "recursively" */
    for (i=0; i<testCnt; i++)
        {
        IV16 = rand() & 0xFFFF;
        IV32 = rand() + (rand() << 16);
        for (j=0;j<TK_SIZE;j++) TK[j]=rand() & 0xFF;
        for (j=0;j<TA_SIZE;j++) TA[j]=rand() & 0xFF;
        TA[0] = TA[0] & 0xFC;        /* set to 0 I/G and U/L bits in OUI */
        printf("\n\nRandom test vector #%d:",i+1);
        DoTestCase(ARC4KEY,IV32,IV16,TA,TK);
```

```
            }
        printf("\n");
        }

/*
**********************************************************************
* Usage text
**********************************************************************
*/
#define NUM_TEST_CNT  8
void Usage(void)
    {
    printf(
        "Usage:   TKIP48 [options]\n"
        "Purpose: Generate test vectors for IEEE 802.11 TKIP48\n"
        "Options  -?   -- output this usage text\n"
        "         -r   -- generate test vectors at random\n"
        "         -sN  -- init random seed to N\n"
        "         -tN  -- generate N tests (default = %d)\n",
        NUM_TEST_CNT
        );
    exit(0);
    }

/*
**********************************************************************
* Main
**********************************************************************
*/
int main(int argc, char **argv)
    {
    char *parg;
    int   i,doRand = 0;
    int   testCnt  = NUM_TEST_CNT;
    u32b  seed     = (u32b) time(NULL);

#if DO_SANITY_CHECK
    SanityCheckTable();
#endif

    for (i=1; i<argc; i++)
        {
        parg = argv[i];
        switch (parg[0])
            {
            case '-':
```

```
            switch (parg[1])
                {
                case '?':
                case 'H':
                case 'h':
                    Usage();
                    return 0;
                case 'R':
                case 'r':  /* generate some random test vectors */
                    doRand  = 1;
                    break;
                case 'S':
                case 's':
                    seed    = atoi(parg+2);
                    break;
                case 'T':
                case 't':
                    testCnt = atoi(parg+2);
                    break;
                default:
                    break;
                }
            break;
        case '?':
            Usage();
            return 0;
        default:
            printf("Invalid argument: \"%s\"\n", parg);
            return 1;
        }
    }
srand(seed);
if (doRand) printf("Seed = %u\n",seed);

/* generate some test vectors */
if (doRand) DoRandomTestCases(testCnt);
else        DoStaticTestCases(testCnt);

return 0;
}
```

M.1.2 Test vectors

The following output is provided to test implementations of the temporal key hash algorithm. All input and output values are shown in hexadecimal.

```
Test vector #1:
```

```
TK     = 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F [LSB on left, MSB on right]
TA     = 10-22-33-44-55-66
PN     = 000000000000
IV32   = 00000000
IV16   = 0000
P1K    = 3DD2  016E  76F4  8697  B2E8
ARC4KEY= 00 20 00 33 EA 8D 2F 60 CA 6D 13 74 23 4A 66 0B

Test vector #2:
TK     = 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F [LSB on left, MSB on right]
TA     = 10-22-33-44-55-66
PN     = 000000000001
IV32   = 00000000
IV16   = 0001
P1K    = 3DD2  016E  76F4  8697  B2E8
ARC4KEY= 00 20 01 90 FF DC 31 43 89 A9 D9 D0 74 FD 20 AA

Test vector #3:
TK     = 63 89 3B 25 08 40 B8 AE 0B D0 FA 7E 61 D2 78 3E [LSB on left, MSB on right]
TA     = 64-F2-EA-ED-DC-25
PN     = 20DCFD43FFFF
IV32   = 20DCFD43
IV16   = FFFF
P1K    = 7C67  49D7  9724  B5E9  B4F1
ARC4KEY= FF 7F FF 93 81 0F C6 E5 8F 5D D3 26 25 15 44 CE

Test vector #4:
TK     = 63 89 3B 25 08 40 B8 AE 0B D0 FA 7E 61 D2 78 3E [LSB on left, MSB on right]
TA     = 64-F2-EA-ED-DC-25
PN     = 20DCFD440000
IV32   = 20DCFD44
IV16   = 0000
P1K    = 5A5D  73A8  A859  2EC1  DC8B
ARC4KEY= 00 20 00 49 8C A4 71 FC FB FA A1 6E 36 10 F0 05

Test vector #5:
TK     = 98 3A 16 EF 4F AC B3 51 AA 9E CC 27 1D 73 09 E2 [LSB on left, MSB on right]
TA     = 50-9C-4B-17-27-D9
PN     = F0A410FC058C
IV32   = F0A410FC
IV16   = 058C
P1K    = F2DF  EBB1  88D3  5923  A07C
ARC4KEY= 05 25 8C F4 D8 51 52 F4 D9 AF 1A 64 F1 D0 70 21

Test vector #6:
TK     = 98 3A 16 EF 4F AC B3 51 AA 9E CC 27 1D 73 09 E2 [LSB on left, MSB on right]
TA     = 50-9C-4B-17-27-D9
PN     = F0A410FC058D
IV32   = F0A410FC
IV16   = 058D
P1K    = F2DF  EBB1  88D3  5923  A07C
ARC4KEY= 05 25 8D 09 F8 15 43 B7 6A 59 6F C2 C6 73 8B 30

Test vector #7:
TK     = C8 AD C1 6A 8B 4D DA 3B 4D D5 B6 54 38 35 9B 05 [LSB on left, MSB on right]
TA     = 94-5E-24-4E-4D-6E
PN     = 8B1573B730F8
IV32   = 8B1573B7
IV16   = 30F8
P1K    = EFF1  3F38  A364  60A9  76F3
ARC4KEY= 30 30 F8 65 0D A0 73 EA 61 4E A8 F4 74 EE 03 19

Test vector #8:
TK     = C8 AD C1 6A 8B 4D DA 3B 4D D5 B6 54 38 35 9B 05 [LSB on left, MSB on right]
```

```
TA     = 94-5E-24-4E-4D-6E
PN     = 8B1573B730F9
IV32   = 8B1573B7
IV16   = 30F9
P1K    = EFF1  3F38  A364  60A9  76F3
ARC4KEY= 30 30 F9 31 55 CE 29 34 37 CC 76 71 27 16 AB 8F
```

M.2 Michael reference implementation and test vectors

M.2.1 Michael test vectors

M.2.1.1 Introduction

To ensure correct implementation of Michael, here are some test vectors. These test vectors still have to be confirmed by an independent implementation.

M.2.1.2 Block function

Table M-1 gives some test vectors for the block function.

Table M-1—Test vectors for block function

Input	# times	Output
(00000000, 00000000)	1	(00000000, 00000000)
(00000000, 00000001)	1	(c00015a8, c0000b95)
(00000001, 00000000)	1	(6b519593, 572b8b8a)
(01234567, 83659326)	1	(441492c2, 1d8427ed)
(00000001, 00000000)	1000	(9f04c4ad, 2ec6c2bf)

The first four rows give test vectors for a single application of the block function *b*. The last row gives a test vector for 1000 repeated applications of the block function. Together these should provide adequate test coverage.

M.2.1.3 Michael

Table M-2 gives some test vectors for Michael.

Table M-2—Test vectors for Michael

Key	Message	Output
0000000000000000	""	82925c1ca1d130b8
82925c1ca1d130b8	"M”	434721ca40639b3f
434721ca40639b3f	"Mi “	e8f9becae97e5d29
E8f9becae97e5d29	"Mic"	90038fc6cf13c1db

Table M-2—Test vectors for Michael

Key	Message	Output
90038fc6cf13c1db	"Mich"	D55e100510128986
D55e100510128986	"Michael"	0a942b124ecaa546

Note that each key is the result of the previous line, which makes it easy to construct a single test out of all of these test cases.

M.2.2 Sample code for Michael

```
//
// Michael.h    Reference implementation for Michael

//
// A Michael object implements the computation of the MIC.
//
// Conceptually, the object stores the message to be authenticated.
// At construction the message is empty.
// The append() method appends bytes to the message.
// The getMic() method computes the MIC over the message and returns the
// result.
// As a side-effect it also resets the stored message
// to the empty message so that the object can be reused
// for another MIC computation.

class Michael
{

public:
    // Constructor requires a pointer to 8 bytes of key
    Michael( Byte * key );

    // Destructor
    ~Michael();

    // Clear the internal message,
    // resets the object to the state just after construction.
    void clear();

    // Set the key to a new value
    void setKey( Byte * key );

    // Append bytes to the message to be MICed
    void append( Byte * src, int nBytes );

    // Get the MIC result. Destination should accept 8 bytes of result.
    // This also resets the message to empty.
    void getMIC( Byte * dst );

    // Run the test plan to verify proper operations
    static void runTestPlan();

private:
    // Copy constructor declared but not defined,
    //avoids compiler-generated version.
    Michael( const Michael & );
    // Assignment operator declared but not defined,
    //avoids compiler-generated version.
    void operator=( const Michael & );

    // A bunch of internal functions

    // Get UInt32 from 4 bytes LSByte first
    static UInt32 getUInt32( Byte * p );

    // Put UInt32 into 4 bytes LSByte first
    static void putUInt32( Byte * p, UInt32 val );

    // Add a single byte to the internal message
    void appendByte( Byte b );

    // Conversion of hex string to binary string
    static void hexToBin( char *src, Byte * dst );

    // More conversion of hex string to binary string
    static void hexToBin( char *src, int nChars, Byte * dst );

    // Helper function for hex conversion
    static Byte hexToBinNibble( char c );

    // Run a single test case
    static void runSingleTest( char * cKey, char * cMsg, char * cResult );

    UInt32  K0, K1;         // Key
    UInt32  L, R;           // Current state
    UInt32  M;              // Message accumulator (single word)
    int     nBytesInM;      // # bytes in M
};
```

```
//
// Michael.cpp  Reference implementation for Michael
//

// Adapt these typedefs to your local platform
typedef unsigned long UInt32;
typedef unsigned char Byte;

#include <assert.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#include "Michael.h"

// Rotation functions on 32 bit values
#define ROL32( A, n ) \
 ( ((A) << (n)) | ( ((A)>>(32-(n)))  & ( (1UL << (n)) - 1 ) ) )
#define ROR32( A, n ) ROL32( (A), 32-(n) )

UInt32 Michael::getUInt32( Byte * p )
// Convert from Byte[] to UInt32 in a portable way
{
    UInt32 res = 0;
    for( int i=0; i<4; i++ )
    {
        res |= (*p++) << (8*i);
    }
    return res;
}

void Michael::putUInt32( Byte * p, UInt32 val )
// Convert from UInt32 to Byte[] in a portable way
{
    for( int i=0; i<4; i++ )
    {
        *p++ = (Byte) (val & 0xff);
        val >>= 8;
    }
}

void Michael::clear()
{
    // Reset the state to the empty message.
    L = K0;
    R = K1;
```

```
        nBytesInM = 0;
        M = 0;
    }

    void Michael::setKey( Byte * key )
    {
        // Set the key
        K0 = getUInt32( key );
        K1 = getUInt32( key + 4 );
        // and reset the message
        clear();
    }

    Michael::Michael( Byte * key )
    {
        setKey( key );
    }

    Michael::~Michael()
    {
        // Wipe the key material
        K0 = 0;
        K1 = 0;

        // And the other fields as well.
        //Note that this sets (L,R) to (K0,K1) which is just fine.
        clear();
    }

    void Michael::appendByte( Byte b )
    {
        // Append the byte to our word-sized buffer
        M |= b << (8*nBytesInM);
        nBytesInM++;
        // Process the word if it is full.
        if( nBytesInM >= 4 )
        {
            L ^= M;
            R ^= ROL32( L, 17 );
            L += R;
            R ^= ((L & 0xff00ff00) >> 8) | ((L & 0x00ff00ff) << 8);
            L += R;
            R ^= ROL32( L, 3 );
            L += R;
            R ^= ROR32( L, 2 );
            L += R;
            // Clear the buffer
            M = 0;
            nBytesInM = 0;
        }
    }

    void Michael::append( Byte * src, int nBytes )
    {
        // This is simple
        while( nBytes > 0 )
        {
            appendByte( *src++ );
            nBytes--;
        }
    }

    void Michael::getMIC( Byte * dst )
    {
        // Append the minimum padding
        appendByte( 0x5a );
        appendByte( 0 );
        appendByte( 0 );
        appendByte( 0 );
        appendByte( 0 );
        // and then 0s until the length is a multiple of 4
        while( nBytesInM != 0 )
        {
            appendByte( 0 );
        }
        // The appendByte function has already computed the result.
        putUInt32( dst, L );
        putUInt32( dst+4, R );
        // Reset to the empty message.
        clear();
    }

    void Michael::hexToBin( char *src, Byte * dst )
    {
        // Simple wrapper
        hexToBin( src, strlen( src ), dst );
    }

    void Michael::hexToBin( char *src, int nChars, Byte * dst )
    {
        assert( (nChars & 1) == 0 );
        int nBytes = nChars/2;
        // Straightforward conversion
        for( int i=0; i<nBytes; i++ )
        {
            dst[i] = (Byte)((hexToBinNibble( src[0] ) << 4) |
                hexToBinNibble( src[1] ));
            src += 2;
        }
    }

    Byte Michael::hexToBinNibble( char c )
    {
        if( '0' <= c && c <= '9' )
        {
            return (Byte)(c - '0');
        }
        // Make it upper case
        c &= ~('a'-'A');

        assert( 'A' <= c && c <= 'F' );
        return (Byte)(c - 'A' + 10);
    }

    void Michael::runSingleTest( char * cKey, char * cMsg, char * cResult )
    {
        Byte key[ 8 ];
        Byte result[ 8 ];
        Byte res[ 8 ];

        // Convert key and result to binary form
        hexToBin( cKey, key );
        hexToBin( cResult, result );

        // Compute the MIC value
        Michael mic( key );
        mic.append( (Byte *)cMsg, strlen( cMsg) );
        mic.getMIC( res );

        // Check that it matches
        assert( memcmp( res, result, 8 ) == 0 );
    }

    void Michael::runTestPlan()
    // As usual, test plans can be quite tedious but this should
    // ensure that the implementation runs as expected.
    {
```

```
Byte key[8] ;
Byte msg[12];
int i;

// First we test the test vectors for the block function

// The case (0,0)
putUInt32( key, 0 );
putUInt32( key+4, 0 );
putUInt32( msg, 0 );

Michael mic( key );
mic.append( msg, 4 );

assert( mic.L == 0 && mic.R == 0 );

// The case (0,1)
putUInt32( key, 0 );
putUInt32( key+4, 1 );
mic.setKey( key );
mic.append( msg, 4 );

assert( mic.L == 0xc00015a8 && mic.R == 0xc0000b95 );

// The case (1,0)
putUInt32( key, 1 );
putUInt32( key+4, 0 );
mic.setKey( key );
mic.append( msg, 4 );

assert( mic.L == 0x6b519593 && mic.R == 0x572b8b8a );

// The case (01234567, 83659326)
putUInt32( key, 0x01234567 );
putUInt32( key+4, 0x83659326 );
mic.setKey( key );
mic.append( msg, 4 );

assert( mic.L == 0x441492c2 && mic.R == 0x1d8427ed );

// The repeated case
putUInt32( key, 1 );
putUInt32( key+4,0 );
mic.setKey( key );

for( i=0; i<1000; i++ )
```

```
    {
        mic.append( msg, 4 );
    }

    assert( mic.L == 0x9f04c4ad && mic.R == 0x2ec6c2bf );

    // And now for the real test cases
    runSingleTest( "0000000000000000", ""         , "82925c1ca1d130b8" );
    runSingleTest( "82925c1ca1d130b8", "M"        , "434721ca40639b3f" );
    runSingleTest( "434721ca40639b3f", "Mi"       , "e8f9becae97e5d29" );
    runSingleTest( "e8f9becae97e5d29", "Mic"      , "90038fc6cf13c1db" );
    runSingleTest( "90038fc6cf13c1db", "Mich"     , "d55e100510128986" );
    runSingleTest( "d55e100510128986", "Michael"  , "0a942b124ecaa546" );
}
```

M.3 PRF reference implementation and test vectors

M.3.1 PRF reference code

```
/*
 * PRF -- Length of output is in octets rather than bits
 *     since length is always a multiple of 8 output array is
 *     organized so first N octets starting from 0 contains PRF output
 *
 *     supported inputs are 16, 24, 32, 48, 64
 *     output array must be 80 octets to allow for sha1 overflow
 */
void PRF(
    unsigned char *key, int key_len,
    unsigned char *prefix, int prefix_len,
    unsigned char *data, int data_len,
    unsigned char *output, int len)
{
    int i;
    unsigned char input[1024]; /* concatenated input */
    int currentindex = 0;
    int total_len;

    memcpy(input, prefix, prefix_len);
    input[prefix_len] = 0; /* single octet 0 */
    memcpy(&input[prefix_len+1], data, data_len);
    total_len = prefix_len + 1 + data_len;
    input[total_len] = 0; /* single octet count, starts at 0 */
    total_len++;
    for (i = 0; i < (len+19)/20; i++) {
            hmac_sha1(input, total_len, key, key_len,
               &output[currentindex]);
            currentindex += 20;/* next concatenation location */
            input[total_len-1]++; /* increment octet count */
        }
    }
```

M.3.2 PRF test vectors

```
Test case 1
Key            0x0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b
Key length     20
Prefix         “prefix”
Prefix length  6
Data           "Hi There"
Data length    8
PRF-512        0xbcd4c650b30b9684951829e0d75f9d54
               0xb862175ed9f00606e17d8da35402ffee
               0x75df78c3d31e0f889f012120c0862beb
               0x67753e7439ae242edb8373698356cf5a

Test case 2
Key            "Jefe"
Key length     4
Prefix         “prefix”
Prefix length  6
Data           "what do ya want for nothing?"
Data length    28
PRF-512        0x51f4de5b33f249adf81aeb713a3c20f4
               0xfe631446fabdfa58244759ae58ef9009
               0xa99abf4eac2ca5fa87e692c440eb4002
               0x3e7babb206d61de7b92f41529092b8fc

Test case 3
Key            0xaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
Key length     20
Prefix         “prefix”
Prefix length  6
Data           0xdd repeated 50 times
Data length    50
PRF-512        0xe1ac546ec4cb636f9976487be5c86be1
               0x7a0252ca5d8d8df12cfb0473525249ce
               0x9dd8d177ead710bc9b590547239107ae
               0xf7b4abd43d87f0a68f1cbd9e2b6f7607
```

M.4 Suggested pass-phrase-to-PSK mapping

M.4.1 Introduction

The RSNA PSK consists of 256 bits, or 64 octets when represented in hex. It is difficult for a user to correctly enter 64 hex characters. Most users, however, are familiar with passwords and pass-phrases and feel more comfortable entering them than entering keys. A user is more likely to be able to enter an ASCII password or pass-phrase, even though doing so limits the set of possible keys. This suggests that the best that can be done is to introduce a pass-phrase to PSK mapping.

This subclause defines a pass-phrase–to–PSK mapping that is the recommended practice for use with RSNAs. This pass-phrase mapping was introduced to encourage users unfamiliar with cryptographic concepts to enable the security features of their WLAN.

Keys derived from the pass phrase provide relatively low levels of security, especially with keys generated form short passwords, since they are subject to dictionary attack. Use of the key hash is recommended only where it is impractical to make use of a stronger form of user authentication. A key generated from a pass-phrase of less than about 20 characters is unlikely to deter attacks.

The pass-phrase mapping defined in this subclause uses the PBKDF2 method from PKCS #5 v2.0 [B53].

$$PSK = \text{PBKDF2}(PassPhrase, ssid, ssidLength, 4096, 256)$$

Here, the following assumptions apply:

- A pass-phrase is a sequence of between 8 and 63 ASCII-encoded characters. The limit of 63 comes from the desire to distinguish between a pass-phrase and a PSK displayed as 64 hexadecimal characters.
- Each character in the pass-phrase must have an encoding in the range of 32 to 126 (decimal), inclusive.
- *ssid* is the SSID of the ESS or IBSS where this pass-phrase is in use, encoded as an octet string used in the Beacon and Probe Response frames for the ESS or IBSS.
- *ssidLength* is the number of octets of the *ssid*.
- 4096 is the number of times the pass-phrase is hashed.
- 256 is the number of bits output by the pass-phrase mapping.

M.4.2 Reference implementation

```
/*
 * F(P, S, c, i) = U1 xor U2 xor ... Uc
 * U1 = PRF(P, S || Int(i))
 * U2 = PRF(P, U1)
 * Uc = PRF(P, Uc-1)
 */

void F(
    char *password,
    unsigned char *ssid,
    int ssidlength,
    int iterations,
    int count,
    unsigned char *output)
{
    unsigned char digest[36], digest1[A_SHA_DIGEST_LEN];
    int i, j;

    for (i = 0; i < strlen(password); i++) {
        assert((password[i] >= 32) && (password[i] <= 126));
    }
    /* U1 = PRF(P, S || int(i)) */
    memcpy(digest, ssid, ssidlength);
    digest[ssidlength] = (unsigned char)((count>>24) & 0xff);
    digest[ssidlength+1] = (unsigned char)((count>>16) & 0xff);
    digest[ssidlength+2] = (unsigned char)((count>>8) & 0xff);
    digest[ssidlength+3] = (unsigned char)(count & 0xff);
    hmac_sha1(digest, ssidlength+4, (unsigned char*) password,
        (int) strlen(password), digest, digest1);

    /* output = U1 */
    memcpy(output, digest1, A_SHA_DIGEST_LEN);

    for (i = 1; i < iterations; i++) {
        /* Un = PRF(P, Un-1) */
        hmac_sha1(digest1, A_SHA_DIGEST_LEN, (unsigned char*) password,
            (int) strlen(password), digest);
        memcpy(digest1, digest, A_SHA_DIGEST_LEN);

        /* output = output xor Un */
        for (j = 0; j < A_SHA_DIGEST_LEN; j++) {
            output[j] ^= digest[j];
        }
    }
}

/*
 * password - ascii string up to 63 characters in length
 * ssid - octet string up to 32 octets
 * ssidlength - length of ssid in octets
 * output must be 40 octets in length and outputs 256 bits of key
 */
int PasswordHash (
    char *password,
    unsigned char *ssid,
    int ssidlength,
    unsigned char *output)
{
    if ((strlen(password) > 63) || (ssidlength > 32))
        return 0;

    F(password, ssid, ssidlength, 4096, 1, output);
    F(password, ssid, ssidlength, 4096, 2,
        &output[A_SHA_DIGEST_LEN]);
    return 1;
}
```

M.4.3 Test vectors

```
Test case 1
Passphrase = “password”
SSID = { ‘I’, ‘E’, ‘E’, ‘E’ }
SSIDLength = 4
PSK = f42c6fc52df0ebef9ebb4b90b38a5f90 2e83fe1b135a70e23aed762e9710a12e

Test case 2
Passphrase = “ThisIsAPassword”
SSID = { ‘T’, ‘h’, ‘i’, ‘s’, ‘I’, ‘s’, ‘A’, ‘S’, ‘S’, ‘I’, ‘D’ }
SSIDLength = 11
PSK = 0dc0d6eb90555ed6419756b9a15ec3e3 209b63df707dd508d14581f8982721af

Test case 3
Password = “aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa”
SSID = {‘Z’,‘Z’,‘Z’,‘Z’, ‘Z’,‘Z’,‘Z’,‘Z’, ‘Z’,‘Z’,‘Z’,‘Z’, ‘Z’,‘Z’,‘Z’,‘Z’,
            ‘Z’,‘Z’,‘Z’,‘Z’, ‘Z’,‘Z’,‘Z’,‘Z’, ‘Z’,‘Z’,‘Z’,‘Z’,‘Z’,‘Z’,‘Z’,‘Z’}
SSIDLength = 32
PSK = becb93866bb8c3832cb777c2f559807c 8c59afcb6eae734885001300a981cc62
```

M.5 Suggestions for random number generation

M.5.1 General

In order to properly implement cryptographic protocols, every platform needs the ability to generate cryptographic-quality random numbers. IETF RFC 4086 explains the notion of cryptographic-quality random numbers and provides advice on ways to harvest suitable randomness. It recommends sampling multiple sources, each of which contains some randomness, and by passing the complete set of samples through a PRF. By following this advice, an implementation can usually collect enough randomness to distill into a seed for a PRNG whose output is unpredictable.

This clause suggests two sample techniques that can be combined with the other recommendations of IETF RFC 4086 to harvest randomness. The first method is a software solution that can be implemented on most hardware; the second is a hardware-assisted solution. These solutions are expository only, to demonstrate that it is feasible to harvest randomness on any IEEE 802.11 platform. They are not mutually exclusive, and they do not preclude the use of other sources of randomness when available, such as a noisy diode in a power circuit; in this case, the more the merrier. As many sources of randomness as possible should be gathered into a buffer, and then hashed, to obtain a seed for the PRNG.

M.5.2 Software sampling

Due to the nature of clock circuits in modern electronics, there is some lack of correlation between two clocks in two different pieces of equipment, even when high-quality crystals are used—crystal clocks are subject to jitter, noise, drift, and frequency mismatch. This randomness may be as little as the placement of the clock waveform edges. Even if one entity were to attempt to synchronize itself to another entity’s clock, the correlation cannot be perfect due to noise and uncertainties of the synchronization.

Two clock circuits in the same piece of equipment may synchronize in frequency, but again the correlation is not perfect due to the noise and jitter of the circuits.

The randomness between the two clocks may not be much per sample—a tenth of a bit or less—but enough samples may be collected to gather enough randomness to form a seed.

A device can use software methods to take advantage of this lack of synchronization, to collect randomness from different sources. As an example, an AP might measure the frame arrival times on Ethernet wireless ports. There is always some amount of traffic on modern Ethernets: ARPs, DHCP requests, NetBIOS advertisements, etc. The sample algorithm in this subclause uses this traffic. In the example, an AP obtains randomness from the available traffic. If Ethernet traffic is available, the AP utilizes that for a source of randomness. Otherwise, it waits for the first association and creates traffic from which it can obtain randomness.

The clocks used to time the frames should be the highest resolution available, preferably 1 ms resolution or better. The clock used to time frame arrival should not be related to the clock used for frame serialization.

```
Initialize result to empty array
LoopCounter = 0
Wait until Ethernet traffic or association
Repeat until global key counter "random enough" or 32 times {
    result = PRF-256(0, "Init Counter",
    Local Mac Address || Time || result || LoopCounter)
    LoopCounter++
    Repeat 32 times {
        If Ethernet traffic available then {
            Take least significant octet of the timestamp when Ethernet packet is received
            Concatenate this octet onto result
        } else {
            Start 4-Way Handshake; after receipt of Message 2, deauthenticate
            Take least significant octet of the timestamp of when Message 1 is sent
            Take least significant octet of the timestamp of when Message 2 is received
            Take the least significant octet of RSSI from Message 2
            Take SNonce from Message 2
            Concatenate the sent time, received time, RSSI and SNonce octets onto result
        }
    }
}
Global key counter = result = PRF-256(0, "Init Counter",
Local Mac Address || Time || result || LoopCounter)
```

NOTE—The Time is set to 0 if it is not available.

M.5.3 Hardware-assisted solution

The sample implementation in this subclause uses hardware ring oscillators to generate randomness, as depicted in Figure M-1.

The circuit in Figure M-1 generates randomness. The clock input should be about the same frequency as the ring oscillator’s natural frequencies. The LFSR should be chosen to be one that is maximal length. Sample LFSRs can be found in Arazi [B6].

The three ring oscillators should be isolated from each other as much as possible to avoid harmonic locking between them. In addition, the three ring oscillators should not be near any other clock circuitry within the system to avoid these ring oscillators’ locking to system clocks. The oscillators should be tested to ensure that their output is not correlated.

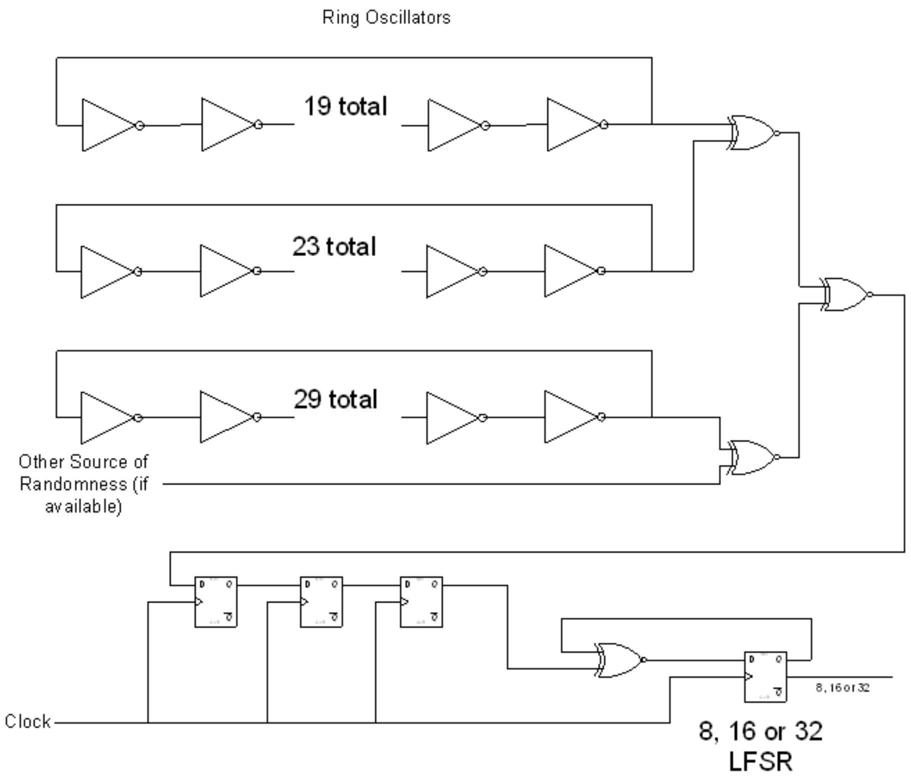


Figure M-1—Randomness generating circuit

The output of the LFSR is read by software and concatenated until enough randomness is collected. As a rule of thumb, reading from the LFSR 8 to 16 times the number of bits as the desired number of random bits is sufficient.

```
Initialize result to empty array
Repeat 1024 times {
    Read LFSR
    result = result | LFSR
    Wait a time period
}
Global key counter = PRF-256(0, "Init Counter", result)
```

M.6 Additional test vectors

M.6.1 Notation

In the examples that follow in M.6, frames are represented as a stream of octets, each octet in hex notation, sometimes with text annotation. The order of transmission for octets is left to right, top to bottom. For example, consider the following representation of a frame in Table M-3.

Table M-3—Notation example

Description #1	00 01 02 03 04 05
Description #2	06 07 08

The frame consists of 9 octets, represented in hex notation as “00”, “01”, ..., “08”. The octet represented by “00” is transmitted first, and the octet represented by “08” is transmitted last. Similar tables are used for other purposes, such as describing a cryptographic operation.

In the text discussion outside of tables, integer values are represented in either hex notation using a “0x” prefix or in decimal notation using no prefix. For example, the hex notation 0x12345 and the decimal notation 74565 represent the same integer value.

M.6.2 WEP cryptographic encapsulation

The discussion in this subclause represents an ARC4 encryption using a table that shows the key, plaintext input, and cipher text output. The MPDU data, prior to WEP cryptographic encapsulation, is shown in Table M-4.

Table M-4—Sample plaintext MPDU

MPDU data	aa aa 03 00 00 00 08 00 45 00 00 4e 66 1a 00 00 80 11 be 64 0a 00 01 22 0a ff ff ff 00 89 00 89 00 3a 00 00 80 a6 01 10 00 01 00 00 00 00 00 00 20 45 43 45 4a 45 48 45 43 46 43 45 50 46 45 45 49 45 46 46 43 43 41 43 41 43 41 43 41 43 41 41 41 00 00 20 00 01

ARC4 encryption is performed as shown in Table M-5.

Table M-5—ARC4 encryption

Key	fb 02 9e 30 31 32 33 34
Plaintext	aa aa 03 00 00 00 08 00 45 00 00 4e 66 1a 00 00 80 11 be 64 0a 00 01 22 0a ff ff ff 00 89 00 89 00 3a 00 00 80 a6 01 10 00 01 00 00 00 00 00 00 20 45 43 45 4a 45 48 45 43 46 43 45 50 46 45 45 49 45 46 46 43 43 41 43 41 43 41 43 41 43 41 41 41 00 00 20 00 01 1b d0 b6 04
Cipher text	f6 9c 58 06 bd 6c e8 46 26 bc be fb 94 74 65 0a ad 1f 79 09 b0 f6 4d 5f 58 a5 03 a2 58 b7 ed 22 eb 0e a6 49 30 d3 a0 56 a5 57 42 fc ce 14 1d 48 5f 8a a8 36 de a1 8d f4 2c 53 80 80 5a d0 c6 1a 5d 6f 58 f4 10 40 b2 4b 7d 1a 69 38 56 ed 0d 43 98 e7 ae e3 bf 0e 2a 2c a8 f7

The plaintext consists of the MPDU data, followed by a 4-octet CRC-32 calculated over the MPDU data.

The expanded MPDU, after WEP cryptographic encapsulation, is shown in Table M-6.

Table M-6—Expanded MPDU after WEP encapsulation

IV	fb 02 9e 80
MPDU data	f6 9c 58 06 bd 6c e8 46 26 bc be fb 94 74 65 0a ad 1f 79 09 b0 f6 4d 5f 58 a5 03 a2 58 b7 ed 22 eb 0e a6 49 30 d3 a0 56 a5 57 42 fc ce 14 1d 48 5f 8a a8 36 de a1 8d f4 2c 53 80 80 5a d0 c6 1a 5d 6f 58 f4 10 40 b2 4b 7d 1a 69 38 56 ed 0d 43 98 e7 ae e3 bf 0e
ICV	2a 2c a8 f7

The IV consists of the first 3 octets of the ARC4 key, followed by an octet containing the Key ID value in the upper 2 bits. In this example, the Key ID value is 2. The MPDU data consists of the cipher text, excluding the last 4 octets. The ICV consists of the last 4 octets of the cipher text, which is the encrypted CRC-32 value.

M.6.3 TKIP test vector

An example of a TKIP MSDU is provided in Table M-7 and Table M-8. The key and PN are used to create the IV, Phase1, and Phase2 keys.

Table M-7—Sample TKIP parameters

Source MAC Address	02 03 04 05 06 07
Destination MAC Address	02 03 04 05 06 08
Key	12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34
PN	0x000000000001
IV	00 20 01 20 00 00 00 00
Phase1	bb 58 07 1f 9e 93 b4 38 25 4b
Phase2	00 20 01 4c fe 67 be d2 7c 86 7b 1b f8 02 8b 1c

M.6.4 CCMP test vector

```
==== CCMP test mpdu ====

-- MPDU Fields

Version  = 0
Type     = 2   SubType  = 0  Data
ToDS     = 0   FromDS   = 0
MoreFrag = 0   Retry    = 1
PwrMgt   = 0   moreData = 0
Encrypt  = 1
Order    = 0
```

Table M-8—Sample plaintext and cipher text MPDUs, using parameter from Table M-7

Plaintext MPDU with TKIP MIC	08 42 2c 00 02 03 04 05 06 08 02 03 04 05 06 07 02 03 04 05 06 07 d0 02 00 20 01 20 00 00 00 00 aa aa 03 00 00 00 08 00 45 00 00 54 00 00 40 00 40 01 a5 55 c0 a8 0a 02 c0 a8 0a 01 08 00 3a b0 00 00 00 00 cd 4c 05 00 00 00 00 00 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 36 37 68 81 a3 f3 d6 48 d0 3c
Encrypted MPDU with MIC and ICV	08 42 2c 00 02 03 04 05 06 08 02 03 04 05 06 07 02 03 04 05 06 07 d0 02 00 20 01 20 00 00 00 00 c0 0e 14 fc e7 cf ab c7 75 47 e6 66 e5 7c 0d ac 70 4a 1e 35 8a 88 c1 1c 8e 2e 28 2e 38 01 02 7a 46 56 05 5e e9 3e 9c 25 47 02 e9 73 58 05 dd b5 76 9b a7 3f 1e bb 56 e8 44 ef 91 22 85 d3 dd 6e 54 1e 82 38 73 55 8a db a0 79 06 8a bd 7f 7f 50 95 96 75 ac c4 b4 de 9a a9 9c 05 f2 89 a7 c5 2f ee 5b fc 14 f6 f8 e5 f8

```
Duration = 11459
A1 = 0f-d2-e1-28-a5-7c    DA
A2 = 50-30-f1-84-44-08    SA
A3 = ab-ae-a5-b8-fc-ba    BSSID
SC = 0x3380
seqNum = 824 (0x0338)  fraqNum = 0 (0x00)
Algorithm = AES_CCM
Key ID = 0

TK = c9 7c 1f 67 ce 37 11 85  51 4a 8a 19 f2 bd d5 2f

PN = 199027030681356  (0xB5039776E70C)

802.11 Header = 08 48 c3 2c 0f d2 e1 28 a5 7c 50 30 f1 84 44 08 ab ae a5 b8 fc ba
             80 33

Muted 802.11 Header = 08 40 0f d2 e1 28 a5 7c 50 30 f1 84 44 08 ab ae a5 b8 fc ba
             00 00

CCMP Header = 0c e7 00 20 76 97 03 b5

CCM Nonce = 00 50 30 f1 84 44 08 b5  03 97 76 e7 0c

Plaintext Data = f8 ba 1a 55 d0 2f 85 ae 96 7b b6 2f b6 cd a8 eb 7e 78 a0 50

CCM MIC = 78 45 ce 0b 16 f9 76 23

-- Encrypted MPDU with FCS

08 48 c3 2c 0f d2 e1 28 a5 7c 50 30 f1 84 44 08 ab ae a5 b8 fc ba 80 33 0c e7 00
             20 76 97 03 b5 f3 d0 a2 fe 9a 3d bf 23 42 a6 43 e4 32 46 e8 0c 3c 04
             d0 19 78 45 ce 0b 16 f9 76 23 1d 99 f0 66
```

M.6.5 PRF test vectors

A set of test vectors are provided for each size of PRF function used in this subclause. See Table M-9 to Table M-12. The inputs to the PRF function are strings for key, prefix, and data. The length can be any multiple of 8, but the values 192, 256, 384, and 512 are used in this subclause. The test vectors were taken from IETF RFC 2202-1997 [B26] with additional vectors added to test larger key and data sizes.

Table M-9—RSN PRF Test Vector 1

Test_case	1
Key	0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b
Prefix	"prefix"
Data	"Hi There"
Length	192
PRF-192	bc d4 c6 50 b3 0b 96 84 95 18 29 e0 d7 5f 9d 54 b8 62 17 5e d9 f0 06 06

Table M-10—RSN PRF Test Vector 2

Test_case	2
Key	'Jefe'
Prefix	"prefix-2"
Data	"what do ya want for nothing?"
Length	256
PRF-256	47 c4 90 8e 30 c9 47 52 1a d2 0b e9 05 34 50 ec be a2 3d 3a a6 04 b7 73 26 d8 b3 82 5f f7 47 5c

Table M-11—RSN PRF Test Vector 3

Test_case	3
Key	aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa
Prefix	"prefix-3"
Data	"Test Using Larger Than Block-Size Key - Hash Key First"
Length	384
PRF-384	0a b6 c3 3c cf 70 d0 d7 36 f4 b0 4c 8a 73 73 25 55 11 ab c5 07 37 13 16 3b d0 b8 c9 ee b7 e1 95 6f a0 66 82 0a 73 dd ee 3f 6d 3b d4 07 e0 68 2a

Table M-12—RSN PRF Test Vector 4

Test_case	4
Key	0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b
Prefix	"prefix-4"
Data	"Hi There Again"
Length	512
PRF-512	24 8c fb c5 32 ab 38 ff a4 83 c8 a2 e4 0b f1 70 eb 54 2a 2e 09 16 d7 bf 6d 97 da 2c 4c 5c a8 77 73 6c 53 a6 5b 03 fa 4b 37 45 ce 76 13 f6 ad 68 e0 e4 a7 98 b7 cf 69 1c 96 17 6f d6 34 a5 9a 49

M.7 Key hierarchy test vectors for pairwise keys

M.7.1 General

The test vectors in this subclause provide an example of PTK derivation for both CCMP and TKIP.

Pairwise keys are derived from the PMK, AA, SPA, SNonce, and ANonce. The values in Table M-13 are used as input to the pairwise key derivation test vectors.

Table M-13—Sample values for pairwise key derivations

PMK	0d c0 d6 eb 90 55 5e d6 41 97 56 b9 a1 5e c3 e3 20 9b 63 df 70 7d d5 08 d1 45 81 f8 98 27 21 af
AA	a0 a1 a1 a3 a4 a5
SPA	b0 b1 b2 b3 b4 b5
SNonce	c0 c1 c2 c3 c4 c5 c6 c7 c8 c9 d0 d1 d2 d3 d4 d5 d6 d7 d8 d9 da db dc dd de df e0 e1 e2 e3 e4 e5
ANonce	e0 e1 e2 e3 e4 e5 e6 e7 e8 e9 f0 f1 f2 f3 f4 f5 f6 f7 f8 f9 fa fb fc fd fe ff 00 01 02 03 04 05

M.7.2 CCMP pairwise key derivation

Using the values from Table M-13 for PMK, AA, SPA, SNonce, and ANonce, the key derivation process for CCMP generates a temporal key as shown in Table M-14.

Table M-14—Sample derived CCMP temporal key (TK)

TK	b2 36 0c 79 e9 71 0f dd 58 be a9 3d ea f0 65 99

M.7.3 TKIP pairwise key derivation

Using the values from Table M-13 for PMK, AA, SPA, SNonce, and ANonce, the key derivation process for TKIP generates the values shown in Table M-15.

Table M-15—Sample derived PTK

KCK	`37 9f 98 52 d0 19 92 36  b9 4e 40 7c e4 c0 0e c8`
KEK	`47 c9 ed c0 1c 2c 6e 5b  49 10 ca dd fb 3e 51 a7`

TK	`b2 36 0c 79 e9 71 0f dd  58 be a9 3d ea f0 65 99`
	`db 98 0a fb c2 9c 15 28  55 74 0a 6c e5 ae 38 27`

Authenticator Tx MIC_key	`db 98 0a fb c2 9c 15 28`
Supplicant Tx MIC_key	`55 74 0a 6c e5 ae 38 27`

M.8 Test vectors for AES-128-CMAC

Test vectors for AES-128-CMAC are in Annex D.1 of NIST SP-800-38B.

M.9 Management frame protection test vectors

M.9.1 BIP with broadcast Deauthentication frame

Unprotected broadcast Deauthentication frame (without FCS):
c0 00 00 00 ff ff ff ff ff ff 02 00 00 00 00 00 02 00 00 00 00 00 09 00 02 00

FC=c0 00
DUR=00 00
DA=ff ff ff ff ff ff
SA=02 00 00 00 00 00
BSSID=02 00 00 00 00 00
SEQ=09 00
Reason Code: 02 00

IGTK: 4e a9 54 3e 09 cf 2b 1e ca 66 ff c5 8b de cb cf

BIP AAD (FC | A1 | A2 | A3): c0 00 ff ff ff ff ff ff 02 00 00 00 00 00 02 00 00 00 00 00

Management Frame Body: 02 00

MME (with MIC=0): 4c 10 04 00 04 00 00 00 00 00 00 00 00 00 00 00 00 00

AES-128-CMAC(AAD | Management Frame Body | MME): 48 df bf a7 b8 27 88 72

Protected broadcast Deauthentication frame (without FCS):

c0 00 00 00 ff ff ff ff ff ff 02 00 00 00 00 00 02 00 00 00 00 00 09 00 02 00 4c 10 04 00 04 00 00 00 00 00 48 df bf a7 b8 27 88 72

FC=c0 00 (note: Protected flag is _not_ set)
DUR=00 00
DA=ff ff ff ff ff ff
SA=02 00 00 00 00 00
BSSID=02 00 00 00 00 00
SEQ=09 00
Reason Code: 02 00

MME: 4c 10 04 00 04 00 00 00 00 00 48 df bf a7 b8 27 88 72

(KeyID = 04 00 (= 4), Seq# = 04 00 00 00 00 00, MIC = 48 df bf a7 b8 27 88 72)

M.9.2 CCMP with unicast Deauthentication frame

Plaintext unicast Deauthentication frame (without FCS):
c0 00 00 00 02 00 00 00 01 00 02 00 00 00 00 00 02 00 00 00 00 00 60 00 02 00

FC=c0 00
DUR=00 00
DA=02 00 00 00 01 00
SA=02 00 00 00 00 00
BSSID=02 00 00 00 00 00
SEQ=60 00
Reason Code: 02 00

CCMP TK: 66 ed 21 04 2f 9f 26 d7 11 57 06 e4 04 14 cf 2e

```
CCM flags: 59 (Adata: 1, M: 011, L: 001)
Nonce = Nonce Flags | A2 | PN
   = 10 (Management)
   02 00 00 00 00 00
    00 00 00 00 00 01
l(m) = 00 02

AAD = FC | A1 | A2 | A3 | SC | A4 | QC
   = c0 40 02 00 00 00 01 00 02 00 00 00 00 00 02 00 00 00 00 00 00 00
```

AAD blocks:
00 16 c0 40 02 00 00 00 01 00 02 00 00 00 00 00
02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Encrypted Deauthentication frame (without FCS):
c0 40 00 00 02 00 00 00 01 00 02 00 00 00 00 00 02 00 00 00 00 00 60 00 01 00 00 20 00 00 00 00 1d 07 ca fd 04 09 bb 8b af ef

FC=c0 40 (note: Protected bit set)
DUR=00 00
DA=02 00 00 00 01 00
SA=02 00 00 00 00 00
BSSID=02 00 00 00 00 00

SEQ=60 00
CCMP Header: 01 00 00 20 00 00 00 00 (PN=1, ExtIV=1)
Encrypted Data: 1d 07

MIC: ca fd 04 09 bb 8b af ef

M.10 SAE test vector

group: 19
Password: 'thisisreallysecret'
Local MAC address: 7b:88:56:20:2d:8d
Peer's MAC address: e2:47:1c:0a:5a:cb

H(e2:47:1c:0a:5a:cb | 7b:88:56:20:2d:8d, thisisreallysecret | 1)

```
69f69099 83675392 d0a3a882 47ffef20 413ee972 15872942 4415e139 46ecc206
```

candidate x value:

```
a16729e0 339c38f8 b06e2b83 76d43066 85578354 ab09d848 a0f140ac 825e6a3d
no solution to the equation of the elliptic curve with counter = 1
```

H(e2:47:1c:0a:5a:cb | 7b:88:56:20:2d:8d, thisisreallysecret | 2)

```
ab4b22f1 0e7cdbb2 9d1fd2de 5823198c 3c66733f 40e3f94a da06ee05 c83dac37
```

candidate x value:

```
103a5b96 8873bab0 fafc6dd8 ff3476ff 56487e7f 072b38e4 c9705497 1ce72b7b
```

PWE (x,y):

```
103a5b96 8873bab0 fafc6dd8 ff3476ff 56487e7f 072b38e4 c9705497 1ce72b7b
31742d39 f380f247 624218e9 45543004 d39973a5 68c5b904 b0cf5d36 2d44f3bf
```

local private value:

```
c5d7019e 7612d5f4 3cf91fe5 62b40bb8 b2640c65 c577b9b1 9994bf50 6baf2859
```

local mask value:

```
19d030fe 5bb11ee4 c27c9dfc 3c06520f 8fbe9290 059b0cc5 50db0d2b 9d3ac452
```

local commit:

```
dfa7329c d1c3f4d8 ff75bde1 9eba5dc8 42229ef5 cb12c676 ea6fcc7c 08e9ecab
3008b40e 01912bc5 3b862cd9 43305e86 46ee3b3e 6f745c5b b3ae8dfc 2ebf654e
d0a4e2a2 8bb98b62 9a4b0084 9df93d22 2999d086 5c9cceed a8e90fcb 53af5ae6
```

peer's commit:

```
10c1e1f1 d008713b 41986cdd 441eb991 bc823b60 118a5fc9 f51b16aa 00342147
19475f6f 50dbc87f 1505c109 e421a7e3 6b3a2e3f 48bfe52e 01b75f2b e7e5f4bc
948fe44c 741bd97f 51654857 7c6f320d 0c349939 857e0c79 30916d6f 323739d6
```

k:

```
6761b6a9 a9421297 f54a97ee c0daf188 49e582b4 bea7f06a cc34686f bd7900b1
```

keyseed:

```
ea72c928 2e364654 5faaa2c3 fb5791d4 74885907 7f3c7b04 208aceab 2ca9dd45
```

KCK:

```
cb7636d9 9b0dad17 2bd6a3fd 40bb76f4 4ecbb874 750396bc 74fba0ea 3a11f86c
```

local confirm:

```
01004664 47ab0962 ae780bcc 7a0ac672 a39c62ec 3009cfb2 34dd1918 37c792b9
548e
```

peer's confirm:

```
01002df5 f62c4610 5b606d76 72b89c3e 615421d2 6d9991da a8183778 811d30ac
e3db
```

PMK:

```
f6ecb8ad e3ae30df e35d31ea ee047161 b3a00d94 45c5dfd2 2cd8d8fb af83d9c7
```

Annex N

(informative)

Admission control

N.1 Example use of TSPEC for admission control

Admission control, in general, depends on vendors' implementations of schedulers, available channel capacity, link conditions, retransmission limits, and the scheduling requirements of a given TSPEC. However, for any given channel capacity, link conditions, and retransmission limits, some TSPEC constructions might be categorically rejected because a scheduler cannot create a meaningful schedule for that TSPEC. There must, for example, be a minimum number of specified fields in the TSPEC in order for the admission control mechanism to create a valid TSPEC. Table N-1 below lists the valid TSPEC parameters that must be present for all admission control algorithms to admit a TSPEC. This represents a set of necessary parameters in order for TSPEC to be admitted; it is not sufficient in and of itself to guarantee TSPEC admittance, which depends upon channel conditions and other factors. Such TSPECs are said to be *admissible*. In the table, S means specified, X means unspecified, and DC means "do not care."

Table N-1—Admissible TSPECs

TSPEC parameter	Continuous time QoS traffic (HCCA)	Controlled-access CBR traffic (HCCA)	Bursty traffic (HCCA)	Unspecified non-QoS traffic (HCCA)	Contention-based CBR traffic (EDCA)
Nominal MSDU Size	S	S	X	DC	S
Minimum Service Interval	S	Nominal MSDU size/mean data rate, if specified (VoIP typically uses this)	Mean data rate/ nominal MSDU size, if mean data rate specified	DC	DC
Maximum Service Interval	S	Delay bound/ number of retries (AV typically uses this)	Delay bound/ number of retries, if delay bound present	DC	DC
Inactivity Interval	Always specified				DC
Suspension Interval	DC				
Minimum Data Rate	Must be specified if peak data rate is specified	Equal to mean data rate	X	DC	DC
Mean Data Rate	S	S	DC	DC	S
Burst Size	X	X	S	DC	DC
Minimum PHY Rate	Always specified				

Table N-1—Admissible TSPECs ***(continued)***

TSPEC parameter	Continuous time QoS traffic (HCCA)	Controlled-access CBR traffic (HCCA)	Bursty traffic (HCCA)	Unspecified non-QoS traffic (HCCA)	Contention-based CBR traffic (EDCA)
Peak Data Rate	Must be specified if Minimum Data Rate Specified DC	Equal to Mean Data Rate	DC	DC	DC
Delay Bound	S	S	DC	X	X
Surplus Bandwidth Allowance	Must be specified if the delay bound is present			DC	S
Medium Time	X (not specified by non-AP STA; only an output from the HC)				

N.2 Recommended practices for contention-based admission control

N.2.1 Use of ACM (admission control mandatory) subfield

It is recommended that admission control not be required for the access categories AC_BE and AC_BK. The ACM subfield for these categories should be set to 0. The AC parameters chosen by the AP should account for unadmitted traffic in these ACs.

When dot11SSPNInterfaceActivated is true, it is recommended that any STA authenticated through an SSPN interface use admission control to access categories AC_VO and AC_VI to ensure network utilization consistent with the policy imposed by the SSPN for admission. AC parameters chosen by the AP should further account for any unadmitted traffic in AC_VO and AC_VI that may be reserved for users of a particular SSPN.

N.2.2 Deriving medium time

It is recommended that the AP use the following procedure to derive medium time in its ADDTS Response frame:

There are two requirements to consider: the traffic requirements of the application and the expected error performance of the medium. The application requirements are captured by two TSPEC parameters: Nominal MSDU Size and Mean Data Rate. The medium requirements are captured by two TSPEC parameters: Surplus Bandwidth Allowance and Minimum PHY Rate. The following formula describes how medium time can be calculated (assuming RTS/CTS protection is not used):

Medium Time = Surplus Bandwidth Allowance × pps × MPDUExchangeTime

where:
pps = Ceiling((Mean Data Rate / 8) / Nominal MSDU Size)
MPDUExchangeTime = duration(Nominal MSDU Size, Minimum PHY Rate) + SIFS + ACK duration (also see the definition of MPDUExchangeTime in 9.19.4.2.3)
duration() is the PLME-TXTIME primitive that returns the duration of a packet based on its payload size and the PHY data rate employed

N.3 Guidelines and reference design for sample scheduler and admission control unit

N.3.1 Guidelines for deriving service schedule parameters

The HC establishes the SI for each admitted TS for a STA to derive the aggregate minimum SI contained in the STA's service schedule. The SI for each TS is equal to a nonzero minimum SI contained in the TSPEC, if it exists; otherwise, it is the nominal MSDU size divided by the mean data rate. The SI contained in the service schedule is equal to the smallest SI for any TSPEC.

The HC can use an aggregate "token bucket specification" to police a STA's admitted flows. The HC must derive the aggregate mean data rate and aggregate burst size to establish the aggregate token bucket specification. The aggregate mean data rate is equal to the sum of the mean data rates of all of the STA's admitted TSs. The aggregate burst size is equal to the sum of the burst size of all of the STA's admitted TSs. An aggregate token bucket is initialized with the aggregate burst size. Tokens are added to the token bucket at the aggregate mean data rate.

When dot11SSPNInterfaceActivated is true, the HC polices all traffic flows from a non-AP STA authenticated against the maximum authorized data rates stored in the dot11InterworkingTable. Each SSPN-authenticated STA is given a maximum bandwidth allowance by the SSPN for each access category as well as scheduled access. The AP polices the SSPN-authenticated STA traffic flows to the maximum bandwidth allowance provided by the SSPN.

N.3.2 TSPEC construction

TSPECs are constructed at the SME from application requirements supplied via the SME and with information specific to the MAC layer. There are no normative requirements on how any TSPEC is to be generated. However, in this subclause a description is given of how and where certain parameters can be chosen. The following parameters typically arise from the application: Nominal MSDU Size, Maximum MSDU Size, Minimum Service Interval, Maximum Service Interval, Inactivity Interval, Minimum Data Rate, Mean Data Rate, Burst Size, Peak Data Rate, and Delay Bound. The following parameters are generated locally within the MAC: Minimum PHY Rate and Surplus Bandwidth Allowance, although the Maximum Service Interval and Minimum Service Intervals can be generated within the MLME as well. This subclause describes how the parameters that are typically generated within the MAC can be derived.

Note that a TSPEC can also be generated autonomously by the MAC without any initiation by the SME. However, if a TSPEC is generated subsequently by the SME, the TSPEC generated autonomously by the MAC is overridden. If one or more TSPECs are initiated by the SME, the autonomous TSPEC, containing the same TSID is terminated.

Typically, TSPEC parameters not determined by the application are built upon the assumption that the following exist:

- A probability *p* of not transmitting the frame (because it would have exceeded its delay bound)
- An MSDU length (which can be considered fixed for constant-bit-rate applications)
- Application throughput and delay requirements
- A channel model of error, in particular a channel error probability for the (fixed) frame length
- Possibly country-specific limits on TXOP limits

The minimum service interval, if determined within the MAC, can typically be given as the nominal MSDU size/mean data rate.

The maximum service interval, if determined within the MAC, can be calculated as the delay bound/number of retries possible. This number should be greater than the minimum SI, when that is specified. The number of retries can be chosen (as below) to meet a particular probability of dropping a packet because it exceeds its delay bound. Note that for multiple streams, this SI should be the aggregate of all SIs requested, because the STA is assigned the TXOPs, not any particular stream.

Typically, it can be assumed that the scheduler would attempt to schedule TXOPs distributed throughout a small multiple of beacon intervals (if not a single beacon interval). In addition, TXOP limits would typically be chosen to be as short as possible (within the constraints of the minimum PHY rate, acknowledgment policy, and so forth), consistent with the goal of maximizing throughput. In other words, because of overhead, not to mention the requirements for transmitting a single Poll frame, MPDU, and possibly ACK frame, the TXOPs need to be at least of a certain duration.

The channel model implies an error ratio and an assumption about dependency (joint probability distribution of channel errors sequentially, i.e., burst error probabilities).

For example, if the channel causes errors independently from frame to frame and the error probability is the same for all frames of the same length at all times, this channel would be said to be an independent, identically distributed error channel. With p as the probability of dropping the frame, and p_e as the probability of the frame not being transmitted successfully (i.e., either the data frame or the ACK frame associated with it is in error), let N_p be the number of retries required to maintain the probability of dropping the frame to be p.

The probability of any given packet being dropped in such a channel after N_p retries is given by

$$p_{\text{drop}} = (p_e)^{N_p + 1}$$

For example, in such a channel, if $p_e = 0.1$ and $p_{\text{drop}} = 10^{-8}$, then up to seven retries within the delay bound are required, and the scheduler should ensure that sufficient cumulative TXOP allocations are made to accommodate retransmissions.

The Surplus Bandwidth Allowance parameter ensures the requesting STA is allocated a minimum amount of excess time by the scheduler to so that application dropped packet rates are bounded. For example, this parameter can be chosen to ensure that when there is a 10% packet error ratio (PER) for 1000-octet packets, that there is a dropped PER (i.e., packets that fail to be received within the delay bound) of 10^{-8}. This parameter can be chosen based on an initial assumption regarding channel/source characteristics and renegotiated by sending ADDTS Request frames if required, based on actual transmission behavior. To understand how this parameter can be specified, consider the case where there are only 100 PPDUs to be transmitted and delay is not an issue. The PER p_e is 10%, with the errors happening independently from packet to packet. To accomplish this, the number of packets transmitted in each beacon interval must exceed the 100 PPDUs by N_{excess} in order to avoid dropping packets with some fixed probability (denoted as p_{drop}). For example, if $p_{\text{drop}} = 10^{-8}$, then the number of retries N_{excess} must satisfy to send *only* 100 packets successfully (based on Bernoulli distributed error probabilities):

$$p_{\text{drop}} = \sum_{k = N_{\text{excess}}}^{N_{\text{excess}} + 100} \binom{N_{\text{excess}} + 100}{k} p_e^k (1 - p_e)^{100 + N_{\text{excess}} - k}$$

where $p_e = 0.1$. Solution of an equation such as this yields the total number of additional retries. This can be found, for this example, using the fact that

$$\sum_{k=a}^{b} \binom{b}{k} p_e^k (1-p_e)^{b-k} = I_{p_e}(a, b-a+1)$$

where $I_{p_e}(a, b)$ is the Incomplete Beta function, and taking *n* sufficiently large (or invoking the law of large numbers). Solving this yields that, on the average, 38 additional MPDUs are required to keep the probability of dropping a packet to less than 10^{-8} to send only 100 packets. For this case, the

Surplus bandwidth allowance = $\frac{100 + N_{\text{excess}}}{100}$, which for this example would be 1.38.

This might represent an upper bound for the excess bandwidth for many applications: it presumes that the observation interval is 100 + N_{excess} frames. When the observation interval (or delay bound) is longer than the time it takes to transmit 100 frames, then it can be shown that the excess bandwidth required decreases. For example, if it were desired to send 100 000 frames with 12 000 frames excess, then the probability of a dropped frame becomes 1.6×10^{-15}.

On the other hand, suppose that an infinitely long stream was to be transmitted without any constraints on delay. In such a case, with an infinite number of retries and with a 10% PER, 1.111 times the bandwidth required to send MPDUs without error is required (because the probability that *n* retries are required for any packet is given by 0.1^n).

In fact, assuming a finite delay bound, the above result (1.111= surplus bandwidth allowance) represents a lower bound on what the surplus bandwidth allowance would be.

Typically, then the excess bandwidth required would be between a "send only *N* + excess packets" scenario within a given delay and "send an infinite number of packets with an infinite number of retries with no delay" scenario.

A more exact calculation can be done via simulation as follows: Suppose there are $N_{\text{allocated}}$ constant length PPDUs per beacon interval (these are actually transmitted on the air), and suppose there are N_{payload} constant length PPDUs to be transmitted. Suppose further that these transmitted and payload PPDUs arrive in a uniformly distributed manner in each beacon interval, then the delay incurred in waiting for a packet to be transmitted can be inferred from examining the transmit queue length and statistically can be inferred from examining how many retries within a certain period of time are required to keep the probability of a dropped packet below a certain amount. The number of retries (and consideration of IFS, polls, etc.) then determines the surplus bandwidth allowance.

In general, then the surplus bandwidth allowance can be given by $\frac{N_{\text{allocated}}}{N_{\text{payload}}}$.

Note that for the case of a Block Ack, a similar calculation applies, although the calculations for the excess bandwidth need to take into account the probability of failing to receive a Block Ack, and so forth.

N.3.3 Reference design for sample scheduler and admission control unit

This subclause provides the guidelines for the design of a simple scheduler and admission control unit (the unit that administers admission policy in the HC's SME) that meet the minimum performance requirements as specified in 9.19.4.3. The scheduler and admission control unit use the minimum set of mandatory TSPEC parameters as specified in 9.19.4.3.

N.3.3.1 Sample scheduler

This subclause includes the reference design for a sample scheduler. This scheduler uses the mandatory set of TSPEC parameters to generate a schedule: Mean Data Rate, Nominal MSDU Size, and Maximum Service Interval or Delay Bound. If both Maximum Service Interval and Delay Bound parameters are specified in the TSPEC, the scheduler uses the Maximum Service Interval parameter for the calculation of the schedule.

The schedule generated by the scheduler meets the normative behavior as specified in 9.19.4.3. The schedule for an admitted stream is calculated in two steps. The first step is the calculation of the scheduled SI. In the second step, the TXOP duration for a given SI is calculated for the stream.

The calculation of the scheduled service interval is done as follows: First, the scheduler calculates the minimum of all maximum SIs for all admitted streams. Let this minimum be *m*. Second, the scheduler chooses a number lower than *m* that is a submultiple of the beacon interval. This value is the scheduled SI for all STAs with admitted streams. See Figure N-1.

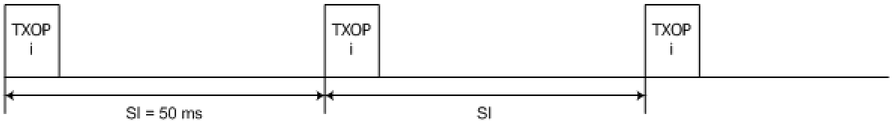


Figure N-1—Schedule for stream from STA *i*

For the calculation of the TXOP duration for an admitted stream, the scheduler uses the following parameters: Mean Data Rate and Nominal MSDU Size; and from the negotiated TSPEC, the Scheduled Service Interval calculated above, Physical Transmission Rate, Maximum Allowable Size of MSDU, and Overhead. The Physical Transmission Rate is the minimum PHY rate negotiated in the TSPEC. If the minimum PHY rate is not committed in the ADDTS Response frame, the scheduler can use the observed PHY rate as the Physical Transmission Rate. The Overhead includes IFSs, ACK frames and CF-Poll frames. For simplicity, details for the Overhead calculation are omitted in this description. The TXOP duration is calculated as follows: First, the scheduler calculates the number of MSDUs that can arrive at the Mean Data Rate during the SI:

$$N_i = \left\lceil \frac{SI \times \rho_i}{L_i} \right\rceil$$

where
SI is the Scheduled Service Interval
ρ is the Mean Data Rate
L is the Nominal MSDU Size

Then the scheduler calculates the TXOP duration as the maximum of

— Time to transmit N_i frames at R_i and
— Time to transmit one maximum size MSDU at R_i (plus overhead):

$$TXOP_i = max\left(\frac{N_i \times L_i}{R_i} + O, \frac{M}{R_i} + O\right)$$

where
R is the Physical Transmission Rate

M is the maximum possible size of MSDU, i.e., 2304 octets
O is the Overhead in time units

An example is shown in Figure N-1. Stream from STA i is admitted. The beacon interval is 100 ms and the maximum SI for the stream is 60 ms. The scheduler calculates a scheduled SI (SI) equal to 50 ms using the steps explained above is this annex.

The same process is repeated continuously while the maximum SI for the admitted stream is larger than the current SI. An example is shown in Figure N-2.

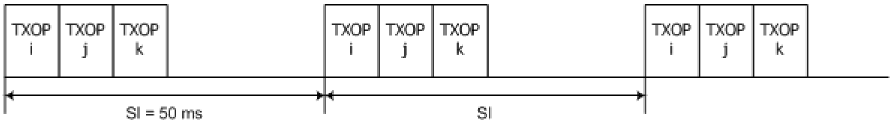


Figure N-2—Schedule for streams from STAs *i* to *k*

If a new stream is admitted with a maximum SI smaller than the current SI, the scheduler needs to change the current SI to a smaller number than the maximum SI of the newly admitted stream. Therefore, the TXOP duration for the current admitted streams needs also to be recalculated with the new SI.

If a stream is dropped, the scheduler might use the time available to resume contention. The scheduler might also choose to move the TXOPs for the STAs following the STA dropped to use the unused time. An example is shown in Figure N-3, when the stream for STA j is removed. However, this option might require the announcement of a new schedule to all STAs.

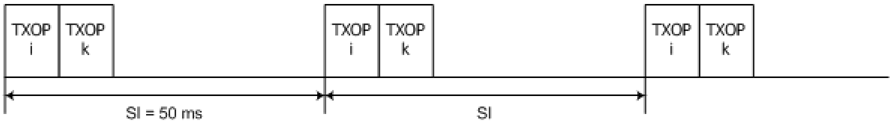


Figure N-3—Reallocation of TXOPs when a stream is dropped

Different modifications can be implemented to improve the performance of the minimum scheduler. For example, a scheduler might generate different scheduled SIs (SI) for different STAs, and/or a scheduler might consider accommodating retransmissions while allocating TXOP durations.

N.3.3.2 Admission control unit

This subclause describes a reference design for an admission control unit (ACU) that administers admission of TS. The ACU uses the same set of parameters that the scheduler uses in N.3.3.1.

When a new stream requests admission, the admission control process is done in three steps. First, the ACU calculates the number of MSDUs that arrive at the mean data rate during the scheduled SI. The scheduled SI (SI) is the one that the scheduler calculates for the stream as specified in N.3.3.1. For the calculation of the number of MSDUs, the ACU uses the equation for N_i shown in N.3.3.1. Second, the ACU calculates the TXOP duration that needs to be allocated for the stream. The ACU uses the equation for $TXOP_i$ shown in N.3.3.1. Finally, the ACU determines that the stream can be admitted when the following inequality is satisfied:

$$\frac{TXOP_{k+1}}{SI} + \sum_{i=1}^{k} \frac{TXOP_i}{SI} \le \frac{T - T_{CP}}{T}$$

where

k is the number of existing streams
$k + 1$ is used as index for the newly arriving stream
T indicates the beacon interval
T_{CP} is the time used for EDCA traffic

The ACU needs to ensure that it complies with the dot11CAPlimit, i.e., the scheduler does not allocate TXOPs that exceed dot11CAPlimit. The ACU might also consider additional time to allow for retransmissions.

The ACU ensures that all admitted streams have guaranteed access to the channel. Any modification can be implemented for the design of the ACU. For example, UP-based ACU is possible by examining the UP field in TSPEC to decide whether to admit, retain, or drop a stream. If the UP is not specified, a default value of 0 is used. If a higher UP stream needs to be serviced, an ACU might drop lower UP streams.

Annex O

(informative)

An example of encoding a TIM virtual bit map

O.1 Introduction

The purpose of this annex is to show an example of encoding a Partial Virtual Bit Map field of the TIM element of the MAC, as described in 8.4.2.7.

O.2 Examples

The following examples help clarify the use of TIM values, both with and without the Multiple BSSID capability.

The first example is one in which there are no group addressed MSDUs buffered in the AP but there is traffic for two STAs queued in the AP. STAs with AID 2 and AID 7 have data buffered in the AP. Figure O-1 shows the values of the Bit Map Control and Partial Virtual Bit Map fields that would be part of the TIM element for this example.

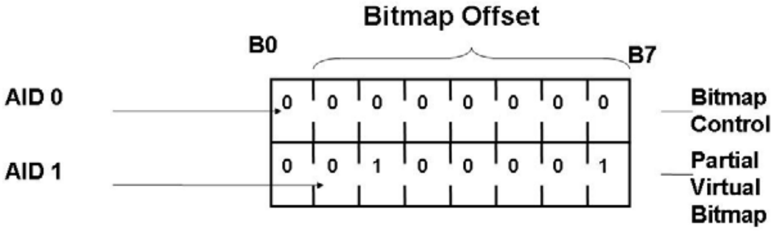


Figure O-1—Virtual bitmap example #1

The next example is one in which group addressed MSDUs are buffered in the AP as well as traffic for STAs. The DTIM Count field in the TIM element equals 0. STAs with AID 2, AID 7, AID 22 and AID 24 have data buffered in the AP. Figure O-2 shows the values of the Bit Map Control and Partial Virtual Bit Map fields.

Another example is one in which group addressed MSDUs are buffered in the AP as well as traffic for STAs. The DTIM Count field in the TIM element equals 0. Only the node with AID 24 has data buffered in the AP. In this example, the Bit Map Offset is used to start the Partial Virtual Bit Map at the third byte. Figure O-3 shows the values of the Bit Map Control and Partial Virtual Bit Map fields.

The three examples listed above describe the construction of the TIM Virtual Bitmap when the Multiple-BSSID capability is not supported. The following three examples demonstrate how to construct the TIM Virtual Bitmap, when Multiple-BSSID is supported.

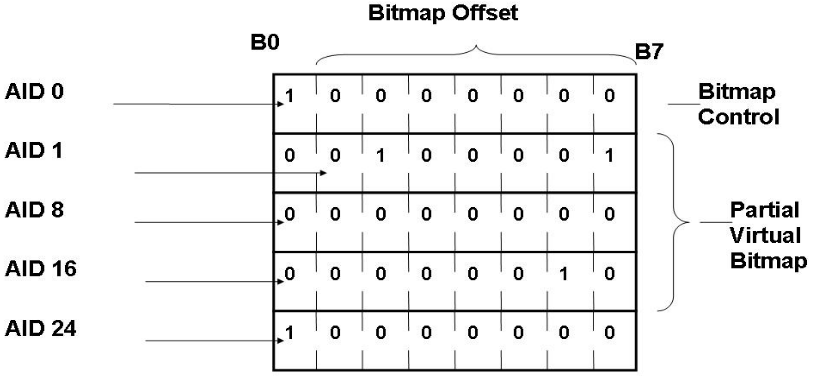


Figure O-2—Virtual bitmap example #2

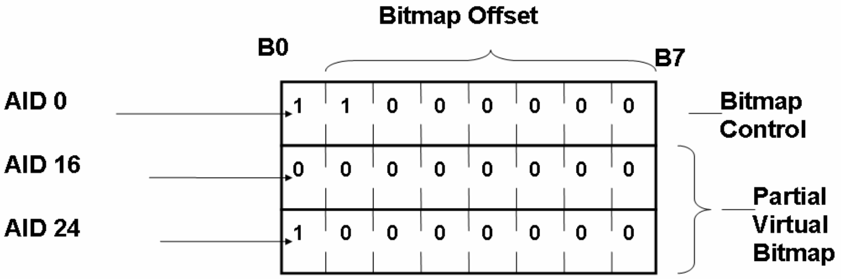


Figure O-3—Virtual bitmap example #3

The first example with Multiple BSSID is one in which there are eight BSSIDs and the lowest possible AID that can be assigned to any STA in this example is 8. There are no group addressed frames buffered in the AP for any of the eight BSSIDs. However, STAs with AID 9 and AID 11 have an individually addressed frames buffered in the AP. Figure O-4 shows the values of the Bitmap Control and Partial Virtual Bitmap fields that would be part of the TIM element for this example when either Method A or Method B is used. It is noted that Method B reduces to Method A in this example.

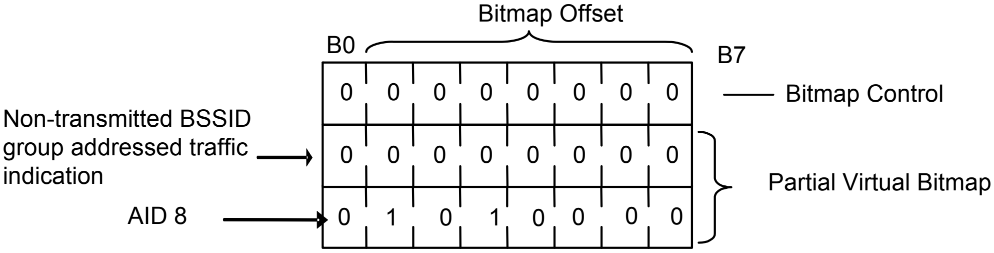


Figure O-4—Virtual Bitmap Example #4, Method A and Method B

In the next example, there are eight BSSIDs and the lowest possible AID that can be assigned to any STA in this example is 8. There are group addressed frames buffered at the AP for the transmitted BSSID, and the DTIM Count field in the TIM element of the transmitted BSSID is 0. The nontransmitted BSSID with BSSID Index 3 also has the DTIM Count field set to 0 and has buffered group addressed frames. All other nontransmitted BSSIDs have no buffered group addressed frames. In addition, STAs with AID 12, AID 17, AID 22 and AID 24 have data buffered at the AP. Figure O-5 shows the values of the Bitmap Control and

Partial Virtual Bitmap fields that would be part of the TIM element for this example when either Method A or Method B is used. It is noted that Method B reduces to Method A in this example.

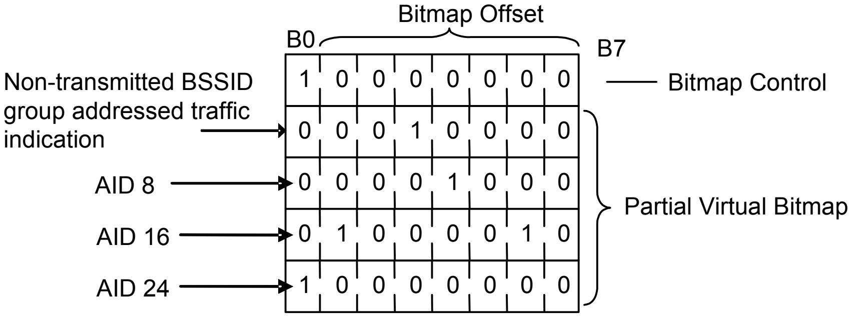


Figure O-5—Virtual Bitmap Example #5, Method A or Method B

In the third example, there are sixteen BSSIDs and the lowest possible AID that can be assigned to any STA is 16. There are no group addressed frames buffered at the AP for the transmitted BSSID, and the DTIM Count field in the TIM element of the transmitted BSSID is 0. The nontransmitted BSSID Index 3 also has the DTIM Count field set to 0 and has group addressed frames buffered at the AP. All other nontransmitted BSSIDs have no buffered group addressed frames. In addition, the STA with AID 39 has individually addressed frames buffered at the AP. Figure O-6 and Figure O-7 show the values of the Bitmap Control and Partial Virtual Bitmap fields that would be part of the TIM element for this example when Method A and Method B are used, respectively.

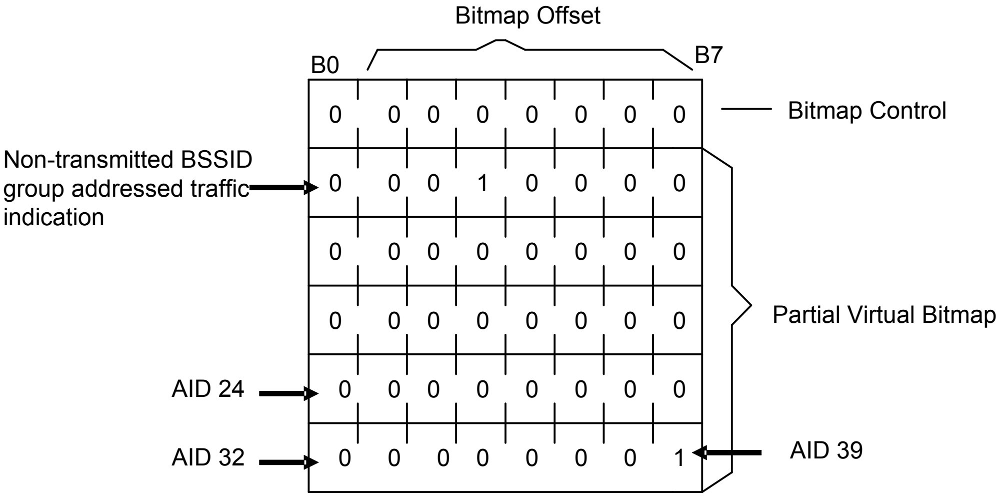


Figure O-6—Virtual Bitmap Example #5, Method A

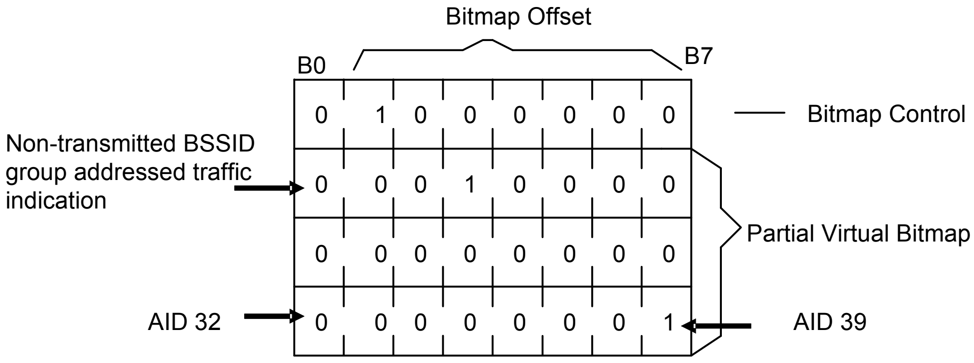


Figure O-7—Virtual Example #5, Method B

O.3 Sample C code

The following C source code illustrates how to construct the TIM Virtual Bit Map. Because this is an illustration, no efficiency or appropriateness for actual implementation is implied.

```
#include <stdio.h>
#include <limits.h>

#define ADD_TIM_BIT     0
#define REMOVE_TIM_BIT  1

#define TIM_ELEMENT_ID  5
#define TIM_BASE_SIZE   3       /* size of TIM fixed fields */

#define AID_SIZE        2008    /* valid AIDs are numBssids thru 2007 */
#define VBM_SIZE        251     /* size of VBM array = 2008/8 = 251 */
#define MAX_BSSIDS      128     /* maximum possible number of BSSIDs per AP, is a power of 2 */

typedef unsigned char UINT8;
typedef unsigned short int UINT16;

struct _tim
   {
   UINT8 Element_id;
   UINT8 IELength;
   UINT8 DtimCount;
   UINT8 DtimPeriod;
   UINT8 BitMapControl;
   UINT8 PartialVirtualBitMap [VBM_SIZE];
   };

UINT8 virtualBitMap [VBM_SIZE];

UINT8 mcast_pending [MAX_BSSIDS] = {0};
UINT8 dtimCount [MAX_BSSIDS] = {0};
UINT8 dtimPeriod [MAX_BSSIDS] = {5};
```

```
void
Build_TIM (struct _tim *Tim, char TIM_method, UINT16 numBssids)
   {
   UINT8 octetIndex = 0;
   UINT8 octetIndex0 = 0;
   UINT8 octetIndex1 = 0;
   UINT8 offset = 0;
   UINT8 lengthOfPartialVirtualBitMap = 0;
   UINT8 bcast_octet = 0;
   UINT8 bcast_bit = 0;
   UINT8 max_bcast_octetIndex = 0;
   UINT16 bssidIndex = 0;
   UINT16 N2 = 0;

   /* Compute the largest octet_index for bcast indication */
   max_bcast_octetIndex = (numBssids - 1) / 8;

   /* Initialize PartialVirtualBitMap */
   for (octetIndex = 0;  octetIndex < VBM_SIZE;  octetIndex++)
      Tim->PartialVirtualBitMap [octetIndex] = 0;
   octetIndex = 0;
   if (numBssids == 1)
      {
      /* Find the first nonzero octet in the virtual bit map */
      for (octetIndex = 0;  ((virtualBitMap [octetIndex] == 0) && (octetIndex < VBM_SIZE));
          octetIndex++)
         /* empty */ ;
      if (octetIndex < VBM_SIZE)
         offset = octetIndex & 0xFE;

      /* Find the last nonzero octet in the virtual bit map */
      for (octetIndex = (VBM_SIZE - 1);  ((virtualBitMap [octetIndex] == 0) && (octetIndex > 0));
          octetIndex--)
         /* empty */ ;
      lengthOfPartialVirtualBitMap = octetIndex - offset + 1;
      Tim->IELength = lengthOfPartialVirtualBitMap + TIM_BASE_SIZE;
      Tim->BitMapControl = offset;

      /* Copy the virtual bit map octets that are nonzero */
      /* Note: A NULL virtualBitMap will still add a single octet of 0 */
      for (octetIndex = 0;  octetIndex < lengthOfPartialVirtualBitMap;  octetIndex++)
         Tim->PartialVirtualBitMap [octetIndex] = virtualBitMap [offset + octetIndex];
      }
   if (numBssids > 1)
      {
      /* Update the broadcast/multicast bits, when numBssids > 1 */
      for (bssidIndex = 1;  bssidIndex < numBssids;  bssidIndex++)
         {
         bcast_octet = (UINT8) (bssidIndex >> 3);
         bcast_bit = (UINT8) (0x01 << (bssidIndex & 0x07));
         if (mcast_pending [bssidIndex])
            virtualBitMap [bcast_octet] |= bcast_bit;
         else
            virtualBitMap [bcast_octet] &= ~bcast_bit;
```

```
    }
for (octetIndex0 = 0;
     (octetIndex0 < VBM_SIZE) && (virtualBitMap [octetIndex0] == 0);
     octetIndex0++)
    /* empty */ ;

/* PVB contains neither bcast nor buffered ucast traffic, PVB is a single all-0 byte */
if (octetIndex0 == VBM_SIZE)
    {
    lengthOfPartialVirtualBitMap = 1;
    offset = 0;
    Tim->IELength = lengthOfPartialVirtualBitMap + TIM_BASE_SIZE;
    Tim->BitMapControl = offset;
    Tim->PartialVirtualBitMap [0] = virtualBitMap [0];
    }
for (octetIndex1 = (max_bcast_octetIndex + 1);
     ((octetIndex1 < VBM_SIZE) && (virtualBitMap [octetIndex1] == 0));
     octetIndex1++)
    /* empty */ ;

/* PVB only contains bcast indication, no buffered ucast traffic */
if ((octetIndex1 == VBM_SIZE) && (octetIndex0 < VBM_SIZE))
    {
    lengthOfPartialVirtualBitMap = max_bcast_octetIndex + 1;
    offset = 0;
    Tim->IELength = lengthOfPartialVirtualBitMap + TIM_BASE_SIZE;
    Tim->BitMapControl = offset;
    for (octetIndex = 0;  octetIndex < (max_bcast_octetIndex + 1);  octetIndex++)
        /* empty */ ;
    Tim->PartialVirtualBitMap [octetIndex] = virtualBitMap [octetIndex];
    }

/* PVB contains ucast indication with or without buffered bcast traffic */
for (octetIndex = 0;  octetIndex < (max_bcast_octetIndex + 1);  octetIndex++)
    Tim->PartialVirtualBitMap [octetIndex] = virtualBitMap [octetIndex];
if ((octetIndex1 < VBM_SIZE) && (octetIndex0 < VBM_SIZE))
    {
    if (TIM_method == 'A')
        {
        offset = 0;
        for (octetIndex = (VBM_SIZE - 1);
             (virtualBitMap [octetIndex] == 0) && (octetIndex > (max_bcast_octetIndex + 1));
             octetIndex--)
            /* empty */ ;
        N2 = octetIndex;
        lengthOfPartialVirtualBitMap = N2 - offset + 1;
        for (octetIndex = (max_bcast_octetIndex + 1);  (octetIndex <= N2);  octetIndex++)
            Tim->PartialVirtualBitMap [octetIndex] = virtualBitMap [octetIndex];
        Tim->IELength = lengthOfPartialVirtualBitMap + TIM_BASE_SIZE;
        Tim->BitMapControl = offset;
        }
    if (TIM_method == 'B')
        {
        offset = octetIndex1 - (max_bcast_octetIndex + 1);
```

```
            offset = offset & 0xFE;
            /* The result of (max_bcast_octetIndex + 1) + offset is equal to N1 that is described in 8.4.2.7 for
the TIM element. */
            for (octetIndex = (VBM_SIZE - 1);
               ((virtualBitMap [octetIndex] == 0) && (octetIndex > (max_bcast_octetIndex + 1)));
               octetIndex--)
              /* empty */ ;
            N2 = octetIndex;
            lengthOfPartialVirtualBitMap = N2 - offset + 1;
            for (octetIndex = (max_bcast_octetIndex + 1);
               octetIndex <= (lengthOfPartialVirtualBitMap - 1);
               octetIndex++)
              Tim->PartialVirtualBitMap [octetIndex] = virtualBitMap [offset + octetIndex];
            Tim->IELength = lengthOfPartialVirtualBitMap + TIM_BASE_SIZE;
            Tim->BitMapControl = offset;
            }
         }
      }
   Tim->Element_id = TIM_ELEMENT_ID;
   Tim->DtimCount = dtimCount [0];
   Tim->DtimPeriod = dtimPeriod [0];

   /* Update broadcast/ multicast indication bit for transmitted BSSID if necessary */
   if ((Tim->DtimCount == 0) && mcast_pending [0])
      Tim->BitMapControl |= 0x01;
   }

void
Update_VirtualBitMap (UINT16 station_id, UINT8 Action)
   {
   UINT16 aid = station_id;
   UINT8 aid_octet;
   UINT8 aid_bit;

   if ((aid > 0) && (aid < AID_SIZE))
      {
      /* Get aid position in Virtual Bit Map. */
      aid_octet = (UINT8) (aid >> 3);
      aid_bit = (UINT8) (0x01 << (aid & 0x07));
      if (Action == REMOVE_TIM_BIT)
         virtualBitMap [aid_octet] &= ~aid_bit;
      else
         virtualBitMap [aid_octet] |= aid_bit;
      }
   }

int
main (void)
   {
   struct _tim Tim;
   UINT8 ExampleCase;
   UINT16 count = 0;
   char TIM_method;
   UINT16 numBssids = 1;
```

```
/* The value of ExampleCase depends on the test case, allowed values are 1 to 22 */
ExampleCase = 1;

/* The value of TIM_method depends on the method to use, allowed values are 'A' and 'B' */
TIM_method = 'A';

switch (ExampleCase)
   {
   /* Nine examples with numBssids = 1, no difference between Method A and B */
   case 1:
      mcast_pending [0] = 0;
      Update_VirtualBitMap (2, ADD_TIM_BIT);
      Update_VirtualBitMap (7, ADD_TIM_BIT);
      break;
   case 2:
      mcast_pending [0] = 1;
      Update_VirtualBitMap (2, ADD_TIM_BIT);
      Update_VirtualBitMap (7, ADD_TIM_BIT);
      Update_VirtualBitMap (22, ADD_TIM_BIT);
      Update_VirtualBitMap (24, ADD_TIM_BIT);
      break;
   case 3:
      mcast_pending [0] = 1;
      Update_VirtualBitMap (24, ADD_TIM_BIT);
      break;
   case 4:
      mcast_pending [0] = 0;
      Update_VirtualBitMap (3, ADD_TIM_BIT);
      Update_VirtualBitMap (37, ADD_TIM_BIT);
      Update_VirtualBitMap (43, ADD_TIM_BIT);
      break;
   case 5:
      mcast_pending [0] = 0;
      Update_VirtualBitMap (35, ADD_TIM_BIT);
      break;
   case 6:
      mcast_pending [0] = 0;
      Update_VirtualBitMap (43, ADD_TIM_BIT);
      break;
   case 7:
      mcast_pending [0] = 0;
      Update_VirtualBitMap (35, ADD_TIM_BIT);
      Update_VirtualBitMap (35, REMOVE_TIM_BIT);
      break;
   case 8:
      mcast_pending [0] = 1;
      Update_VirtualBitMap (13, ADD_TIM_BIT);
      Update_VirtualBitMap (43, ADD_TIM_BIT);
      Update_VirtualBitMap (63, ADD_TIM_BIT);
      Update_VirtualBitMap (73, ADD_TIM_BIT);
      break;
   case 9:
      mcast_pending [0] = 1;
```

```
        Update_VirtualBitMap (2007, ADD_TIM_BIT);
        break;


    /* Thirteen examples with numBssids > 1, TIM_method = 'A' or 'B' */
    case 10:
        numBssids = 8;
        Update_VirtualBitMap (9, ADD_TIM_BIT);
        Update_VirtualBitMap (11, ADD_TIM_BIT);
        break;
    case 11:
        numBssids = 8;
        mcast_pending [0] = 1;
        mcast_pending [3] = 1;
        Update_VirtualBitMap (12, ADD_TIM_BIT);
        Update_VirtualBitMap (17, ADD_TIM_BIT);
        Update_VirtualBitMap (22, ADD_TIM_BIT);
        Update_VirtualBitMap (24, ADD_TIM_BIT);
        break;
    case 12:
        numBssids = 16;
        mcast_pending [3] = 1;
        Update_VirtualBitMap (39, ADD_TIM_BIT);
        break;
    case 13:
        numBssids = 8;
        mcast_pending [5] = 1;
        mcast_pending [7] = 1;
        Update_VirtualBitMap (23, ADD_TIM_BIT);
        break;
    case 14:
        numBssids = 8;
        mcast_pending [2] = 1;
        mcast_pending [7] = 1;
        Update_VirtualBitMap (10, ADD_TIM_BIT);
        break;
    case 15:
        numBssids = 15;
        mcast_pending [5] = 1;
        mcast_pending [7] = 1;
        Update_VirtualBitMap (2007, ADD_TIM_BIT);
        break;
    case 16:
        numBssids = 15;
        mcast_pending [5] = 1;
        mcast_pending [7] = 1;
        Update_VirtualBitMap (1997, ADD_TIM_BIT);
        Update_VirtualBitMap (1999, ADD_TIM_BIT);
        break;
    case 17:
        numBssids = 32;
        for (count = 0;  count < numBssids;  count += 2)
            mcast_pending [count] = 1;
        Update_VirtualBitMap (32, ADD_TIM_BIT);
```

```
            Update_VirtualBitMap (33, ADD_TIM_BIT);
            Update_VirtualBitMap (39, ADD_TIM_BIT);
            break;
        case 18:
            numBssids = 14;
            mcast_pending [5] = 1;
            mcast_pending [7] = 1;
            break;
        case 19:
            numBssids = 14;
            mcast_pending [0] = 1;
            mcast_pending [12] = 1;
            break;
        case 20:
            numBssids = 15;
            Update_VirtualBitMap (38, ADD_TIM_BIT);
            break;
        case 21:
            numBssids = 15;
            mcast_pending [0] = 1;
            Update_VirtualBitMap (44, ADD_TIM_BIT);
            break;
        case 22:
            numBssids = 16;
            break;
        default:
            break;
        }
    Build_TIM (&Tim, TIM_method, numBssids);

    printf ("\nCase = %d, method %c.\n", ExampleCase, TIM_method);
    printf ("numBssids = %d.\n", numBssids);
    printf ("Element_id = %d.\n", Tim.Element_id);
    printf ("IELength = %d.\n", Tim.IELength);
    printf ("DtimCount = %d.\n", Tim.DtimCount);
    printf ("DtimPeriod = %d.\n", Tim.DtimPeriod);
    printf ("BitMapControl = 0x%02X\n", Tim.BitMapControl);
    if (Tim.IELength - TIM_BASE_SIZE > 0)
        {
        int octetIndex;

        for (octetIndex = 0;  octetIndex < Tim.IELength - TIM_BASE_SIZE;  octetIndex++)
            printf ("PartialVirtualBitMap [%d] = 0x%02X\n", octetIndex,
                    Tim.PartialVirtualBitMap [octetIndex]);
        }
    return 0;
    }

/* The End. */
```

Annex P

(informative)

Integration function

P.1 Introduction

The purpose of this annex is to guide the implementor of a WLAN system that includes a portal that integrates the WLAN system with a wired LAN.

P.2 Ethernet V2.0/IEEE 802.3 LAN integration function

It is recommended that any WLAN system that logically incorporates a portal that integrates the WLAN system with an Ethernet V2.0/IEEE 802.3 LAN use the procedures defined in ISO/IEC Technical Report 11802-5:1997(E) (previously known as IEEE Std 802.1H-1997), with the two-entry selective translation table (STT) shown in Table P-1, to perform the integration service. Note that the majority of such IEEE 802.11 implementations currently use this STT, rather than the single-entry STT recommended in Annex A of IEEE Std 802.1H-1997 [B21].

Table P-1—IEEE 802.11 integration service STT

Protocol use	Ethernet type value	Encoding in Type field	
		First octet	Second octet
AppleTalk Address Resolution Protocol	0x80F3	80	F3
Novell NetWare Internetwork Packet exchange (IPX)	0x8137	81	37

P.3 Example

In order to illustrate the translations performed by the integration service using the encapsulation/decapsulation procedures defined in ISO/IEC Technical Report 11802-5:1997(E) with the IEEE 802.11 integration service STT, the following tables show how the octets in an IEEE 802.11 MSDU correspond to the octets in the Ethernet/IEEE 802.3 MSDU that represents the same LLC SDU on the integrated Ethernet/IEEE 802.3 LAN. Table P-2 shows the encapsulation example, and Table P-3 shows the decapsulation example.

Note that examples in both tables showing a Type/Length field value of 81-00 represents bridging between an Ethernet/IEEE 802.3 LAN and an IEEE 802.11 LAN, both of which are carrying VLAN-tagged MSDUs (User Priority=4, CFI-0, VLAN ID=1893).

Table P-2—Ethernet/IEEE 802.3 to IEEE 802.11 translation

Protocol	Type / Length	LLC header	IEEE 802.11 LLC header
IP	08-00	—	AA-AA-03-00-00-00-08-00
IP 802.3[a]	length	AA-AA-03-00-00-00-08-00	AA-AA-03-00-00-00-08-00
IP ARP	08-06	—	AA-AA-03-00-00-00-08-06
AppleTalk (1)	80-9B	—	AA-AA-03-00-00-00-80-9B
AppleTalk (2)	length	AA-AA-03-08-00-07-80-9B	AA-AA-03-08-00-07-80-9B
AppleTalk AARP (1)	80-F3	—	AA-AA-03-00-00-F8-80-F3
AppleTalk AARP (2)	length	AA-AA-03-00-00-00-80-F3	AA-AA-03-00-00-00-80-F3
IPX Ethernet II	81-37	—	AA-AA-03-00-00-F8-81-37
IPX SNAP	length	AA-AA-03-00-00-00-81-37	AA-AA-03-00-00-00-81-37
IPX 802.2	length	E0-E0-03	E0-E0-03
IPX 802.3[b]	length	FF-FF	FF-FF
VLAN-tagged IP	81-00	87-65-08-00	AA-AA-03-00-00-00-81-00-87-65-AA-AA-03-00-00-00-08-00[c]

[a] This format of IP packet over IEEE Std 802.3 is denigrated, and the change to the canonical Ethernet IP format is not considered harmful.

[b] The use of this nonstandard format happens to work with these rules, even though the FF-FF is not actually a valid LLC header value. (The broadcast LSAP is not valid as a source SAP in LLC. See IEEE Std 802.2.)

[c] The sequence of octets AA-AA-03-00-00-00-81-00-87-65 represents the SNAP-encoded VLAN header. The sequence of octets AA-AA-03-00-00-00-08-00 represents the IEEE 802.1H-translated Type/Length field, using the same translation as the untagged IP MSDU on the first line of Table P-2.

Table P-3—IEEE 802.11 to Ethernet/IEEE 802.3 translation

Protocol	IEEE 802.11 LLC header	Type / Length	LLC header
IP	AA-AA-03-00-00-00-08-00	08-00	—
IP 802.3[a]	AA-AA-03-00-00-00-08-00	length	—
IP ARP	AA-AA-03-00-00-00-08-06	08-06	—
AppleTalk (1)	AA-AA-03-00-00-00-80-9B	80-9B	—
AppleTalk (2)	AA-AA-03-08-00-07-80-9B	length	AA-AA-03-08-00-07-80-9B
AppleTalk AARP (1)	AA-AA-03-00-00-F8-80-F3	80-F3	—
AppleTalk AARP (2)	AA-AA-03-00-00-00-80-F3	length	AA-AA-03-00-00-00-80-F3
IPX Ethernet II	AA-AA-03-00-00-F8-81-37	81-37	—
IPX SNAP	AA-AA-03-00-00-00-81-37	length	AA-AA-03-00-00-00-81-37
IPX 802.2	E0-E0-03	length	E0-E0-03

Table P-3—IEEE 802.11 to Ethernet/IEEE 802.3 translation *(continued)*

Protocol	IEEE 802.11 LLC header	Type / Length	LLC header
IPX 802.3	FF-FF	length	FF-FF
VLAN-tagged IP ARP	AA-AA-03-00-00-00-81-00-87-65-AA-AA-03-00-00-00-08-06	81-00	87-65-08-06

[a] This format of IP packet does not survive the trip across the non-IEEE-802.3 LAN intact.

P.4 Integration service versus bridging

There are a number of differences between the IEEE 802.11 integration service and the service provided by an IEEE 802.1 bridge. In the IEEE 802.11 architecture, a portal provides the minimum connectivity between an IEEE 802.11 WLAN system and a non-IEEE-802.11 LAN. Requiring an IEEE 802.1D bridge in order to be compliant with IEEE Std 802.11 would unnecessarily render some implementations noncompliant.

The most important distinction is that a portal has only one “port” (in the sense of IEEE Std 802.1D, for example) through which it accesses the DS. This renders it unnecessary to update bridging tables inside a portal each time a STA changes its association status. In other words, the details of distributing MSDUs inside the IEEE 802.11 WLAN need not be exposed to the portal.

Another difference is that the DS is not an IEEE 802 LAN (although it carries IEEE 802 LLC SDUs). Requiring that the DS implements all behaviors of an IEEE 802 LAN places an undue burden on the architecture.

Finally, it is an explicit intent of this standard to permit transparent integration of an IEEE 802.11 WLAN into another non-IEEE-802.11 LAN, including passing bridge PDUs through a portal. While an implementer may wish to attach an IEEE 802.1D bridge to the portal (note that the non-IEEE-802.11 LAN interface on the bridge need not be any particular type of LAN), it is not an architectural requirement of this standard to do so.

Annex Q

(informative)

AP functional description

Q.1 Introduction

This informative annex seeks to clarify the AP functional description. At times there is some confusion surrounding the term *AP* and the relation of that term to the AP functions and common implementations of AP devices. The core IEEE 802.11 conceptual definitions that surround the AP (refer to Clause 4) are abstract (and can sometimes cause confusion), but Clause 4 definitions are crafted to be flexible and hence serve to allow the adaptation and extension of this standard in a wide variety of ways.

Q.2 Terminology

An enhanced description of these access entities begins with clarification of several terms.

This standard defines an entity called a STA. STAs can operate in different modes. The possible operational modes of a STA are

a) Infrastructure mobile STAs
b) Ad hoc mobile STAs
c) Access control mode STAs
d) Mesh STAs

The mobile STAs are the STA entities that are ordinarily moving around, but may also be in a fixed location. The mobile adjective prefix often helps in visualizing the type of STA under discussion.

Infrastructure mobile STAs operate in infrastructure BSS mode, i.e., they are the users of an AP. Devices that incorporate an infrastructure mobile STA are referred to in this annex by the term *mobile unit* (MU). An MU device may consist of just a mobile STA implementation, but also likely includes an SME and a client. The exact configuration of the MU is not relevant to the descriptions in this annex.

Ad hoc mobile STAs operate in IBSS mode. Ad hoc mobile STAs form autonomous networks that do not require an AP.

A STA can also form an integral part of an AP. To do so, the STA must operate in access control mode. This type of STA is called an *access control mode STA* (ACM_STA).

The primary function of an AP is to provide the MUs with access to the DS, as shown in the Unified Modeling Language (UML) use case diagram in Figure Q-1. Complete UML specifications can be found in Rumbaugh [B54]. The UML diagram shows a system boundary box containing a single use case (ellipse). Entities that are outside the system boundary box are shown as stick people. These external entities represent the actors (formal term), or users, of the system. Since the actors are outside the system boundary, their internal behavior is not described (i.e., it is out of scope for the current view). Instead, references to the external entities are limited to descriptions of their interactions with, and expectations of, the system, which is accomplished by describing the use cases and scenarios (i.e., functions) of the system and later (in Figure Q-4) a decomposition of the entities (objects and behaviors) within the system that provide that functionality. The use case diagram employs connecting lines to indicate relationships between the various artifacts present in the diagram. Relationship lines lacking arrows on both ends indicate that there is a bidirectional relationship between the artifacts.

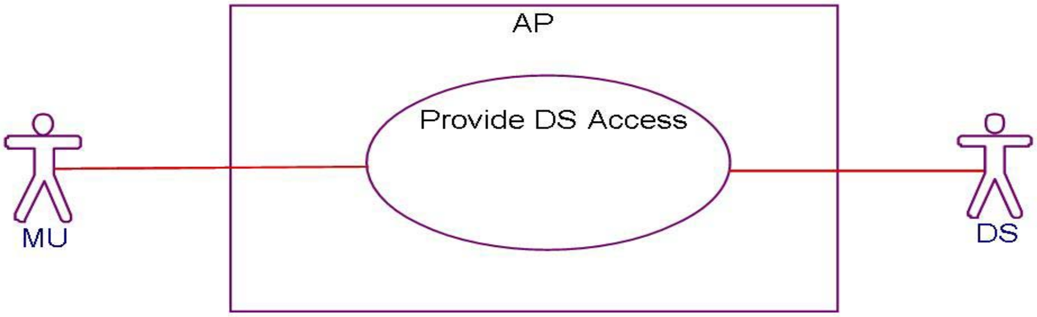


Figure Q-1—Very high level UML use case diagram for the AP

The DS enables communication between MUs and the construction of collections of APs. To enable communication between MUs and a non-IEEE-802.11-LAN entity requires the presence of a (logical) portal from the DS to the non-IEEE-802.11 LAN.

Often the functions of an AP, which includes an ACM_STA, a DS, and a portal, are combined into a single device, referred to in this annex as an *access unit* (AU). While reference to that basic implementation is commonplace, it is helpful to discuss the abstract case: a WLAN system. The WLAN system includes the DS, AP, the AP's STA, and portal entities. It is also the logical location of distribution and integration service functions of an ESS. An infrastructure WLAN system contains one or more APs and zero or more portals in addition to the DS.

The primary function of a WLAN system is to provide the MUs with access to the non-IEEE-802.11 LAN, as shown in Figure Q-2. A secondary function is to provide the MUs with access to each other.

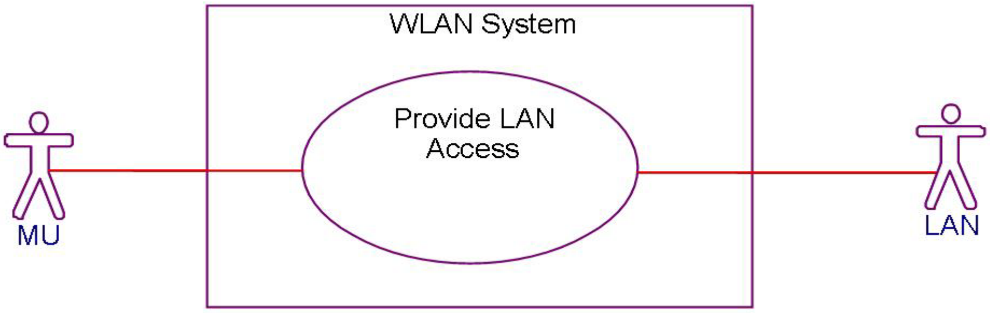


Figure Q-2—Very high level UML use case diagram for the WLAN system

The primary functions of the WLAN system can be further characterized as follows:

a) Provide non-IEEE-802.11-LAN access.
   1) Includes MU validation.
   2) Includes moving data between the MUs and the non-IEEE-802.11 LAN.
      i) Uses a special data movement function called *filtering data*.

b) Configure the system.

Those high-level use cases of the WLAN system are shown in the UML use case diagram in Figure Q-3. The UML diagram shows multiple use cases with two types of relationships between those use cases: include and generalization. The include relationship “includes” functionality in one use case that is described by another use case. The “Provide LAN Access” use case includes the functionality of the “Validate MU” use case. The generalization relationship indicates that one use case is a more general form of another use case. The relationship line with a hollow triangle at the end indicates that the “Move Data” use case is a more generalized form of the “Filter Data” use case. Or, equivalently, “Filter Data” is a more specialized form of “Move Data.” The constraint artifact (in the lower left corner of the diagram) requires the WLAN system and the MUs to be set to the same SSID.

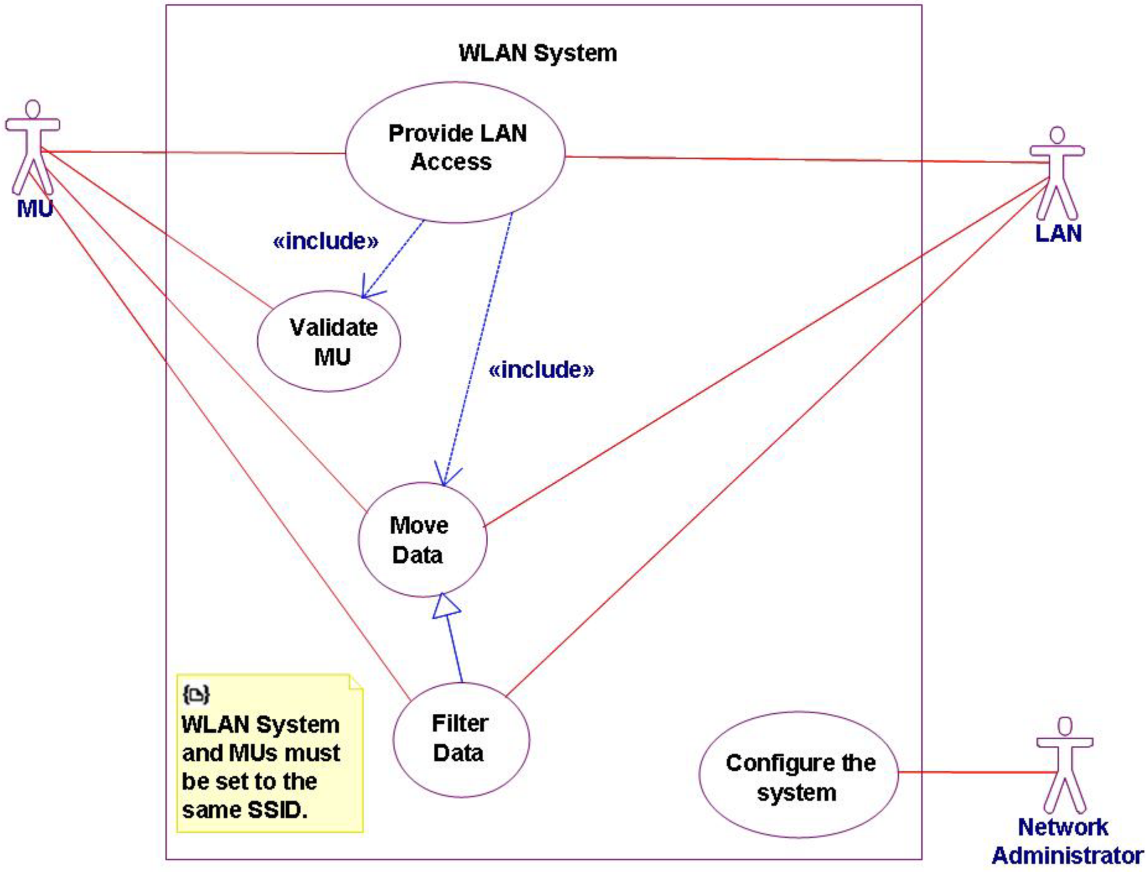


Figure Q-3—High-level UML use case diagram for the WLAN system

The primary functions of the WLAN system, depicted in the UML object model diagram in Figure Q-4, are provided by the ACM_STA, AP, and DS entities, the latter via the DSSs. The portal is merely a conceptual link from the DS to the non-IEEE-802.11 LAN. The UML object model diagram shows a package containing a set of object classes. References to the actors (external entities) are limited to descriptions of their interactions with, and expectations of, the object classes, which is accomplished by describing the attributes and behaviors of the class entities. The object model diagram uses connecting lines to indicate interfaces (called *associations* in UML terminology) between the various artifacts present in the diagram. UML association lines lacking arrows on both ends indicate that there is a bidirectional interface between the artifacts. The UML association lines are annotated with multiplicity counts to indicate the how many entities may fill the position at that end of the association.

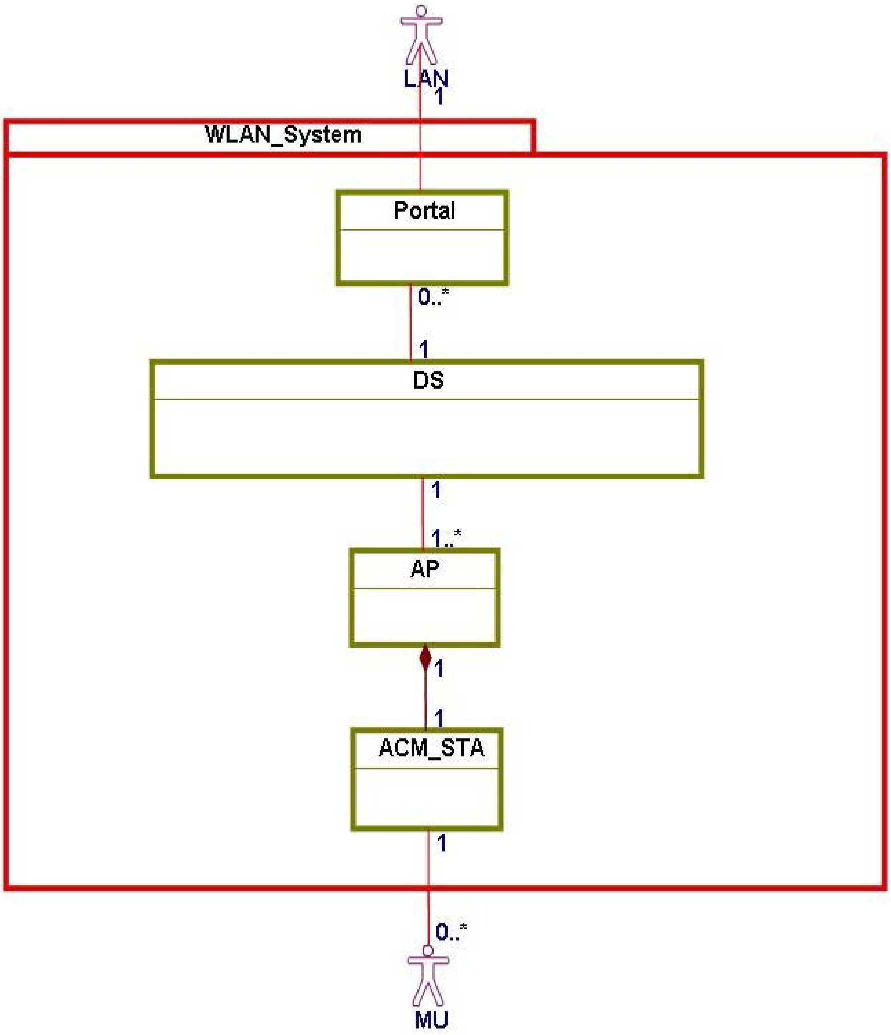


Figure Q-4—High-level UML entity diagram for the WLAN system

Figure Q-4 also shows the relationships between the WLAN system entities. There exists a bidirectional association between zero or more [0..*] MUs and a single (given) ACM_STA. The solid diamond terminated line indicates that there is a composition relationship between the ACM_STA and the AP, i.e., the AP is composed of (or always has an) ACM_STA. Hence there is a one-to-one mapping between APs and ACM_STAs. This composition and one-to-one relationship can also be drawn as shown in Figure Q-5. These two forms are equivalent. There are one or more [1..*] APs connected to a single DS. The DS in turn connects to zero or more [0..*] portals. Each portal connects to a single non-IEEE-802.11 LAN.

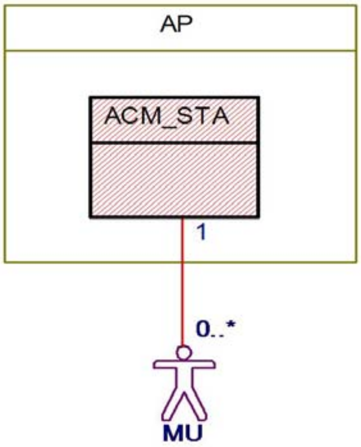


Figure Q-5—AP UML composition diagram (alternate syntax)

Q.3 Primary ACM_STA functions

The primary functions of an ACM_STA are as follows:

a) Instantiate the infrastructure BSS.

b) Move data between the MUs and the AP.

Instantiating the infrastructure BSS consists of advertising the BSS and defining timing for the entire BSS (i.e., infrastructure mode TSF). Advertising the BSS includes creating an infrastructure mode Beacon frame, transmitting that Beacon frame, and replying to MU probe requests with corresponding probe response transmissions. Beacon frames and probe responses provide a way for the MUs to find, join with the ACM_STA, and (subsequently) associate with the AP. This includes, for example, the AP providing channel, regulatory, and country information to the MUs.

Moving data between the MUs and the AP consists of translating between MSDUs and MPDUs, buffering data, and transmitting/receiving MPDUs via an IEEE 802.11 PHY.

Q.4 Primary AP functions

The primary functions of an AP are as follows:

a) Provide DS access for the MUs.
   1) Includes MU validation and extends (in some cases) to notifying the DS.
   2) Includes moving data between the MUs and the DS.
      i) Uses a special data movement function called *data filtering*.

b) Configure the AP (both the AP itself and the included ACM_STA).

Those high-level use cases of the AP are shown in the UML use case diagram in Figure Q-6. The UML extend relationship “extends” a use case with optional functions provided by another use case that are applied only in some scenarios. In Figure Q-6, the “Notify the DS” use case extends the “Validate MU” use case in some scenarios.

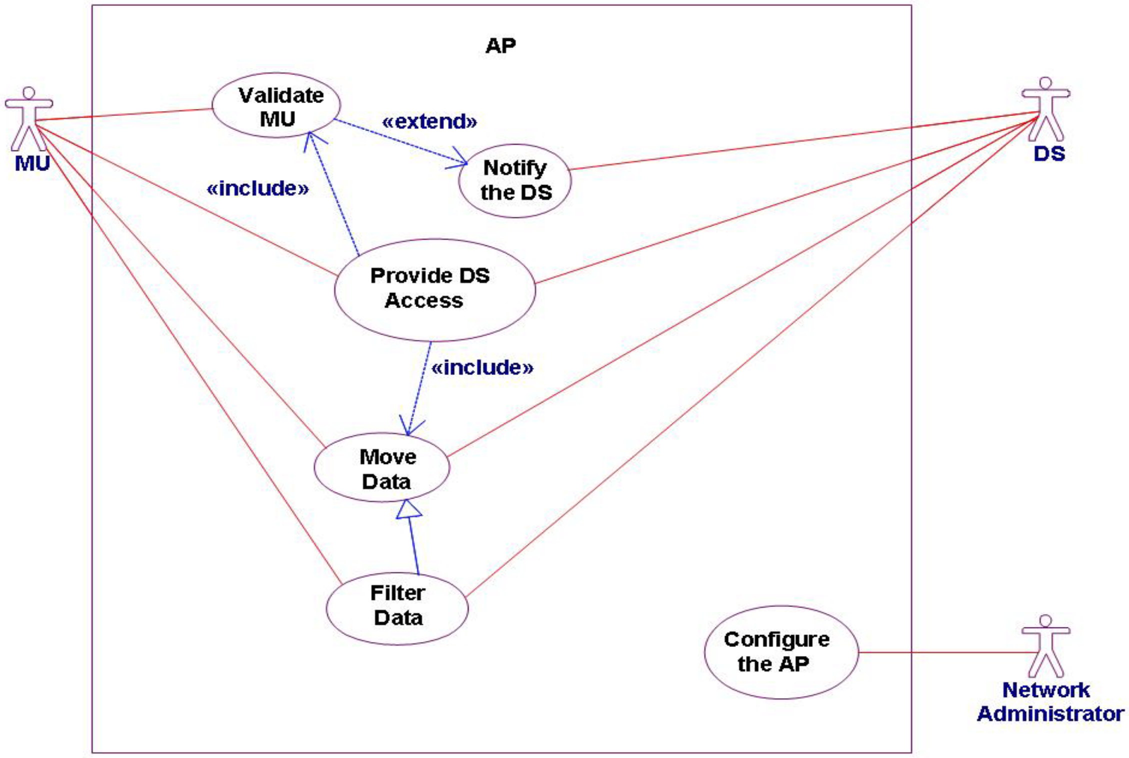


Figure Q-6—High-level UML use case diagram for the AP

The AP provides DS access for the MUs, including validating the MUs (e.g., via STA and/or client authentication) and providing access and admission control (e.g., via the association process). An AP is literally a point of access to the DS (and, by extension, to the non-IEEE-802.11 LAN beyond). Upon validating an MU, the AP updates the DS mapping of the MU to the AP. DS mapping updates can, for example, be based on association and reassociation requests (received from the ACM_STA) or aging of inactive links (based on session timers). An AP may also receive access control updates directly from other APs, via a protocol outside the scope of this standard, in the form of inter-AP notifications of MU association events and transitions. In this way, MU validation and subsequent changes in MU access control lead to adjustments in how data are allowed to move through the AP (i.e., between the MUs and the DS).

Providing DS access for the MUs also includes moving data between the MUs and the DS, which is accomplished by moving MSDUs between the ACM_STA and the DS (bidirectionally).

The moving data function includes “filtering data.” The filtering data function controls which MSDUs (if any) are moved between the DS and the MUs. For example, filtering of data to and from a particular MU is adjusted based on the various stages of validation of that MU.

The AP also provides a function for configuration. Configuring the AP includes configuring the AP itself as well as the included ACM_STA. For example, configuration may include setting the local values of the SSID, PHY channel, beacon interval, and so on.

Q.5 Primary DS functions

The primary functions of the DS (i.e., the DSSs) are as follows:

a) Map MU to AP (for DS-to-MU traffic delivery).

b) Move data.

The DS's MU-to-AP map determines which AP is to be used for a given MU's data delivery. This function includes mapping update adjustments, which are based on notification from the APs, of changes in MU access control.

Moving data consists of moving MSDUs among the APs (including returning MSDUs to the source AP for MU-to-MU communications) and between the APs and the portal(s).

Q.6 Primary portal function

The primary function of a portal is as follows:

a) Move data with a special data movement function called *data transformation*.

Moving data consists of moving MSDUs between the DS and the external non-IEEE-802.11 LAN. Moving data through the portal transforms the MSDUs using the integration function. The integration function translates external non-IEEE-802.11-LAN MSDUs to and from IEEE 802.11 MSDUs using, for example, the procedures defined in Annex P.

Q.7 AU example

Now that the functions of a WLAN system have been described, here is the example of an AU implementation. Quite simply, an AU is an instantiation of a WLAN system as described in this annex.

Since transiting from a DS through a portal onto an integrated non-IEEE-802.11 LAN and then subsequently via another portal onto its DS is transparent, it is possible to define a DS in terms of not only the portals that are directly connected to a particular DS, but also in terms of all the integrated MAC endpoints. Therefore, a collection of AUs connected to an integrated non-IEEE-802.11 LAN can define a DS that consists of the union of all the DSs inside the AUs. The union of such a set of AUs would itself constitute a WLAN system.

Annex R

(informative)

DS SAP specification

R.1 Introduction

The purpose of this informative annex is to describe and clarify the DS SAP. The DS SAP is the interface between the DS SAP service users and the DS SAP service provider. The DS SAP service users are the connected APs, mesh gates, and portals. The DS SAP service provider is the DS. Figure R-1 shows the location of the DS SAP in the IEEE 802.11 architecture.

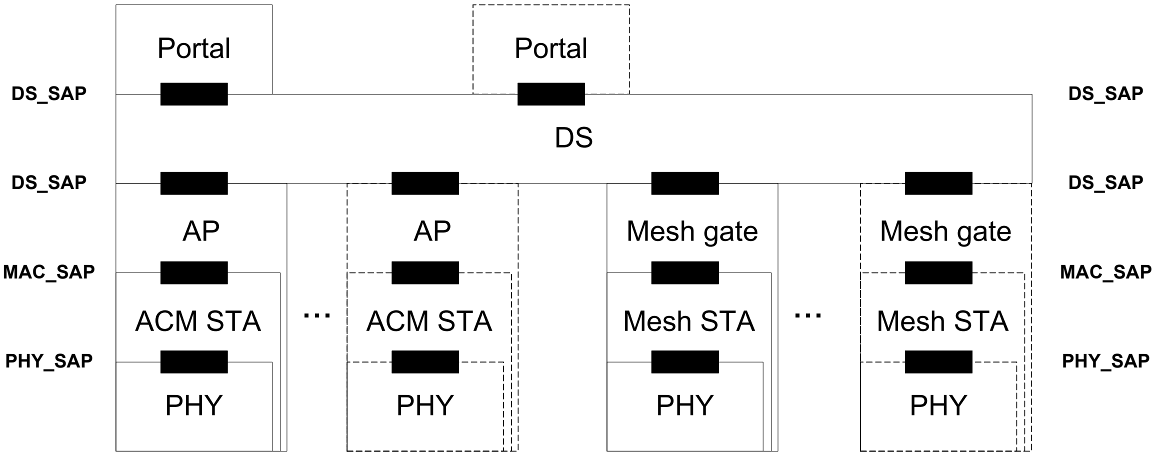


Figure R-1—Location of the DS SAP

The DS SAP interface specification describes the primitives required to get MSDUs in and out of the DS and update the DS's mapping of STAs to APs or to mesh gates. Describing the DS itself or the functions thereof is out of scope of this annex.

The DS SAP actions are as follows:

a) Accept MSDUs from APs, mesh gates, and portals.
b) Deliver MSDUs to APs, mesh gates, or portals.
c) Accept STA-to-AP mapping updates from the APs.
d) Accept STA-to-mesh gate mapping updates from the mesh gates.

When the DS delivers the MSDUs to an AP, the AP then determines when and how to deliver the MSDUs to the AP's MAC (via the MAC SAP). When the DS delivers the MSDUs to a mesh gate, the mesh gate then determines when and how to deliver the MSDUs to the mesh gate's MAC (via the MAC SAP).

R.2 SAP primitives

R.2.1 General

The DS SAP service interface primitives are as follows:

a) DS-UNITDATA.request

b) DS-UNITDATA.indication

c) DS-STA-NOTIFY.request

R.2.2 MSDU transfer

R.2.2.1 General

The DS-UNITDATA primitives accept and deliver IEEE 802.11 MSDUs, including all the parameters and data as defined in 5.2.2.2. These tuples are called *distribution system service data units* (DSSDUs).

R.2.2.2 DS-UNITDATA.request

R.2.2.2.1 Function

This primitive requests distribution of a DSSDU across the DS.

R.2.2.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
DS-UNITDATA.request(
                              Dssdu,
                              SourceType
                              )
```

Name	Type	Valid range	Description
Dssdu	IEEE 802.11 MSDU	Any valid data unit according to 5.2.2.2, including data and all parameters	Specifies the DSSDU to be distributed via the DS.
SourceType	Enumeration	SRC_AP, SRC_MESH_GATE, SRC_PORTAL	Specifies the type of entity that is requesting distribution of a DSSDU.

R.2.2.2.3 When generated

This primitive is generated by an AP, mesh gate, or portal to submit a DSSDU to the DS for distribution.

R.2.2.2.4 Effect of receipt

This primitive initiates distribution of the DSSDU through the DS. An individually addressed DSSDU from an AP, mesh gate, or portal is distributed through the DS to the corresponding AP, mesh gate, or portal. A group addressed DSSDU from an AP is distributed to all APs, mesh gates, and portals, including the originating entity. A group addressed DSSDU from a portal is distributed to all APs, mesh gates, and portals, except the originating portal.

R.2.2.3 DS-UNITDATA.indication

R.2.2.3.1 Function

This primitive indicates delivery of a DSSDU from the DS to an AP, mesh gate, or portal.

R.2.2.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
DS-UNITDATA.indication(
                                    Dssdu
                                    )
```

Name	Type	Valid range	Description
Dssdu	IEEE 802.11 MSDU	Any valid data unit according to 5.2.2.2, including data and all parameters	Specifies the DSSDU that is being delivered by the DS.

R.2.2.3.3 When generated

This primitive is generated by the DS to deliver a DSSDU to an AP, mesh gate, or portal.

R.2.2.3.4 Effect of receipt

This primitive delivers a DSSDU to an AP, mesh gate, or portal.

R.2.3 Mapping updates

R.2.3.1 General

The DS-STA-NOTIFY primitive is used to maintain the STA-to-AP and mesh STA-to-mesh gate mapping data of the DS.

R.2.3.2 DS-STA-NOTIFY.request

R.2.3.2.1 Function

This primitive requests an update to the DS’s STA-to-AP map or mesh STA-to-mesh gate map.

R.2.3.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
DS-STA-NOTIFY.request(
                                    STAAddress,
                                    UpdateType
                                    )
```

Name	Type	Valid range	Description
STAAddress	MACAddress	Any valid individual MAC address	When generated by an AP, specifies the address of the STA whose association status with the AP has changed. When generated by a mesh gate, specifies the address of the mesh STA whose reachability status through the mesh gate has changed.
UpdateType	Enumeration	ADD, MOVE, DELETE	Specifies the DS mapping update operation to be performed.

R.2.3.2.3 When generated

This primitive is generated by an AP or mesh gate to update the DS’s STA-to-AP map or mesh STA-to-mesh gate map.

R.2.3.2.4 Effect of receipt

When generated by an AP, this primitive updates the DS’s STA-to-AP map, which controls to which AP the DS delivers DSSDUs that are destined for a given STA.

When generated by a mesh gate, this primitive updates the DS's mesh STA-to-mesh gate map, which controls to which mesh gate the DS delivers DSSDUs that are destined for a given mesh STA. The mesh STA-to-mesh gate map can be incomplete.

There are many mechanisms to implement this mapping update for the cases of ADD and MOVE. One example mechanism, in the case where the DS is an IEEE 802 LAN, is to use an IEEE 802.2 XID null frame.

Annex S

(informative)

Additional HT information

S.1 Waveform generator tool

As an informative extension to this standard, the waveform generator tool has been written to model the PHY transmission process described in Clause 18, Clause 19, and Clause 20.

The waveform generator can be downloaded from the public IEEE 802.11 document website. The waveform generator code may be found in document 11-06/1715, and the waveform generator description may be found in document 11-06/1714.

The purpose of the tool is to promote common understanding of complex PHY algorithms, facilitate device interoperability by providing reference test vectors, and assist researchers in industry and academia to develop next generation wireless solutions.

The code is written in the MATLAB computing language and can be configured to generate test vectors for most PHY configurations, defined by this standard. Instructions on how to configure and run the Tool are specified in the documentation files that are supplied with the code. A command line interface and graphic user interface (GUI) exist to configure the tool. For consistency with this standard, the configuration interface is made very similar to the TXVECTOR parameters defined in 20.2.2.

The waveform generator tool produces test vectors for all transmitter blocks, defined in Figure 20-2 and Figure 20-3, generating reference samples in both frequency and time domains. Outputs of the tool are time domain samples for all transmitting chains.

S.2 A-MPDU deaggregation

This subclause contains a description of the deaggregation process. Other implementations are also possible.

The receiver checks the MPDU delimiter for validity based on the CRC. It can also check that the length indicated is within the value of the LENGTH parameter indicated in RXVECTOR.

If the MPDU delimiter is valid, the MPDU is extracted from the A-MPDU. The next MPDU delimiter is expected at the first multiple of 4 octets immediately after the current MPDU. This process is continued until the end of the PPDU is reached.

If the MPDU delimiter is not valid, the deaggregation process skips forward 4 octets and checks to see whether the new location contains a valid MPDU delimiter. It continues searching until either a valid delimiter is found or the end of the PSDU is reached based on the value of the LENGTH parameter indicated in the RXVECTOR.[57]

An A-MPDU parsing (deaggregation) algorithm is expressed (as a C programming language snippet) in Figure S-1.

[57] This procedure occasionally wrongly interprets a random bit-pattern as a valid delimiter. When this happens, the MAC attempts to interpret a random MPDU. The MAC discards it with a high probability based on a bad MAC CRC check.

```
void Parse_A_MPDU (int length)
{
  int offset = 0; /* Octet offset from start of PSDU */
  while (offset+4 < length)
  {
    if (valid_MPDU_delimiter(offset) &&
        get_MPDU_length(offset) <= (length -(offset+4)))
    { /* Valid delimiter */

      /* Receive the MPDU */
      Receive_MPDU(offset+4, get_MPDU_length(offset));

      /* advance by MPDU length rounded up to a multiple of 4 */
      offset += 4 + 4*((get_MPDU_length(offset)+3)/4);
    }
    else /* Invalid delimiter */
    {
      /* Advance 4 octets and try again */
      offset += 4;
    }
  }
}
```

NOTE 1—This algorithm is not optimized for efficiency.

NOTE 2—The delimiter signature can be used to reduce the amount of computation required while scanning for a valid delimiter. In this case, the receiver tests each possible delimiter for a matching Delimiter Signature field. Only when a match is discovered does it then check the CRC.

Figure S-1—A-MPDU parsing

S.3 Example of an RD exchange sequence

Figure S-2 shows an example of the operation of the RD rules, defined in 9.25.

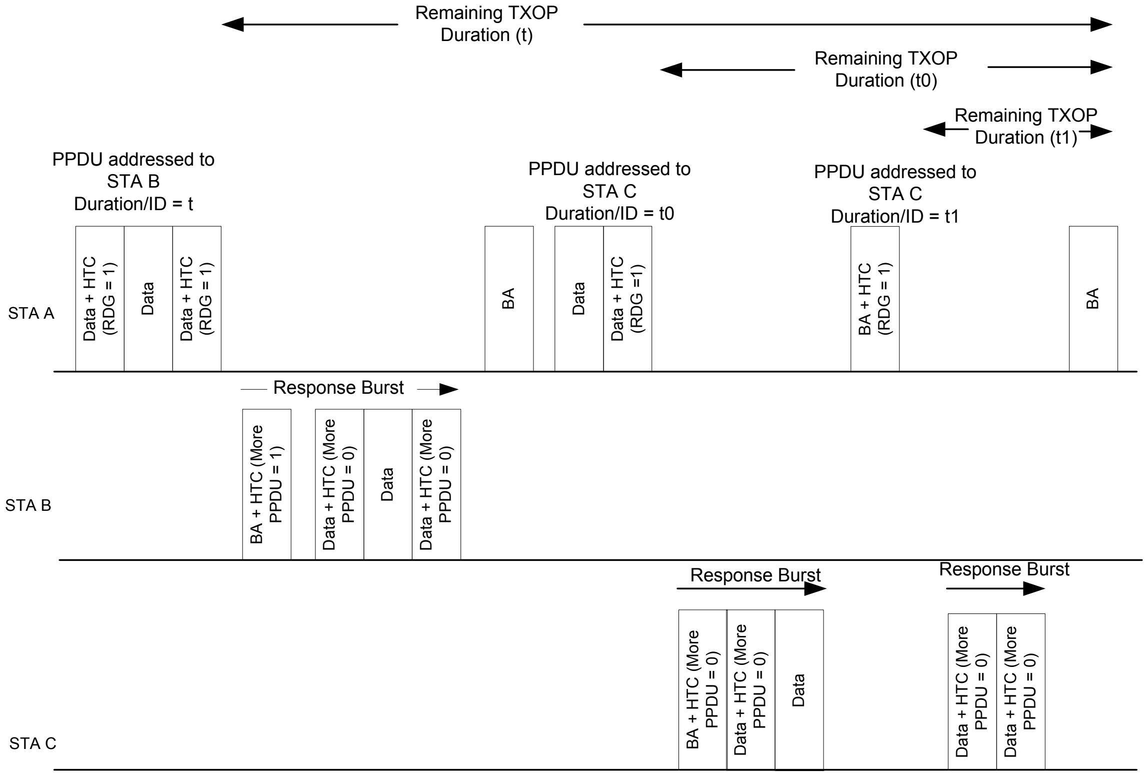

Remaining TXOP Duration (t)
Remaining TXOP Duration (t0)
Remaining TXOP Duration (t1)
PPDU addressed to STA B Duration/ID = t
PPDU addressed to STA C Duration/ID = t0
PPDU addressed to STA C Duration/ID = t1
STA A
Data + HTC (RDG = 1)
Data
Data + HTC (RDG = 1)
BA
Data
Data + HTC (RDG =1)
BA + HTC (RDG = 1)
BA
Response Burst
STA B
BA + HTC (More PPDU = 1)
Data + HTC (More PPDU = 0)
Data
Data + HTC (More PPDU = 0)
Response Burst
Response Burst
STA C
BA + HTC (More PPDU = 0)
Data + HTC (More PPDU = 0)
Data
Data + HTC (More PPDU = 0)
Data + HTC (More PPDU = 0)

Figure S-2—Example of RD exchange sequence showing response burst

The following is a summary of Figure S-2:

a) STA A (acting as RD initiator) transmits a PPDU containing MPDUs addressed to STA B (acting as RD responder). The Ack Policy field of the QoS data MPDUs in this PPDU is set to Implicit Block Ack Request. One or more MPDUs within this PPDU include an HT Control field with the RDG/More PPDU subfield set to 1, indicating an RDG. The Duration/ID field of MPDUs within the PPDU contains the remaining duration of the TXOP, t µs.

b) STA B (the RD responder) responds with the transmission of a +HTC BlockAck frame in which the RDG/More PPDU subfield is set to 1, indicating that another PPDU will follow a SIFS or RIFS interval after the end of the PPDU containing the BlockAck MPDU.

c) STA B transmits a PPDU (the second PPDU of an RD response burst) to STA A, with the Ack Policy field of its QoS data MPDUs set to Implicit Block Ack Request and containing one or more +HTC MPDUs in which the RDG/More PPDU subfield is set to 0, indicating that this is the last PPDU in the response burst.

d) STA A (the RD initiator) regains control of the TXOP and transmits a BlockAck MPDU addressed to STA B to acknowledge the MPDUs transmitted by STA B in the RD response burst.

e) STA A (the RD initiator) transmits a PPDU containing MPDUs addressed to STA C (acting as RD responder). The Ack Policy field of the QoS data MPDUs in this PPDU is set to Implicit Block Ack. This PPDU includes one or more +HTC MPDUs in which the RDG/More PPDU subfield is set to 1, indicating an RDG. The Duration/ID field of MPDUs in the PPDU contains the remaining duration of the TXOP, t0 µs.

f) STA C (the RD responder) transmits a PPDU to STA A, containing one or more +HTC MPDUs with the RDG/More PPDU subfield set to 0, indicating that this is the last PPDU in the response burst. This PPDU contains a BlockAck MPDU that is a response to the Implicit Block Ack request of the previous PPDU, plus QoS data MPDUs with the Ack Policy field set to Implicit Block Ack.

g) STA A (the RD initiator) regains control of the TXOP and transmits a BlockAck MPDU to STA C that acknowledges the MPDUs transmitted by STA C. This PPDU contains one or more +HTC MPDUs with the RDG/More PPDU subfield set to 1, indicating an RDG. The Duration/ID field of MPDUs in the PPDU contains the remaining duration of the TXOP, t1 µs.

h) STA C (the RD responder) transmits a PPDU to STA A, containing QoS data +HTC MPDUs with the Ack Policy field set to Implicit Block Ack Request and the RDG/More PPDU subfield set to 0. This is the only PPDU in the RD response burst.

i) STA A transmits a BlockAck MPDU to STA C that acknowledges the MPDUs transmitted by STA C in the RD response burst.

S.4 Illustration of determination of NDP addresses

Determination of NDP SA and DA are illustrated in Figure S-3 and Figure S-4.

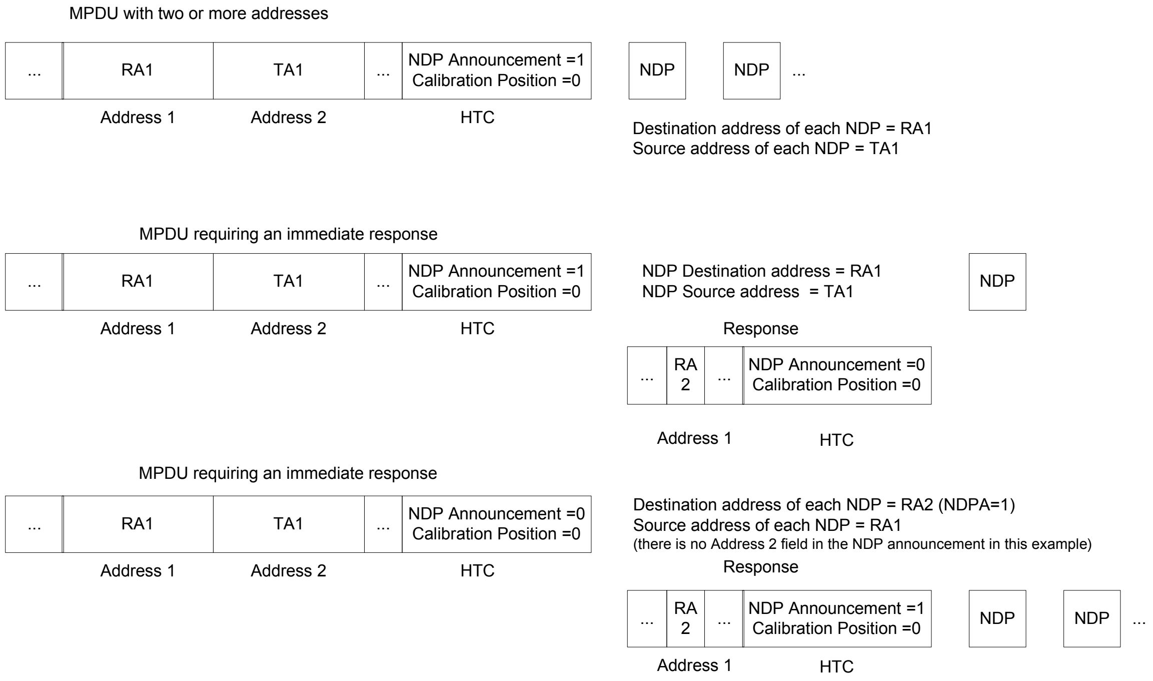


Figure S-3—Determination of NDP source and destination for unidirectional NDP sequences

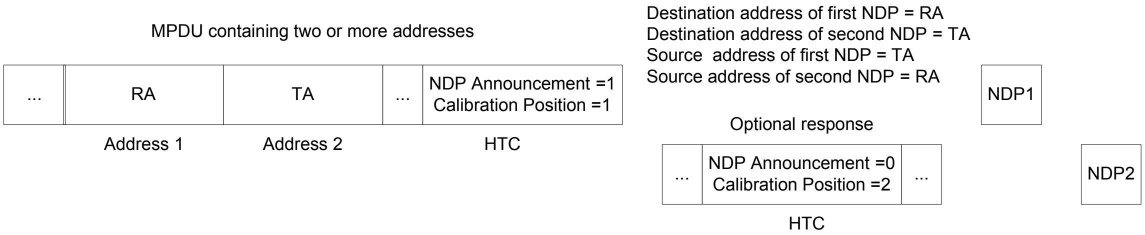

MPDU containing two or more addresses
...
RA
TA
...
NDP Announcement =1
Calibration Position =1
Address 1
Address 2
HTC
Destination address of first NDP = RA
Destination address of second NDP = TA
Source address of first NDP = TA
Source address of second NDP = RA
NDP1
Optional response
...
NDP Announcement =0
Calibration Position =2
...
HTC
NDP2

Figure S-4—Determination of NDP source and destination for bidirectional NDP sequence

S.5 20/40 MHz BSS establishment and maintenance

S.5.1 Signaling 20/40 MHz BSS capability and operation

A BSS that occupies 40 MHz of bandwidth and that is administered by an HT AP is called a 20/40 MHz BSS.

An HT AP that has dot11FortyMHzOperationImplemented equal to true sets the Supported Channel Width Set subfield of the HT Capabilities element to a nonzero value and may optionally operate a 20/40 MHz BSS. The Supported Channel Width Set subfield of the HT Capabilities element that is transmitted by the AP indicates the possible operating mode of the BSS and of the AP, but the value in this field is not an indication of the current operating channel width of either the AP or the BSS.

An HT AP signals the operating width of the BSS through the Secondary Channel offset field of the HT Operation element. A nonzero value in this field indicates that a secondary channel exists; in other words, the BSS is a 20/40 MHz BSS. A value of 0 in this field indicates that the BSS is operating as a 20 MHz BSS.

An HT AP that has dot11FortyMHzOperationActivated equal to true sets its STA Channel Width field of the HT Operation element to a nonzero value. This field signals the current operating mode of the AP, not the BSS. An HT AP may operate a 20/40 MHz BSS while it is operating as a 20 MHz device. Such a situation would support, for example, 40 MHz bandwidth DLS traffic among associated STAs, but only 20 MHz bandwidth traffic between STAs and the AP.

S.5.2 Establishing a 20/40 MHz BSS

Before starting a 20/40 MHz BSS, an 40-MHz-capable HT AP is required by the rules defined in 10.15.5 to examine the channels of the current regulatory domain to determine whether the operation of a 20/40 MHz BSS might unfairly interfere with the operation of existing 20 MHz BSSs. The AP (or some of its associated HT STAs) is required to scan all of the channels of the current regulatory domain in order to ascertain the operating channels of any existing 20 MHz BSSs and 20/40 MHz BSSs. This type of scanning is called *OBSS scanning*. The particulars of OBSS scanning are controlled by the following MIB attributes:

- dot11FortyMHzOptionImplemented
- dot112040BSSCoexistenceManagementSupported
- dot11FortyMHzIntolerant
- dot11BSSWidthTriggerScanInterval
- dot11BSSWidthChannelTransitionDelayFactor
- dot11OBSSScanPassiveDwell
- dot11OBSSScanActiveDwell

- dot11OBSSScanPassiveTotalPerChannel
- dot11OBSSScanActiveTotalPerChannel
- dot11OBSSScanActivityThreshold

Specific values for these MIB attributes are provided to set minimum scan times for passive scanning of each channel, and a separate minimum time is provided for active scanning of each channel. A total minimum amount of scanning per channel is required before a determination can be made to allow the operation of a 20/40 MHz BSS.

The rules that are applied when determining whether a 20/40 MHz BSS can be established are intended to avoid a full or partial overlap of the secondary channel of the 20/40 MHz BSS with an existing primary channel of either a 20 MHz BSS or a 20/40 MHz BSS. The lack of partially overlapping channels in the 5 GHz band allows these rules to be written as recommendations, while in the 2.4 GHz band, they are requirements.

An additional constraint on establishing a 20/40 MHz BSS includes the allowance for any IEEE 802.11 device to explicitly prohibit the operation of the 20/40 BSS mode due to other considerations. For example, if an IEEE 802.15.1 WPAN device is operating in the area, that device is likely to be unable to communicate successfully with a paired receiver if the number of available IEEE 802.15.1 WPAN channels falls below a given threshold. Operation of a 20/40 MHz BSS in the 2.4 GHz band can contribute to the reduction of the number of available IEEE 802.15.1 WPAN channels, possibly pushing the available channels below that threshold.

To promote sharing of the spectrum resource under such circumstances, it might be desirable to prohibit the operation of a 20/40 MHz BSS. As such, the 20/40 BSS coexistence mechanism allows a STA to transmit management frames containing a value of 1 for the Forty MHz Intolerant field. (The MIB attribute dot11FortyMHzIntolerant determines the setting of the value of the Forty MHz Intolerant field in transmitted frames, and the setting of the value of the MIB attribute is beyond the scope of this standard.) Receivers of such frames on any channel in the band are not allowed to establish a 20/40 MHz BSS anywhere in the band for a duration of dot11BSSWidthChannelTransitionDelayFactor × dot11BSSWidth-TriggerScanInterval seconds. To effect this requirement, monitoring STAs and APs maintain a countdown timer to indicate that a prohibition is in force. The countdown timer is reloaded with the value dot11BSSWidthChannelTransitionDelayFactor × dot11BSSWidthTriggerScanInterval seconds each time that the STA or AP observes a management frame containing a value of 1 for the Forty MHz Intolerant field. STAs communicate changes in their countdown counter (i.e., transitions between a zero value and a nonzero value) to their associated AP through the 20 MHz BSS Width Request field of the 20/40 BSS Coexistence Management frame.

S.5.3 Monitoring channels for other BSS operation

Some of the STAs that are associated with a 20/40 MHz BSS are required to perform monitoring in order to ensure that the conditions which allowed the establishment of the 20/40 MHz BSS do not change to conditions that would disallow the existence of the 20/40 MHz BSS.

Monitoring STAs keep a local record of channels that are in use by other BSSs. STAs that receive Beacon frames determine the primary channel by examining the DSSS Parameter Set element. Secondary channel existence and channel information are determined by examining the Secondary Channel Offset field of the 20/40 BSS Coexistence element. Monitoring STAs also record receptions of frames that contain a value of 1 for the Forty MHz Intolerant field. Any changes to the local record that would create a prohibition against 20/40 MHz BSS operation are immediately reported to the associated AP through the transmission of a 20/40 BSS Coexistence Management frame (i.e., with the 20 MHz BSS Width Request field set to 1). The reception of a 20 MHz BSS Width Request field equal to 1 at the AP causes the AP to switch the BSS to 20 MHz operation immediately.

Any change of a channel in use that had not previously been in use is also reported immediately within a 20/40 BSS Coexistence Management frame. The AP examines the new in-use channel information to determine whether any changes in BSS width operation are required (i.e., to see if any changes have occurred that indicate an overlap of the secondary channel). If a change to 20 MHz BSS operation is required, the change occurs immediately.

Conditions that prevent the operation of a 20/40 MHz BSS might be transient. If the number of channels in use is reduced or all STA signaling 40 MHz intolerance leave the area, an AP might choose to revert to 20/40 MHz operation, if allowed to do so. However, the conditions that allow 20/40 MHz BSS operation have to persist for a period of dot11BSSWidthChannelTransitionDelayFactor × dot11BSSWidthTriggerScanInterval seconds before a STA can signal that the conditions have changed, and the same period of time has elapsed before an AP can resume 20/40 MHz BSS operation.

STAs that do not monitor channels through OBSS scanning and do not report any channel information or received Forty MHz Intolerant field information to their associated AP are listed here:

- non-HT STAs
- HT STAs that are exempt from scanning as specified in 10.15.6
- HT APs, once the 20/40 MHz BSS is established
- HT STAs that are associated with an AP whose BSS is operating on a channel that is not in the 2.4 GHz band
- HT STAs that are associated with an HT AP that is not 40-MHz-capable (as indicated by a value of 0 in the Supported Channel Width Set subfield of the HT Capabilities element)

All other HT STAs that are associated with a 40-MHz-capable HT AP whose BSS is operating on a channel in the 2.4 GHz band monitor channels through OBSS scanning and report any channel information or received Forty MHz Intolerant field information to their associated AP.

All MIB attributes that are employed by the 20/40 BSS Coexistence mechanism are maintained by the AP, which has the ability to provide updates to the MIB attribute values to the associated STA by transmitting an OBSS Scan Parameters element.

Annex T

(informative)

Location and Time Difference accuracy test

T.1 Location via Time Difference of arrival

The location of a device may be determined in multiple methods, including the following:

- Signal strength at different sensors
- Time of flight between the device and different sensors
- Time difference of arrival between the device and pairs of sensors

A typical implementation of the time difference of arrival method requires that the sensors are co-channel, have synchronized clocks, and receive the same transmission from the device. The sensor's time of arrival measurements are shared, and the device location is determined via multilateration, i.e., each pair of sensor measurements provides a time difference of arrival measurement that represents a hyperbola in 2D space of most likely candidate locations, and the overlap of the multiple hyperbola from multiple pairs of sensors leads to the location estimate. The single time of departure is not relevant in this typical multilateration implementation as it is canceled out when time differences of arrival are computed.

When the sensors are on different channels, such as APs in a typical multi-channel deployment, the device transmits on each channel and thus with multiple times of departure. In this environment it is necessary for the device to advertise each time of departure so that it can be subtracted from the times of arrivals measured by sensors on different channels. Furthermore, the device's clock frequency typically does not match the clock frequency of the synchronized sensors, and so the device should transmit on the same channel multiple widely spaced times in order for the sensors to estimate the device's clock frequency relative to themselves and to be able to suitably scale the device's advertised times of departure.

An example transmission sequence comprises frames transmitted on channels 1, 6, 11, 1 at 2.4 GHz. From this, the device's clock frequency can be determined relative to the synchronized APs on channel 1, and all APs on channels 1, 6, and 11 can measure a time of arrival. These four transmissions enable a single location calculation. The Time of Departure Accuracy test in T.2 is designed to measure errors within such a transmission sequence that would degrade a multi-channel time difference of arrival location scheme.

T.2 Time Difference of departure accuracy test

The Time Difference of Departure accuracy test is an informative description of how time difference of departure accuracy can be measured for any parameterizable PHY waveform. This accuracy test does not apply when the Time of Departure timestamps are exclusively used for timing measurement (see 10.23.5).

The Time Difference of Departure accuracy test is parameterized by the following test parameters:

- TIME_OF_DEPARTURE(j,i), $1 \le j \le 500$, $1 \le i \le I$ (scalar entries)
- MULTICHANNEL_SAMPLING_RATE (scalar)
- FIRST_TRANSITION_FIELD(j,i), $1 \le j \le 500$, $1 \le i \le I$ (waveform entries)
- SECOND_TRANSITION_FIELD(j,i), $1 \le j \le 500$, $1 \le i \le I$ (waveform entries)
- TRAINING_FIELD(j,i), $1 \le j \le 500$, $1 \le i \le I$ (waveform entries)
- TIME_OF_DEPARTURE_ACCURACY_TEST_THRESH (scalar)

TIME_OF_DEPARTURE(j,i) is exposed externally through the TOD Timestamp field in the Time of Departure subelement in Location Track Notification frames.

The Time Difference of Departure accuracy test is performed as follows or in an equivalent or more accurate manner.

The Time Difference of Departure accuracy test is performed by instrumentation capable of converting signals transmitted on one or more channels into a stream of complex samples at f_S sample/s or more, with sufficient accuracy in terms of I/Q arm amplitude and phase balance, dc offsets, phase noise, etc, and at a fixed delay from the transmitter. The minimum sampling rate is MULTICHANNEL_SAMPLING_RATE sample/s respectively. A possible embodiment of such a setup is converting the signal to a low IF frequency with a cabled microwave synthesizer, sampling the signal with a digital oscilloscope and decomposing it digitally into quadrature components. The sampled signal is processed in a manner similar to an actual time of arrival processor, according to the following steps:

a) Repeat steps b) to j) indexed by $j = 1, \ldots , 500$
b) Repeat steps c) to i) indexed by $i = 1, \ldots , I$
c) Start of frame is detected.
d) Channel number, coarse and fine frequency offsets are estimated.
e) The packet is derotated according to estimated frequency offsets.
f) The transition from FIRST_TRANSITION_FIELD(*j,i*) to SECOND_TRANSITION_FIELD(*j,i*) is detected; and fine timing (with one sample resolution) is established.
g) The TRAINING_FIELD(*j,i*) of the derotated signal is up-sampled to meet the TIME_OF_DEPARTURE_ACCURACY_TEST_THRESH requirement. For example, a TIME_OF_DEPARTURE_ACCURACY_TEST_THRESH of 1 ns requires up-sampling of at least 1 GHz.
h) The up-sampled signal is cross-correlated with a reference waveform of the TRAINING_FIELD(*j,i*)
i) The measured time of departure $x_{j,i}$ is determined from the time of the peak of the magnitude of the cross-correlation.

NOTE—The time of the peak of the magnitude of the cross-correlation is actually a time of arrival measurement that equals the time of departure up to a fixed delay. Since the fixed delay is removed within step j), the fixed need not be known or explicitly compensated for.

j) Having repeated steps c) to i) I times, the (j,i)th time of departure error $e_{j,i}$ is calculated as TIME_OF_DEPARTURE(*j,i*) minus the synchronized time of departure. Defining $x_j = [x_{j,1},..,x_{j,I}]^T$ as the I measured times of departure, y = [TIME_OF_DEPARTURE(j,1), … TIME_OF_DEPARTURE(j,I)]^T, $e_j = [e_{j,1}, e_{j,I}]^T$ are the I time of departure errors and $X_j = [1_{I\times 1}, x_j]$, where $1_{I\times 1}$ is an $I \times 1$ matrix of ones, then the relative clock intercept, rci_j, and slope, rcs_j, between device and instrumentation are determined as the linear least squares line of best fit: i.e., $[rci_j, rcs_j]^T = (X_j^T X_j)^{-1} X_j^T y^j$. With these definitions and calculations, the synchronized time of departure $s_j = [s_{j,1},..,s_{j,I}]^T$ equals $rcs_j \times x_j + rci_j \times 1_{I\times 1}$, and so the I time of departure errors equal $e_j = y_j - s_j$.
k) Having repeated steps b) to j) 500 times, there are $500 \times I$ values of the time of departure errors $e = [e_{1,1}, e_{500,I}]$
l) The Time of Departure accuracy test is passed if
   1) The RMS value of e is less than aTxPmdTxStartRMS, and
   2) aTxPmdTxStartRMS is less than TIME_OF_DEPARTURE_ACCURACY_TEST_THRESH, where the units of e, aTxPmdTxStartRMS, and TIME_OF_DEPARTURE_ ACCURACY_TEST_THRESH are properly accounted for.

NOTE 1—One possible implementation of a time of departure measurement system is a free-running oscillator clocking (a) the digital-to-analog converter(s) used to transmit the packet, (b) a 32-bit continuously counting counter and (c) a hardware finite state machine such that PMD_TXSTART.request causes a transition within the finite state machine that in turn causes frame transmission at the DACs a fixed number of cycles later; where the time of departure is recorded as the value of the counter at that transition minus aTxPmdTxStartRFDelay (using TIME_OF_DEPARTURE_ClockRate), where aTxPmdTxStartRFDelay can vary by channel. In this implementation, the principal source of time of departure error is short term oscillator imperfection (e.g., phase noise) and RF group delay variation across channels uncompensated by aTxPmdTxStartRFDelay.

NOTE 2—1 ns of time of departure error corresponds to approximately 0.3 m of distance error, so high location accuracy depends upon a tight time of departure standard deviation.

Annex U

(informative)

Example use of the Destination URI for Event and Diagnostic Reports

U.1 Destination URI payload

An example of the payload used to transmit Event and Diagnostic reports shown in Table U-1. The protocol used to transport the Destination URI payload is beyond the scope of this standard. An example use of the Destination URI is given in U.2.

Table U-1—Destination URI payload

Size (octets)	Information
6	BSSID
6	Reporting STA Address
variable	Event or Diagnostic Report frame contents

U.2 Use of HTTP (or HTTPS) for Destination URI of Event and Diagnostic Reports

Under certain conditions, a non-AP STA may need to send Event and Diagnostic reports to an AP using the Destination URI advertised by the AP in the request frame. A suitable higher layer protocol that could be used to transport the Event or Diagnostic report is HTTP or HTTPS.

For example, consider the following:

1) IT is investigating a WLAN coverage problem and uses a non-AP STA with a WLAN and Ethernet adapters to collect some additional information.
2) The non-AP STA with MAC 00:ff:fd:00:00:01 has received a Diagnostic Report request from AP 00:ff:fe:00:00:10 and is in fringe coverage. The non-AP STA has an Ethernet adapter that is connected.
3) The AP includes an Alternate Destination URI of http://www.myserver.mycompany.com in the Diagnostic Report frame.
4) The non-AP STA loses WLAN connectivity while trying to transmit a Diagnostic Report frame to the AP and the non-AP STA's SME determines that it can use the Alternate Destination URI to send the Diagnostic Report frame using the Ethernet link.
5) The non-AP STA POSTs the Diagnostic Report as follows:

```
POST /wnm/msg/00-ff-fd-00-00-01/msg1 HTTP/1.1
Host: http://www.myserver.mycompany.com
Content-Type: application/octet-stream
Content-Encoding-Type: base64
Content-Length: ?? (length of data as specified in Figure 8-470 in 8.5.14.5)
```

<encoded data = 00 ff fe 00 00 10 00 ff fd 00 00 01 …>

In the HTTPS case, the non-AP STA would need to be provisioned with credentials to establish the TLS connection prior to posting the message over HTTP. The HTTP post would work as described previously.

Annex V

(informative)

Interworking with external networks

V.1 General

The purpose of this informative annex is to describe and clarify the support for interworking with external networks including the support for network discovery and selection, QoS mapping, SSPN interface, and emergency services and to provide some background information and recommended practices.

V.2 Network discovery and selection

V.2.1 General

Interworking service provides features to support the network discovery and selection process a STA uses to choose the network with which to associate. GAS provides a non-AP STA access to an advertisement server (e.g., an access network server or an IEEE 802.21 information server), which can provide a rich set of information to aid the network discovery and selection process. In addition, interworking service provides lightweight features that also facilitate this process. The following subclauses describe several use cases illustrating how these features can be used to aid in network discovery and selection:

— **Airport:** A traveling businesswoman needs to connect via an airport hotspot to her enterprise network to download email and information from the customer database.

— **Shopping:** A shopper visits a shopping mall and wants to use a smartphone to discover items on sale.

— **Sales meeting:** A salesman visiting a customer accesses his guest network.

— **Museum:** A visitor to a museum uses a smartphone to obtain virtual docent service.

— **Emergency call:** A traveler needs to make an emergency call while in another country.

— **Emergency alert:** A traveler, having enabled the display of emergency alerts, arrives at a new destination.

V.2.2 Airport

A traveling businesswoman arrives for the first time at an airport having a WLAN. This user wants to download email onto her laptop utilizing the airport's hotspot, a chargeable network. Once associated, the woman needs to connect via VPN connection back to her company's servers to access email and information from the customer database. This is performed as follows:

a) The laptop's non-AP STA performs an active scan by transmitting a Probe Request frame containing the wildcard SSID and an Interworking element with Access Network Type subfield set to "Chargeable Public Network." In response, it receives Probe Response frames from several of the airport's APs, in the immediate neighborhood, for the SSID "Narita Hotspot."

b) The Probe Response received by the laptop indicated the following capabilities:

   1) Extended Capabilities element indicates: AP provides interworking service.

2) Interworking element indicates: venue group = 1 (Assembly) and venue type = 3 (passenger terminal), Internet = 1 (Internet access available), ASRA = 1 (there is an additional step required for network access).
3) Advertisement Protocol element including the Advertisement Protocol ID set to MIH Information Service.
4) Roaming Consortium element present containing an OI for “Hotspot Roaming International.”
5) There is no RSNE present in the received beacon frame.

c) Since the laptop’s SME does not recognize the Roaming Consortium OI, it invokes the GAS protocol to query the network’s IEEE 802.21 IS. The IEEE 802.21 IS’s response indicates the roaming partners for “Narita Hotspot” and the laptop have security credentials for one of them.
d) Since the AP indicated ASRA = 1, the SME again invokes the GAS protocol to retrieve the Network Authentication Type ANQP-element. The response indicates that https redirection is in use and provides the Redirect URL of hotspot.narita.co.jp. Note that this is helpful since some networks use conditional redirection—that is, access to a walled garden is provided for free, but a subscription fee is required to access the Internet.
e) Since the laptop’s SME now knows it should be able to successfully authenticate with the network, the STA associates to the AP.
f) The following operations are then carried out by higher layers operating within the laptop:
   1) The laptop’s SME autonomously launches an http client and provides to it the URL of hotspot.narita.co.jp, which provides the proper security credentials to the network and thereby successfully authenticates it to the network.
   2) The VPN client is autonomously launched and a secure session to the user’s corporate network is established. Then the user launches the email application to download email and other required information.

V.2.3 Shopping

A shopper visits a shopping mall and wants to use a smartphone to discover items on sale. In this mall, the mall’s IT department is providing WLAN facilities for all the stores in the mall; therefore, there is only one SSID for shoppers (i.e., there is not a different SSID for each store in the mall). The user arrives at the mall and taps an icon on the screen to put the smartphone in “shopping mode.” The smartphone’s shopping application causes the non-AP STA to carry out the following steps:

a) The smartphone’s non-AP STA performs an active scan by transmitting a Probe Request frame containing the wildcard SSID and an Interworking element with Access Network Type subfield set to “Free Public Network.” In response, it receives Probe Response frames from several of the mall’s APs, but only one SSID is provided, which is “Silicon Valley Mall.” The mall’s APs did not transmit Probe Responses for the SSIDs “Engineering,” “Deliveries,” and “Janitorial” since their access network type is “Private network.”
b) The Probe Response received by the smartphone indicated the following capabilities:
   1) Extended Capabilities element indicates: AP provides interworking service.
   2) Interworking element indicates: venue group = 6 (mercantile) and venue type = 4 (shopping mall), Internet = 0 (unspecified).
   3) RSNE indicates: IEEE 802.1X authentication.
c) Since the AP indicated Interworking service is available, the smartphone’s non-AP STA uses the MLME-GAS.request primitive to invoke GAS to request the Capability List ANQP-element (see 8.4.4.3). In the Capability List ANQP-element, the AP has indicated support for Venue Name and Domain Name. Subsequent to receipt of the Capability List ANQP-element, the non-AP STA invokes the MLME-GAS.request primitive to retrieve the other two lists.

d) Next, the non-AP STA's Supplicant searches the received Domain Name list to determine whether it has any stored credentials for these domains. If so,
   1) The smartphone autonomously associates to the "Silicon Valley Mall Shopping" SSID and displays the following information:
      i) Venue Name: Silicon Valley Mall, 1234 Main Street, Rownhams, CA 98765-1234
      ii) SSID: Silicon Valley Mall
      iii) Venue type: Shopping Mall
   2) The Supplicant autonomously provides the security credentials for the selected domain.
e) Higher layer protocols then download discount coupons being offered for items on sale.

V.2.4 Sales meeting

A salesman travels across town to a meeting at ACME Manufacturing. While there, this user needs to send email to get a document from engineering. On his laptop, he requests the WLAN via the laptop's UI to search for guest networks. The laptop performs the following steps:

a) The laptop's non-AP STA performs an active scan by transmitting a Probe Request frame containing the wildcard SSID and an Interworking element with Access Network Type subfield set to "Private Network with Guest Access." In response, it receives Probe Response frames from several of ACME Manufacturing's APs, but only one SSID is provided, which is "Guest." ACME Manufacturing's APs did not transmit Probe Responses for the SSIDs "Engineering" and "Finance" since their access network type is "Private network."
b) The Probe Response received by the laptop indicated the following capabilities:
   1) Extended Capabilities element indicates: AP provides interworking service.
   2) Interworking element indicates: Internet is available, venue group = 2 (Business) and venue type = 8 (Research and Development Facility).
   3) RSNE indicates: IEEE 802.1X authentication with CCMP pairwise and group cipher suites.
c) Since the AP indicated interworking service is available, the laptop's non-AP STA uses the MLME-GAS.request primitive to invoke GAS to request the Capability List ANQP-element (see 8.4.4.3). In the Capability List ANQP-element, the AP has indicated support for Venue Name. Upon receipt of the Capability List ANQP-element, the non-AP STA again invokes the MLME-GAS.request primitive to retrieve the Venue Name.
d) The laptop's UI displays the following information and automatically associates to the network:
   1) SSID: Guest (Type: Private network with Guest access)
   2) Venue Name: ACME Manufacturing, 1234 Main Street, Rownhams, CA 98765-1234
   3) Venue type: Research and Development Facility
   4) Internet is available
e) Upon prompt, the user enters the username and password supplied by his point of contact from ACME Manufacturing and is then able to send and receive email.

V.2.5 Museum

A visitor enters a museum that is advertising virtual docent service (audio tracks describing each of the major exhibits). The visitor taps an icon on a smartphone and requests it to search for free networks. The smartphone then carries out the following:

a) The smartphone's non-AP STA performs an active scan by transmitting a Probe Request frame containing the wildcard SSID and an Interworking element with Access Network Type subfield set to "Free Public Network." In response, it receives Probe Response frames from several of the museum's APs, but only one SSID is provided, which is "Visitors." The museum's APs did not transmit Probe Responses for the SSID “Maintenance” since its access network type is “Private network.”

b) The Probe Response received by the smartphone indicated the following capabilities:
   1) Extended Capabilities element indicates: AP provides interworking service.
   2) Interworking element indicates: venue group = 1 (assembly), venue type = 9 (museum), and ASRA = 0 (no additional steps are required for access).
c) Since the AP indicated interworking service is available, the smartphone’s non-AP STA uses the MLME-GAS.request primitive to invoke GAS to request the Capability List ANQP-element (see 8.4.4.3). In the Capability List ANQP-element, the AP has indicated support for Venue Name. Upon receipt of the Capability List ANQP-element, the non-AP STA again invokes the MLME-GAS.request primitive to retrieve the Venue Name.
d) The smartphone’s UI displays the following information, asking the user whether they wish to connect to the network:
   1) Venue Name: Museum of Modern Art (MOMA)
   2) SSID: Visitors
   3) Venue type: Museum
   4) No authentication required
e) The user taps the “Connect” icon on the smartphone’s display. Note that the smartphone’s non-AP STA knows that the network uses open system authentication since there is no RSNE present in the beacon and ASRA = 0.

V.2.6 Emergency call

A traveler needs to make an emergency call while in another country. The traveler may not be aware of the emergency call numbers that are used in that country. Being in this new location for the first time, the traveler is not aware of which local access points provide access for emergency calling. (Note that emergency calling requires higher layer application(s) that are outside the scope of this standard.) This is performed as follows:

a) The traveler’s non-AP STA performs an active scan by transmitting a Probe Request frame containing the wildcard SSID and an Interworking element. (If the non-AP STA is already associated with an AP that has indicated support for emergency service in its beacon, the probe request would not be necessary.)
b) In response, it receives Probe Response frames from one or more APs indicating support for emergency service in the Access Network Options field of the Interworking element.
   1) Extended Capabilities element indicates: AP provides interworking service.
   2) Emergency services reachability is indicated by the ESR and UESA fields in the Access Network Options field in the Interworking element.
   3) A dedicated emergency services network is indicated by the Access Network Type field in the Interworking element.
c) The traveler’s non-AP STA then requests the Emergency Call Number ANQP-element with a GAS Initial Request frame.
d) The GAS Initial Response frame from the AP provides the Emergency Call Number, for example the sequence of digits “911” or an emergency service URN label, URI pair “urn:service:sos.fire, tel:112;sip:+15555551002@fire.com” [B41].
e) This information is passed to the higher layer application in the traveler’s non-AP STA.
f) The traveler’s non-AP STA then associates with the AP.
g) The traveler’s non-AP STA then places the emergency call, with an expedited bandwidth request (EBR) in an ADDTS frame to provide priority to the emergency call.

V.2.7 Emergency alert

A traveler has enabled the display of emergency alerts on a wireless device (non-AP STA) by appropriately setting the higher layer emergency alert application on the device. The traveler arrives at a new destination and turns on the device. The device, when switched on, will perform a search and then associate with an AP to which the traveler has a subscription or associate with an open AP (if the traveler has enabled the device to do that). During the steps leading up to association, the device, when it becomes aware of an emergency alert, will obtain and display it. The emergency alert will likely be obtained from the AP that has the service to which the traveler has subscribed, but the device may also obtain the emergency alert from other APs, if an AP to which the traveler has a subscription is not available. This is performed as follows:

a) The traveler’s non-AP STA performs an active scan by transmitting a Probe Request frame containing the wildcard SSID. If it is associated with an AP, it checks the beacon.
b) In response, it receives Probe Response frames from one or more APs, which it checks (or, alternately, checks the beacon) for the subscription and also whether the Emergency Alert element indicates that there are emergency alert message(s).
c) If there are one or more emergency alert messages to be downloaded, the Emergency Alert Identifier element provides an Alert Identifier Hash value.
d) The device uses the hash of each of the available emergency message(s) to determine whether that message has already been received or whether it is a new message that must be downloaded.
e) If there are one or more emergency alert messages to be downloaded, the device forms the EAS Message URI by concatenating the Emergency Alert Server URI with the hexadecimal numerals of the Alert Identifier Hash, using the specified method.
f) Should the device be unassociated with an AP when it determines that a new emergency alert is available, it retrieves the EAS message using either GAS procedures with Advertisement Protocol ID field set to the value for EAS. If the device is associated with the AP, it retrieves the EAS message using either GAS procedures with Advertisement Protocol ID field set to the value for EAS or, for example, using HTTP.

V.3 QoS mapping guidelines for interworking with external networks

V.3.1 General

The EDCA and HCCA mechanism defined in 9.19 provide QoS control at the MAC layer. However, the QoS control parameters used by the EDCA and HCCA cannot match directly with other QoS control parameters of the interworked external networks, e.g., SSPN. For example, the SSPN could have different metrics for defining the QoS levels. Destination Network 1 (DN1) and DN2 can use DSCP values differently, in which case, STA1 and STA2 would require different QoS mapping information. Therefore, mapping from these external QoS control parameters to the QoS parameters of this standard is necessary.

The QoS parameters mapping can be used for both uplink and downlink data transmission:

- For uplink: at the non-AP STA, external QoS parameters are mapped to IEEE 802.11 QoS parameters, e.g., DSCP to IEEE 802.11 User Priority and in turn to EDCA ACs. This mapping helps the non-AP STA to construct correct QoS requests to the AP, e.g., ADDTS Request and to transmit frames at the correct priority.
- For downlink: at the AP, DSCP values are mapped to EDCA UPs. Optionally, the non-AP STA can use TSPEC and TCLAS elements in an ADDTS Request frame to setup a traffic stream in the BSS. In this method, the User Priority is specified in the TCLAS element. The policy used by the AP to choose a specific method to map frames to user priorities is outside the scope of this standard.

Different external networks can use different DSCP sets for the same services as described in V.3.3. For example, a 3GPP network can use different code points from that of an enterprise network. The QoS Map distribution mechanism defined in 10.24.9 provides means to communicate to the STA's mapping information from the network.

V.3.2 Determination of the mapping for a STA

The QoS mapping to be applied depends upon the network the non-AP STA is accessing. In an interworking IEEE 802.11 infrastructure setting, the same physical AP can serve non-AP STAs from different SSPNs on different BSSIDs. As such, these STAs are separated into different BSSs. Figure V-1 presents an example of the scenario using authentication, authorization, and accounting (AAA). In Figure V-1, AAA Server 1 controls access to DN-1 and AAA Server 2 controls access to DN-2.

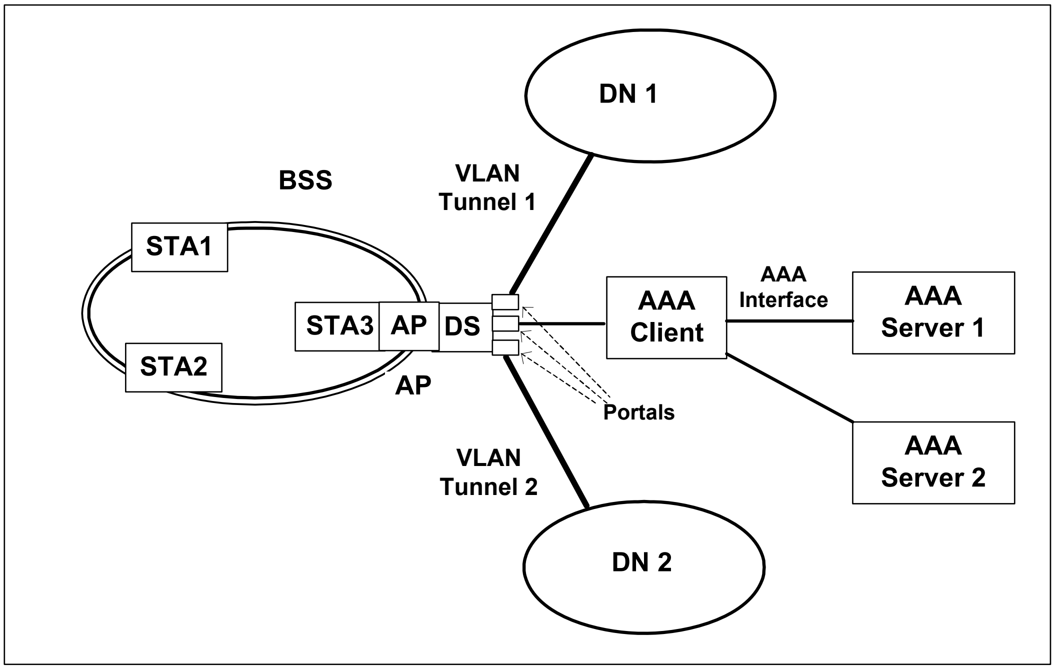


Figure V-1—Interworking IEEE 802.11 infrastructure supporting multiple SSPNs

V.3.3 Example of QoS mapping from different networks

IEEE 802.1d UPs map to EDCA ACs, as described in Table 9-1. The use of DSCP sets differs from network to network. Table V-1 shows examples of DCSP mappings.

NOTE—The mapping of the DSCP to 3GPP Traffic Class is available in GSMA, IR.34 v4.6 [B16] (similar to that of GSMA IREG34). See TR 21.905 [B2] for definition of general packet radio service (GPRS) roaming exchange. Table V-1 is extended to cover the EDCA ACs mapping. This mapping can also apply to other networks that adopt the 3GPP QoS definitions, e.g., 3GPP2.

Table V-2 shows an example mapping based on application classes defined in IETF RFC 4594. Mapping between DSCP and UP can be done using Exception fields or by range. The use of Exception fields will map a DSCP to a UP according to Table V-2. Mapping by range will require the setting of DSCP ranges as shown in Table V-3.

Table V-1—Mapping table of DSCP to 3GPP QoS information and EDCA ACs

3GPP QoS Information		**DiffServ PHB**	**DSCP**	**QoS Requirement on GPRS Roaming Exchange**				**EDCA Access Category**	**UP (as in IEEE 802.1d)**
Traffic Class	THP			Max Delay	Max Jitter	MSDU Loss	MSDU Error Ratio		
Conversational	N/A	EF	101110	20 ms	5 ms	0.5%	10^{-6}	AC_VO	7, 6
Streaming	N/A	$AF4_1$	100010	40 ms	5 ms	0.5%	10^{-6}	AV_VI	5, 4
Interactive	1	$AF3_1$	011010	250 ms	N/A	0.1%	10^{-8}	AC_BE	3
	2	$AF2_1$	010010	300 ms	N/A	0.1%	10^{-8}	AC_BE	3
	3	$AF1_1$	001010	350 ms	N/A	0.1%	10^{-8}	AC_BE	0
Background	N/A	BE	000000	400 ms	N/A	0.1%	10^{-8}	AC_BK	2,1

Table V-2—Example Enterprise DSCP to UP/AC mapping

Application Class	**Per-hop behavior (PHB)**	**IEEE 802.1d User Priority**	**Access Category**
Network Control	CS6	7	AC_VO
Telephony	EF	6	AC_VO
RT Interactive	CS4	6	AC_VO
Multimedia Conference	AF4x	5	AC_VI
Signaling	CS5	5	AC_VI
Broadcast Video	CS3	4	AC_VI
Multimedia Stream	AF3x	4	AC_VI
Low Latency Data	AF2x	3	AC_BE
High Throughput Data	AF1x	2	AC_BE
OAM	CS2	2	AC_BE
Standard	DF	0	AC_BE
Low Priority/Background	CS1	1	AC_BK

Table V-3—UP to DSCP range mapping example

UP Range	DSCP Low	DSCP High
UP 0 Range	0	0
UP 1 Range	1	9
UP 2 Range	10	16
UP 3 Range	17	23
UP 4 Range	24	31
UP 5 Range	32	40
UP 6 Range	41	47
UP 7 Range	48	63

Furthermore mapping by range will require an additional exceptional element to map DSCP 32 to UP 6.

NOTE—Twenty-one Exception fields are provided to give more flexibility in defining the QoS Map and it is currently the number of Fibs defined by the IETF.

V.4 Interworking and SSPN interface support

V.4.1 General

The interworking service architecture defines the scope of the SSPN interface. This interface is provided by the IEEE 802.11 MAC to support the interworking service. In an interworking scenario, the IEEE 802.11 infrastructure is operating in infrastructure mode.

Figure V-2 shows an example implementation of the control aspect of the Interworking Interface. As shown in the figure, the Interworking Interface consists of two parts: the generic SSPN Interface between the AP and the AAA Client; and the AAA Interface between the AAA Client and the corresponding AAA Server in the SSPN. Depending on the implementation the AAA Client can be collocated with the AP or stand alone serving as a proxy or translation agent between the SSPN Interface and AAA Interface. The AAA Interface serves as a transparent carrier of the SSPN interface.

The possible interactions over the SSPN interface are defined in 10.24.5. The information transferred over the SSPN Interface is defined in V.4.2. This interface results in parameters being set in the dot11InterworkingTable MIB. The AP's SME thereafter uses these parameters to permit or deny, as appropriate, services to non-AP STAs.

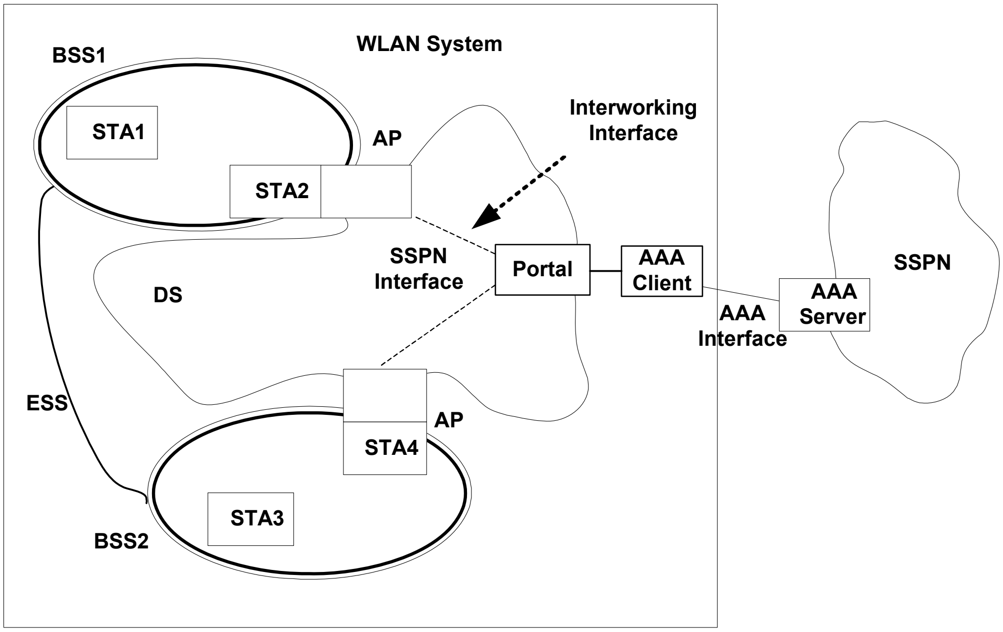


Figure V-2—Basic architecture of the interworking service

V.4.2 SSPN interface parameters

The parameters for each associated non-AP STA defined in this clause cross the SSPN Interface, i.e., between AP and AAA Client as shown in Table V-4.

Table V-4—SSPN Interface information or permission parameters

Information or permission name	From AN to SSPN	From SSPN to AN	Per non-AP STA entry
Non-AP STA MAC	+		+
Non-AP STA User ID	+	+	+
Non-AP STA Interworking Capability	+		+
Link Layer Encryption Method	+		+
Authorized Priority		+	+
Authorized Rate		+	+
Authorized Delay		+	+
Authorized Service Access Type		+	+
Authorized Service Access Information		+	+
non-AP STA Transmission Count	+		+
non-AP STA Location Information	+		+
non-AP STA state Information	+		+

The SSPN Interface parameters are stored in the AP with corresponding MIB attributes as defined in Annex C, and are used by the Interworking Service Management function in the SME. The MIB variables themselves, which are used by the APs SME, are read only.

V.4.2.1 Non-AP STA MAC

This is the MAC address of the non-AP STA accessing the interworking service through the AP. It can be requested by the external network, e.g., a 3GPP network, for fraud prevention. The non-AP STA MAC address is normally available through MLME-SAP, e.g., MLME-ASSOCIATE.indication, and should be forwarded by the AS to the AAA Server entity in the SSPN through the AAA Interface.

The AP stores the non-AP STA MAC address in the corresponding dot11NonAPStationMacAddress element of its MIB.

V.4.2.2 Non-AP STA user ID

This parameter contains the subscriber information of the non-AP STA for the interworking service. It is provided by the non-AP STA through the RSNA establishment process to the AAA Server; in turn, the AAA Server provides it back to the AP via the SSPN interface. It is in the form of a NAI, i.e., it contains both the user's identity and its SSP information.

NOTE—The reason the AAA Server provides the user identity back to the AP is that some EAP methods use encrypted tunnels to maintain confidentiality of the user and thus the AP might not otherwise be able to learn the user's identity.

The AP stores the associated non-AP STA User ID in the corresponding dot11NonAPStationUserIdentity element of its MIB.

V.4.2.3 Non-AP STA interworking capability

This parameter is derived from the non-AP STA's Extended Capabilities element, which is included in (Re)Association Request frames.The AP SME obtains this information from the MLME-SAP, e.g., MLME-ASSOCIATE.indication. This information needs to be passed over the SSPN interface since the service authorization decisions can depend on the non-AP STA capabilities.

The AP stores the associated non-AP STA Interworking Capability in the corresponding dot11NonAPStationInterworkingCapability element of its MIB.

V.4.2.4 Link layer encryption method

This parameter indicates the link layer encryption method selected during the RSNA establishment process for protecting the unicast communication between the non-AP STA and the AP. The cipher suite format of this element is drawn from the RSNE defined in 8.4.2.27. The AP obtains this information about the STA via the MLME SAP.

In the interworking service, the SSPN also participates in the selection of the cipher suite selection, as described in 10.24.5. Therefore, the link layer encryption method selected will meet or exceed the security requirement of the SSPN.

NOTE—In interworking, the SSPN can require visibility and configurability of the STA access.

With this information available to the SSPN, the operator would be able to have better control, e.g., barring access to IEEE 802.11 networks if null encryption is used. This is also related to the operator network's configuration, e.g., if preauthentication should be supported.

The AP stores the information in the corresponding dot11NonAPStationCipherSuite element of its MIB.

V.4.2.5 Authorized priority

This parameter is used for admission control and user-priority policing at the AP. It is based on the Infrastructure Authorization Information delivered from the SSPN during the AAA procedure. The Authorized Priority specifies the authorized User Priorities that the non-AP STA is allowed to use during the Interworking access. It also specifies whether the non-AP STA can use HCCA.

For EDCA operation, the AP stores the information in its corresponding dot11NonAPStationAuthAccessCategories element of its MIB after mapping the priority according to Table 9-1. For HCCA operation, the AP stores the information in dot11NonAPStationAuthHCCAHEMM.

V.4.2.6 Authorized maximum rate

This parameter is used for admission control decisions or policing actions at the AP. It is based on the Infrastructure Authorization Information delivered from the SSPN during the AAA procedure. For EDCA operation, this element contains a list of four MaxRate subelements indicating the maximum rate allowed for the access categories. For HCCA operation, there is one MaxRate subelement. Each of the MaxRate is an unsigned integer and in the unit of kilobits per second. An additional subelement provides the maximum rate at which a non-AP STA can source group addressed frames.

The AP stores the information in the corresponding dot11NonAPStationAuthMaxVoiceRate, dot11NonAPStationAuthMaxVideoRate, dot11NonAPStationAuthMaxBestEffortRate, dot11NonAPStationAuthMaxBackgroundRate, dot11NonAPStationAuthMaxHCCAHEMMRate and dot11NonAPStationAuthMaxSourceMulticastRate elements of its MIB.

V.4.2.7 Authorized service access type

This per-non-AP STA parameter indicates the access type allowed for the non-AP STA based on the SSPN decision. The AP will use this information for authorization requests from the STA, e.g., allow or disallow direct link operation and group addressed services. The element uses TruthValues to indicate the service type authorized. The following MIB variables are used:

- dot11NonAPStationAuthDls is to authorize a non-AP STA to use DLS
- dot11NonAPStationAuthSinkMulticast is to authorize a non-AP STA to request group addressed stream(s) from the network
- dot11NonAPStationAuthMaxSourceMulticastRate is to authorize a non-AP STA to source group addressed stream(s) to towards the network

V.4.2.8 Authorized delay

This parameter is used for admission control decisions at the AP. It is based on the Infrastructure Authorization Information delivered from the SSPN during the AAA procedure. This element is only used for HCCA operation, and contains one subelement. An AP should deliver frames to a non-AP STA within the time period specified in this attribute. Furthermore, when a non-AP STA requests admission control, the requested delay is only approved if it is equal to or greater than the value stored in the corresponding element. Each element is an unsigned integer that measures delay in units of microseconds.

The AP stores the information in the corresponding dot11NonAPStationAuthHCCAHEMMDelay elements of its MIB.

V.4.2.9 Authorized service access information

This parameter contains the relevant information for the AP to enforce the authorized service access type indicated in the Authorized Service Access Type element.

The Authorized Service Access parameters provide the VLAN assignment (VLAN ID and name) to which frames to or from the non-AP STA are bridged. The element includes VLAN ID (dot11NonAPStationVLANId) and VLAN Name (dot11NonAPStationVLANName).

V.4.2.10 Non-AP STA transmission count

This parameter indicates the count of the data traffic transmitted to and received from a non-AP STA. Such information would be used by the on-line charging and accounting function, especially for the IEEE 802.11 WLAN local service, where the data traffic does not necessarily go through the SSPN network. In such cases, Layer 3 accounting/charging information is not reliable since addresses could be spoofed. Layer 2 would be a better place to collect such information since due to the cryptographic security association that exists between the non-AP STA and AP.

The non-AP STA Transmission Count element includes information stored in the corresponding dot11NonAPStationVoiceMSDUCount, dot11NonAPStationVideoMSDUCount, dot11NonAPStationBestEffortMSDUCount, dot11NonAPStationBackgroundMSDUCount, dot11NonAPStationHCCAHEMMMSDUCount, dot11NonAPStationMulticastMSDUCount, dot11NonAPStationVoiceOctetCount, dot11NonAPStationVideoOctetCount, dot11NonAPStationBestEffortOctetCount, dot11NonAPStationBackgroundOctetCount, dot11NonAPStationHCCAHEMMOctetCount, dot11NonAPStationMulticastOctetCount elements of the AP's MIB.

V.4.2.11 Non-AP STA location information

This parameter provides information about the STA's location to the SSPN. It is required by the SSPN applying location based service control. In the IEEE 802.11 network, the non-AP STA location is approximated using the AP's location information. This includes two type of formats, Geospatial and Civic Location.

The information to be placed in the non-AP STA Location element is obtained from the dot11APGeospatialLocation and dot11APCivicLocation elements of the AP MIB.

V.4.2.12 Non-AP STA State Information

This parameter indicates whether non-AP STA is Active Mode or Power Saving. Information in this element is obtained from the corresponding dot11NonAPStationPowerManagementMode element of the associated AP MIB.

V.5 Interworking with external networks and emergency call support

V.5.1 General

Emergency services define the IEEE 802.11 functionality to support an emergency call (e.g., E911) service as part of an overall multi-layer solution, specifically capability advertisement and access to emergency services by STAs not having proper security credentials. "Multi-layer" indicates that emergency services will be provided by protocols developed in part by other standards bodies; see IETF ECRIT [B23], 3GPP TS 23.167 [B1], and 3GPP TS 22.067 [B3]. Three features of interworking with external networks support emergency call services.

The first feature is a mechanism for a non-AP STA to signal to an AP that a call is an emergency call. This is useful in the case where the access category to be used to carry the emergency call traffic (typically AC_VO) is configured for mandatory admission control. If the WLAN is congested, then the AP can deny the TSPEC request for bandwidth to carry the call. However, if the AP is able to determine that the call is an emergency call, then it can invoke other options to admit the TSPEC request.

The second and third features provide the means for a client without proper security credentials to be able to place an emergency call. The second feature makes use of the Interworking element, which can be included in Association request frames in order to bypass the IEEE 802.1X port at an AP for unauthenticated access to emergency services. This is described further in V.5.5. The third feature makes use of an SSID configured for Open Authentication to provide emergency services and is described in V.5.3.

The STA has the burden to confirm the availability of emergency services from the IEEE 802.11 network, including that the network is authorized for emergency services. The time it takes for a client to find an authorized emergency services network is related to the speed of forward progress the authorized network can make over the air with the STA, relative to all of the other networks (attackers as well), and is inversely related to the number of false advertisements. A STA can confirm the availability of emergency services by observing the value of the Access Network Type, ESR and UESA fields in the Interworking element of any received Beacon or Probe response frame.

V.5.2 Background on emergency call support over IEEE 802.11 infrastructure

Special handling for emergency service calls is required over IEEE 802.11. To use a public hotspot a user will go typically through an authentication process (e.g., EAP-based, or http/https redirect or DNS redirection) before being able to use it for emergency calls.

There is a need to support these emergency services both when the user has a relationship with the IEEE 802.11 network (credentials to access the network) and when it does not have any relationship with the IEEE 802.11 network.

The former case requires no changes to the authentication process—the user, having already been authenticated to and associated with the WLAN, simply dials the emergency number thereby placing the call.

In the latter case, the non-AP STA will be able to gain access to the network without using security credentials and make an emergency call.

Another difficulty is that once the user gains access to the network, there is no mechanism to prioritize their emergency traffic in the IEEE 802.11 MAC over that of other users, even with IEEE 802.11 QoS capability.

Supporting emergency services, such as E911 calling, requires a multi-layer solution with support at various protocol layers. Apart from MAC level access and support for transfer of data between non-AP STA and AP with appropriate QoS at layer 2, there is a clear need, above this layer, to setup the call, conduct call control and management, and use an appropriate audio codec.

One specific example is when a user arrives in a new country and needs to make an emergency call in a public hotspot where there is no prior relationship with the available WLAN network or WLAN hotspot operator.

NOTE—The callback feature, if required in a regulatory domain, is dealt with at a higher layer.

V.5.3 System aspects for emergency call support

An IEEE 802.11 infrastructure by itself cannot ensure that all factors are compatible for an emergency service call to actually take place. The client device may have to register with a call manager (SIP agent or some other signaling endpoint) for the call to be placed successfully. Different signaling systems such as

SIP, H.323, etc., can be deployed for supporting emergency service calling. Higher layers can also verify an emergency service call is being placed so that appropriate level of resources can be granted to the emergency call. Voice endpoints (e.g., non-AP STAs) can use different codecs such as G.711, AMR, and iLBC. All these functionalities are outside the scope of this standard.

IEEE 802.11 can provide priority for emergency traffic both for the initial call establishment and during an ongoing emergency call, which assumes advertisement of this functionality supported in the BSS.

This subclause describes general design assumptions to support emergency services with IEEE 802.11:

a) It is assumed that there is a higher layer (above IEEE 802.11 Layer 2) protocol (or protocol suite) for making emergency calls or using any other emergency services.

b) In order to make the emergency call procedure work properly, the non-AP STA has the following responsibilities:

   1) Recognize the user's request to make an emergency call.

   2) Select an AP that supports QoS and EBR capability.

   3) Non-AP STA will associate with the AP if it is not already done so. In an RSN, if the user does not have valid authentication credentials for network access then non-AP STA can bypass the RSN that will provide access to the network to make emergency calls.

   4) If location information is required in a particular regulatory domain, request location information from the WLAN. If the STA cannot determine its own location by its own means, then Location information should be obtained from the network prior to initiating the emergency call request. There are two methods a non-AP STA can use to obtain location services from the IEEE 802.11 network:

      i) If the non-AP STA can use location information in Geospatial format (i.e., latitude, longitude and altitude), then the RM capability can be used to obtain this information. The AP advertises RM capability in its Beacon management frame (bit1 set to 1 in the Capability information field). In this case, the non-AP STA transmits an LCI Request to the AP using the procedures in 10.11.9.6.

      NOTE—The non-AP STA can receive an LCI Report with the incapable field set. According to the procedures in 10.11.9.6, the non-AP STA can resubmit an LCI Request with a location subject of "remote." If the AP still responds with incapable, then location services are not available from the AP via RM capability.

      ii) If the non-AP STA requires location information in Civic or Geospatial formats, then an AP's wireless network management capability can be used. In this case, an AP advertises its ability to provide its location in with Civic or Geospatial format by setting the Civic Location or Geospatial Location field in the Extended Capabilities element to 1. in the Beacon frame. A non-AP STA requests its location using the procedures in 10.24.7. Unlike an AP providing RM capability, an AP Advertisement location capability will not return an "incapable" response if the non-AP STA requests the "remote" location.

   5) Selects one of possibly several SSPNs advertising support for emergency services and VoIP service.

c) There are two methods described in this annex by which a user lacking security credentials can gain access to the network. The method selected in any particular deployment is at the discretion of the IEEE 802.11 infrastructure provider, SSPN or system administrator as appropriate. The AP and non-AP STA should permit users lacking security credentials to gain access to a network using one of two methods:

   1) Using an emergency services association (see 8.4.2.94) in a BSS configured for RSNA. Using this type of association means the AP and non-AP STA will exchange unprotected frames for emergency service access only during the lifetime of the association. In this situation, cryptographic keys are not exchanged, the IEEE 802.1X uncontrolled port is bypassed without

invoking the IEEE 802.1X state machine. Since protection is used for authenticated STAs, their traffic is protected.

2) Using an SSID configured for Open System authentication. Network elements necessary to complete an emergency call are reachable via this SSID. How to reach these network elements (e.g., a call manager) and which protocol to use (e.g., SIP) are outside the scope of this standard.

d) The AP can separate the backhaul of emergency services traffic from other traffic, typically via a dedicated VLAN.

To ease burden of implementation on the network side, some basic means should exist to allow easy filtering, routing and basic access control of "regular" BSS traffic and emergency-type BSS traffic. This can be assisted by the downloading of emergency call number information, as described in 8.4.4.5.

V.5.4 Description of the Expedited Bandwidth Request element

For access categories configured for mandatory admission control, a non-AP STA requests bandwidth using a TSPEC element in an ADDTS Request frame. The TSPEC Request includes parameters describing the characteristics of the traffic stream, but no information on the use of the traffic stream. The Expedited Bandwidth Request (EBR) element describes the "use" of a traffic stream. To use this element, it is the responsibility of the station to transmit this element in response to certain call signaling messages. How this is done is outside the scope for the interworking service. The following bandwidth uses are provided in the EBR element:

- Emergency call, defined in NENA 08-002 [B52]
- Public first responder (e.g., fire department)
- Private first responder (e.g., enterprise security guard)
- Multi-level precedence and preemption (MLPP)

MLPP services are provided by other voice networking technologies such as 3GPP (see 3GPP TS 22.067 [B3]), H.323 (see ITU-T H4.60.14) and other proprietary signaling protocols. MLPP is used as a subscription service to provide differentiated levels of consumer service; it is also used by military organizations so that commanding officers will not get a network busy signal.

If the AP is provided additional information regarding the nature of the Traffic Stream, it can invoke additional policy that can be configured on the AP to accept the TSPEC request when it would be otherwise denied. Policy configured at AP defines how bandwidth is allocated. Specification of these policies is outside the scope of interworking with external networks. Policy examples include the following:

- No action
- Preemptive action: delete a TS of lower priority if necessary to make room for new TS
- Use capacity allocated for nonvoice services if priority is above a certain value (assuming TSPEC is for AC_VO)
- Interpret MLPP codes as defined 3GPP specification
- Interpret MLPP codes as defined in proprietary specification

V.5.5 Access to emergency services in an RSN

If an infrastructure BSS requires authentication and encryption with RSN, a non-AP STA placing an emergency call associates and authenticate to the network by using an emergency services association (see 8.4.2.94). If the non-AP STA has user credentials that allow it to use a particular network, the non-AP STA can use its credentials to authenticate to the SSPN through the IEEE 802.11 infrastructure.

When a mesh STA has an emergency services association, and it receives a Mesh Peering Open frame that includes the Interworking element, with ASRA bit equal to 1 and UESA bit equal to 0, and the Authenticated Mesh Peering Exchange element (see 8.4.2.120) it allows access to emergency services. If the mesh STA has user credentials that allow the accessing mesh STA to use the mesh network, the mesh STA can use its credentials to authenticate with the accessing mesh peer.

In order to use an emergency services association in an infrastructure BSS, a STA lacking security credentials can associate with a BSS in which emergency services are accessible by including an Interworking Element with the UESA field set to 1 in a (Re)Association Request frame. An AP receiving this type of (re)association request recognizes this as a request for unauthenticated emergency access. The AP can look up the VLAN ID to use with a AAA Server, or it can have an emergency services VLAN configured. The STA can either have, policies configured locally for quality-of-service parameters and network access restrictions, or it can look them up through external policy servers. When a mesh STA has an emergency services association, and it receives a Mesh Peering Open frame, from a mesh STA lacking security credentials, that includes the Interworking element, with ASRA bit equal to 1 and UESA bit equal to 1, and the Mesh Peering Management element (see 8.4.2.104) it allows access to emergency services.

When an emergency services association is used, an infrastructure BSS or an MBSS should be designed to restrict access to emergency services only (or alternatively prioritize the emergency services to the highest level of access). Methods of such restriction are beyond the scope of this standard, but can include an isolated VLAN for emergency services, filtering rules in the AP or network entity (e.g., router) in an external network to limit network access to only network elements involved in emergency calls, and per-session bandwidth control to place an upper limit on resource utilization.

V.6 Peer information

Peer information allows TDLS peer STAs to discover their capabilities. An example of peer information could be expressed in an XML format as follows:

```
<Peer information>
        <mode> tdls </mode>

        <wireless info>
                <mac address> 00:01:02:03:04:05 </mac address>
                <bssid> 00:0a:0b:0c:0d:0e </bssid>
        </wireless info>

        < network info>
                < DHCP> No</DHCP>
                < IP address>192.168.2.1 </IP address>
                < Subnet> 255.255.255.0 </subnet>
                <Hostname> myphone </Hostname>
        </network info>
</Peer information>
```

The minimum required peer information for TDLS is contained in “wireless info.” The “mac address” field contains the MAC address of the transmitting STA and the “bssid” identifies the BSS in which the STA is a member. The “network info” could optionally be used to aid the peer devices in establishing a TDLS link.

Annex W

(informative)

Mesh BSS operation

W.1 Clarification of Mesh Data frame format

The Mesh Data frame consists of MAC Header, Frame Body, and the FCS. The fields in the MAC Header are described in 8.2.3.

In a Mesh Data frame containing a single MSDU, the Mesh Control field is placed immediately before the Data (PDU) in the encrypted Frame Body. The Data (PDU) comprises the LLC/SNAP headers and the higher layer data. This is shown in the example frame format for a CCMP-encrypted Mesh Data frame containing a single MSDU in Figure W-1.

When the Mesh Data frame is fragmented, only the first fragment contains the Mesh Control field.

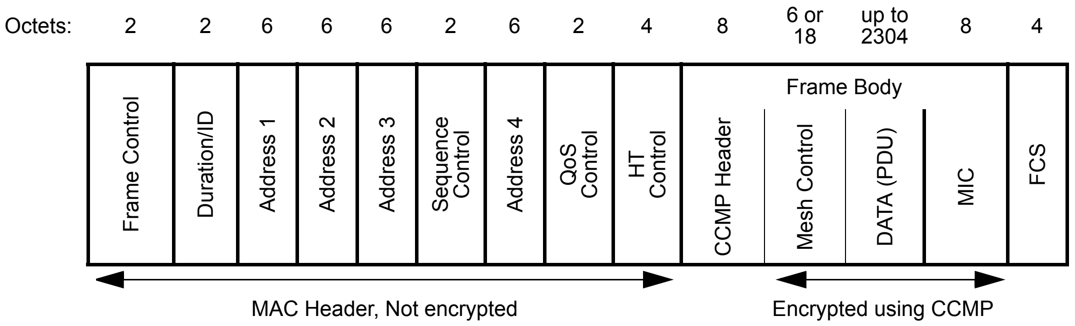


Figure W-1—Format of a CCMP-encrypted Mesh Data frame containing a single MSDU

W.2 Operational considerations for interworking

W.2.1 Formation and maintenance of the IEEE 802.1D spanning tree

No special action is required to support formation of the IEEE 802.1D spanning tree. Spanning tree control messages are typically delivered to bridges in group addressed frames. These messages are data frames from the point of view of the mesh BSS.

W.3 Power save parameters selection

W.3.1 General

Power save mechanisms enable mesh STA operation in Doze state. In Doze state a mesh STA may set the transceivers off. The power save mechanism may adjust the mesh power mode of a mesh STA for various reasons that are beyond the scope of the standard, including but not limited to traffic load or scanning and network maintenance needs.

This annex specifies recommendations on how a mesh STA selects mesh power modes based on traffic load, recommendations on how to do scanning in mesh BSSs that may contain mesh STAs in light or deep sleep mode, and recommended default values for power save related parameters.

W.3.2 Selecting the mesh power mode based on traffic load

A mesh STA may adjust its mesh power mode based on the traffic load or the QoS requirements of the forwarded traffic. If a mesh STA has high traffic load or if a mesh STA is congested, the mesh STA should operate in active mode on the corresponding mesh peering to reduce delays or overhead that may be created by the operation in light or deep sleep mode.

A mesh STA may consider staying in active mode for all peer mesh STAs, even if only a single link is congested. When mesh STA operates in active mode, the mesh STA may receive frames at any time and mesh peer service periods may be triggered on demand, as in case of an AP that receives a trigger frame at times controlled by a non-AP STA.

If traffic load is moderate or low and best effort data is transmitted, a mesh STA may consider staying in light sleep mode for all or some mesh peerings. The mesh STA in light sleep mode for a mesh peering receives beacons from the peer mesh STA with periodic indication of buffered traffic.

If traffic load is low or no traffic is transmitted, a mesh STA may consider staying in deep sleep mode for a mesh peering. The mesh STA does not need to wake up to receive a beacon from the peer mesh STA to which it is in deep sleep mode.

The mesh STA may use deep sleep mode to control the number of times it enters the Awake state to receive Beacon frame from a peer mesh STA. If the mesh STA is in deep sleep mode for all of its mesh peering, the mesh STA needs only to remain in Awake state during its own beacon transmission and mesh awake window.

The mesh STA that forwards real time traffic (AC3 or AC2 with EDCA) should be in active mode for the corresponding link. A mesh STA forwarding real time traffic with EDCA may stay in light or deep sleep mode for the corresponding link, if the mesh STA is capable of initiating mesh peer service periods frequently and the other forwarding peer mesh STAs are in active mode. Poor handling of the mesh power modes may result to delays and inappropriate QoS of the forwarded MSDUs.

W.3.3 Scanning of mesh BSSs

A mesh BSS may have mesh STAs that alternate Awake state and Doze state. With active scanning only devices in Awake state at the transmission time of Probe Request frame may be found. However, with passive scanning also mesh STAs in light or deep sleep mode for a mesh peering can be found.

Since mesh STAs in light or deep sleep mode may transmit beacons at long intervals, a mesh STA seeking for candidate mesh STAs for a new mesh peerings should perform passive scanning relatively for a longer time compared to passive scanning in BSS infrastructure mode. Mesh STAs in light or deep sleep mode with long DTIM interval might not be discovered with short scanning durations. Mesh STAs that operate in light or deep sleep mode for a mesh peering may use a short DTIM interval, if they intend to establish new mesh peerings.

W.3.4 Default parameters

The following are the recommended default values for power save related parameters for mesh STAs:

Table W-1—Default parameters for mesh STAs that intend to operate in light or deep sleep mode for mesh peerings

	For moderate power save	For aggressive power save
Beacon period	200 TU	800 TU
DTIM Period	4	1
Mesh awake window	10 TU	10 TU

A mesh STA that is eager to conserve power and likely to remain in deep sleep mode with most of the mesh peerings should utilize the value from aggressive power save parameters.

Although mesh STAs may utilize individual parameters regardless of the parameters used by neighbor mesh STAs or peer mesh STAs, each implementer should recognize balance between the power save efficiency and delay in the service initiation.

W.3.5 MSDU forwarding in an MBSS containing mesh STAs in light or deep sleep mode

The battery powered mesh STAs should avoid forwarding MSDUs, to avoid power consumption and possible additional delays and inefficiencies that power save mechanisms may cause. If a light or deep sleep mode mesh STA forwards MSDUs, it should select its mesh power mode based on the traffic load and the traffic type as described in W.3.2.

The mesh STAs that desire to save power may select the light or deep sleep mode for a mesh peering as follows:

- mesh STAs that are “mains powered” may apply light or deep sleep mode if they do not have MSDUs to forward
- mesh STAs that are “battery powered” and desire to minimize the power consumption may freely use all mesh power modes

Mesh STAs that are in light or deep sleep mode for a mesh peering may degrade the corresponding link metric value. The use of worse metric values reduces the probability of a link being used for MSDU forwarding. If the mesh STA will operate in active mode for the link in forwarding path, it should apply the link metric value without degradation.

W.3.6 Synchronization maintenance of mesh STAs in deep sleep mode

A mesh STA in deep sleep mode for a mesh peering might not receive Beacon frames from the corresponding peer mesh STA, but the mesh STA is required to maintain synchronization with the neighbor peer mesh STAs. Neighbor offset synchronization method imposes the maintenance of TSF offset values between neighbor peer mesh STAs, which generally requires the reception of Beacon frame or Probe Response frame. The simplest way to maintain the synchronization is that the mesh STA in deep sleep mode for a mesh peering listens to the corresponding peer mesh STA’s Beacon frame for certain periods and check the TSF offset value.

W.4 SIV key wrapping test vector

This test vector is from the appendix of IETF RFC 5297.

```
Input:
-----
Key:
        fffefdfc fbfaf9f8 f7f6f5f4 f3f2f1f0
        f0f1f2f3 f4f5f6f7 f8f9fafb fcfdfeff

Associated Data (ad):
        10111213 14151617 18191a1b 1c1d1e1f
        20212223 24252627

Plaintext:
        11223344 55667788 99aabbcc ddee

S2V-CMAC-AES
------------
CMAC(zero):
        0e04dfaf c1efbf04 01405828 59bf073a

double():
        1c09bf5f 83df7e08 0280b050 b37e0e74

CMAC(ad):
        f1f922b7 f5193ce6 4ff80cb4 7d93f23b

xor:
        edf09de8 76c642ee 4d78bce4 ceedfc4f

double():
        dbe13bd0 ed8c85dc 9af179c9 9ddbf819

pad:
        11223344 55667788 99aabbcc ddee8000

xor:
        cac30894 b8eaf254 035bc205 40357819

CMAC(final):
        85632d07 c6e8f37f 950acd32 0a2ecc93

CTR-AES
-------
CTR:
        85632d07 c6e8f37f 150acd32 0a2ecc93

E(K,CTR):
        51e218d2 c5a2ab8c 4345c4a6 23b2f08f

ciphertext:
        40c02b96 90c4dc04 daef7f6a fe5c

output
------
```

```
    IV || C:
            85632d07 c6e8f37f 950acd32 0a2ecc93
            40c02b96 90c4dc04 daef7f6a fe5c
```

W.5 Airtime link metric usage example

The airtime cost constants (Table 13-4) and estimates of the average data rate and frame error rate will vary from one implementation and configuration of the IEEE 802.11 PHY and MAC to the other. While no mechanism is defined to measure the average data rate and the frame error rate, it is expected that numeric values will not exhibit large non-monotonic variations in amplitude over the lifetime of a path. Unstable measurements may cause path selection instabilities.

An example of an airtime link metric is provided to illustrate how it may be calculated.

Assume a DSSS PHY with an average data rate of 1 Mb/s to a given neighbor and a frame size of 8192 bits.

The overhead O for the data frame is comprised of the PLCP preamble (144 µs) and the PLCP header (48 µs). The payload Bit is 8192 bits at an r of 1 Mb/s, or 8192 µs. The data transmission time is therefore 8416 µs. Other transmission times for different frame types are calculated in the same way (based on their size, rate, and overhead).

If RTS/CTS is used, the data transmission time (including PLCP preamble and header) is 8416 µs, the RTS transmission time (including PLCP preamble and header) is 352 µs, the CTS transmission time (including PLCP preamble and header) is 304 µs, the ACK transmission time (including PLCP preamble and header) is 304 µs and the interframe spacing overhead is 390 µs. The total time, including overhead, is 9766 µs.

This airtime and overhead value is converted to units of 0.01 TU (10.24 µs), i.e., 953.71 (rounded to 954).

If the frame error rate to the neighbor is 0%, the metric is 954. If the frame error rate is 80%, the metric is 4769 (i.e., 953.71/(1–0.8), rounded).

W.6 Generation of proactive PREPs in proactive PREQ mechanism of HWMP

W.6.1 General

In the proactive PREQ mechanism of HWMP, the generation of a proactive PREP in response to the receipt of a proactive PREQ depends on the value of the Proactive PREP subfield in the received proactive PREQ (see 13.10.4.2). Furthermore, if the Proactive PREP subfield is 0, the mesh STA may respond with the proactive PREP in case it needs a bidirectional path between the root mesh STA and itself. This is usually the case if the mesh STA has data to be sent to the root mesh STA. This clause provides a unified mechanism that controls the generation of proactive PREPs in the proactive PREQ mechanism of HWMP.

A proactive PREQ is defined by all of the following:

- The Target Address is set to all ones; and
- The TO subfield in the Per Target Flags field is 1.

W.6.2 Additions to forwarding information

The forwarding information to a root mesh STA contains two additional Boolean information fields. The Proactive PREP field indicates whether the mesh STA will generate a proactive PREP to the root mesh STA in response to a proactive PREQ (Proactive PREP = 1) or not (Proactive PREP = 0). The Proactive PREP Sent field indicates whether the mesh STA has sent a proactive PREP to the root mesh STA (Proactive PREP Sent = 1) or not (Proactive PREP Sent = 0). Both fields are initialized with 0.

W.6.3 Actions when sending data frames as source mesh STA

If a mesh STA receives proactive PREQs with the Proactive PREP subfield set to 0, the recipient mesh STA sends a proactive PREP before sending a data frame as source mesh STA only if the mesh STA has data to send to the root mesh STA, which requires establishing a bidirectional path with the root mesh STA and the field Proactive PREP Sent of the forwarding information to the root mesh STA is not set (=0).

If the mesh STA sends a data frame as source mesh STA to the root mesh STA, the mesh STA sets the field Proactive PREP of the forwarding information to 1.

W.6.4 Actions on receipt of proactive PREQ

If the mesh STA receives a proactive PREQ, the field Proactive PREP Sent of the forwarding information to the root mesh STA is set to 0. If the field Proactive PREP of the forwarding information to the root mesh STA is 1, the mesh STA generates a proactive PREP to the root mesh STA (see 13.10.10.3 Case D), sets the field Proactive PREP of the forwarding information to 0, and sets the field Proactive PREP Sent of the forwarding information to 1.

If the Proactive PREP subfield of the Flags field of the received proactive PREQ is 1, the mesh STA sets the field Proactive PREP of the forwarding information to the root mesh STA to 1.

W.6.5 Generation of proactive PREPs

A mesh STA will generate a proactive PREP according to 13.10.10.3 Case D if one of the following applies:

— [The mesh STA has received a proactive PREQ] AND [in the forwarding information to the root mesh STA of the proactive PREQ, field Proactive PREP is set to 1 AND field Proactive PREP Sent is set to 0].

— [The mesh STA has data to send to the root mesh STA, which requires establishing a bidirectional path with the root mesh STA] AND [the field Proactive PREP Sent of the forwarding information to the root mesh STA is not set (=0)].

If the mesh STA generates a proactive PREP to the root mesh STA, the field Proactive PREP Sent of the forwarding information is set to 1 and the field Proactive PREP of the forwarding information is set to 0.

W.7 Generation of PREQs in proactive RANN mechanism of HWMP

W.7.1 General

In the proactive RANN mechanism of HWMP, the generation of a PREQ for root path confirmation in response to the receipt of a RANN depends on the path metric from the mesh STA to the root mesh STA as computed by the RANN propagation. However, the RANN mechanism does not setup the necessary forwarding information. This is done with individually addressed PREQs. This clause provides further details to the generation of individually addressed PREQs in the proactive RANN mechanism of HWMP.

An individually addressed PREQ is defined by the following:

— The Addressing Mode subfield in the Flags field is 1 (individually addressed).

W.7.2 Additions to forwarding information

The forwarding information to a root mesh STA contains an additional address field. The RANN Sender Address field contains the MAC address of the neighbor peer mesh STA that has sent the RANN with the best metric.

W.7.3 Actions when sending data frames as source mesh STA

If the mesh STA sends a data frame as source mesh STA to the root mesh STA, the mesh STA uses the forwarding information to the root mesh STA. The RANN Sender Address is not used for forwarding Mesh Data frames to the root mesh STA.

W.7.4 Actions on receipt of proactive RANN

If the mesh STA receives a proactive RANN, the field RANN Sender Address of the forwarding information to the root mesh STA is set to the sender of the RANN element if the metric to the root mesh STA is better than the path metric of the existing mesh path in the forwarding information.

In this case, the mesh STA generates an individually addressed PREQ to the root mesh STA—see 13.10.10.3 Case D (Root Path Confirmation (Original Transmission)). This individually addressed PREQ is not sent according to the general forwarding information. Instead, it is sent to the neighbor peer mesh STA indicated in the RANN Sender Address field.

Since all mesh STAs between the mesh STA and the root mesh STA have already a value in the RANN Sender Address field, the individually address PREQ will eventually reach the root mesh STA. Since the TO subfield in the Per Target Flags field is 1, only the root mesh STA will generate a PREQ as reply to the individually addressed PREQ.

The individually addressed PREQs will setup forwarding information (that can be used for forwarding Mesh Data frames) from the root mesh STA towards the mesh STA that initiated the root path confirmation in all intermediate mesh STAs according to the normal HWMP procedures.

Note, since the PREQ is sent in an individually addressed frame, every sender of the individually addressed PREQ will be able to determine whether the next mesh STA towards the root mesh STA has received the PREQ.

The root mesh STA will generate a PREP according to the normal HWMP procedures—see 13.10.10.3 Case A (Original Transmission).

W.7.5 Actions on receipt of PREP

The PREP generated by the root mesh STA will be forwarded on the mesh path from the root mesh STA to the mesh STA as setup by the individually addressed PREQ. The PREP will be propagated and will eventually reach the mesh STA. The PREP will setup forwarding information (that can be used for forwarding Mesh Data frames) towards the root mesh STA in the mesh STA that initiated the root path confirmation and in all intermediate mesh STAs on this path according to the normal HWMP procedures.

After establishing the forwarding information (mesh path) towards the root mesh STA, the path to the root mesh STA is updated to the better one.

Note that since the PREP is sent in an individually addressed frame, every sender of the PREP will be able to determine whether the next mesh STA has received the PREP.