IEEE STANDARDS ASSOCIATION IEEE

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

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

Amendment 10: Mesh Networking

IEEE Computer Society

Sponsored by the
LAN/MAN Standards Committee

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

10 September 2011

IEEE Std 802.11s™-2011
(Amendment to IEEE Std 802.11™-2007
as amended by IEEE Std 802.11k™-2008,
IEEE Std 802.11r™-2008, IEEE Std 802.11y™-2008,
IEEE Std 802.11w™-2009, IEEE Std 802.11n™-2009,
IEEE Std 802.11p™-2010, IEEE Std 802.11z™-2010,
IEEE Std 802.11v™-2011, and IEEE Std 802.11u™-2011)

IEEE Std 802.11s™-2011
(Amendment to IEEE Std 802.11™-2007
as amended by IEEE Std 802.11k™-2008, IEEE Std 802.11r™-2008,
IEEE Std 802.11y™-2008, IEEE Std 802.11w™-2009, IEEE Std 802.11n™-2009,
IEEE Std 802.11p™-2010, IEEE Std 802.11z™-2010,
IEEE Std 802.11v™-2011, and IEEE Std 802.11u™-2011)

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

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

Amendment 10: Mesh Networking

Sponsor

LAN/MAN Standards Committee
of the
IEEE Computer Society

Approved 10 September 2011
IEEE-SA Standards Board

Abstract: This amendment describes protocols for IEEE 802.11 stations to form self-configuring multi-hop networks that support both broadcast/multicast and unicast data delivery.

Keywords: forwarding, IEEE 802.11s, medium access control, mesh, multi-hop, path selection, wireless LAN, zero-knowledge proof

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

PDF: ISBN 978-0-7381-6731-2 STD97150
Print: ISBN 978-0-7381-6732-9 STDPD97150

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

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

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

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

In publishing and making this document available, the IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Nor is the IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing this, and any other IEEE Standards document, should rely upon the advice of a competent professional in determining the exercise of reasonable care in any given circumstances.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests.For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration. A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal interpretation of the IEEE.At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position, explanation, or interpretation of the IEEE. Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Recommendations to change the status of a stabilized standard should include a rationale as to why a revision or withdrawal is required.

Comments and recommendations on standards, and requests for interpretations should be addressed to:

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

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

Introduction

This introduction is not part of IEEE Std 802.11s-2011, IEEE Standard for Information Technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 11: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications—Amendment 10: Mesh Networking.

This amendment specifies enhancements to support mesh networking. The networks described in this amendment make use of layer-2 mesh path selection and forwarding (that is, a wireless mesh network that performs routing at the link layer). Wireless mesh networks have advantageous properties in terms of robustness, range extension and density, but also have potential challenges such as power consumption and security. This amendment is specifically designed to address these challenges.

Notice to users

Laws and regulations

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

Copyrights

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

Updating of IEEE documents

Users of IEEE standards should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect. In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit the IEEE Standards Association website at http://ieeexplore.ieee.org/xpl/standards.jsp, or contact the IEEE at the address listed previously.

For more information about the IEEE Standards Association or the IEEE standards development process, visit the IEEE-SA website at http://standards.ieee.org.

Errata

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

Interpretations

Current interpretations can be accessed at the following URL: http://standards.ieee.org/reading/ieee/interp/index.html.

Patents

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims or determining whether any licensing terms or conditions are reasonable or non-discriminatory. Further information may be obtained from the IEEE Standards Association.

Participants

At the time this amendment was submitted to the IEEE-SA for approval, the IEEE 802.11 Working Group had the following membership:

Bruce Kraemer, *Chair*
Jon Rosdahl and **Adrian Stephens**, *Vice-chairs*
Stephen McCann, *Secretary*

Osama S. Aboul-Magd
Santosh P. Abraham
Tomoko Adachi
Carlos H. Aldana
Gary Anwyl
Lee R. Armstrong
Alex Ashley
Malik Audeh
Geert A. Awater
David Bagby
Michael Bahr
Fan Bai
Gabor Bajko
Raja Banerjea
Kaberi Banerjee
John R. Barr
Gal Basson
Tuncer Baykas
John L. Benko
Mathilde Benveniste
Daniel Borges
Anthony Braskich
Joseph Brennan
Walter Buga
George Bumiller
Nancy Cam-Winget
Necati Canpolat
Javier Cardona
Philippe Chambelin
Douglas S. Chan
Clint F. Chaplin
Jiunn-Tsair Chen
Lidong Chen
Minho Cheong
Woong Cho
Jee-Yon Choi
Nakjung Choi
Liwen Chu
Terry L. Cole
Charles I. Cook
Carlos Cordeiro
Xavier Perez Costa
David E. Cypher
Marc De Courville
Rolf J. de Vegt
Theodorus Denteneer
Jeremy deVries
Susan Dickey
John Dorsey
Roger P. Durand
Srinivasa Duvvuri
Donald E. Eastlake, III
Peter Ecclesine
Stephen P. Emeott
Marc Emmelmann
Darwin Engwer
Vinko Erceg
Stefan Fechtel
Matthew J. Fischer
Wayne K. Fisher
Wen Gao
Matthew Gast
James P. K. Gilb
Jeffrey Gilbert
Reinhard Gloger
Michelle Gong
David Goodall
Sudheer A. Grandhi
Mark Grodzinsky
Jianlin Guo
Mark Hamilton
Christopher J. Hansen
Hiroshi Harada
Dan N. Harkins
Brian D. Hart
Chris Hartman
Amer A. Hassan
Vegard Hassel
Robert F. Heile
Guido R. Hiertz
Garth D. Hillman
Seungeun Hong
Naoki Honma
Wendong Hu
Robert Y. Huang
Tian-Wei Huang
David Hunter
Akio Iso
Wynona Jacobs
Hongseok Jeon
Yeonkwon Jeong
Lusheng Ji
Daniel Jiang
Sunggeun Jin
V. K. Jones
Padam Kafle
Carl W. Kain
Naveen K. Kakani
Shuzo Kato
Douglas Kavner
Richard H. Kennedy
John Kenney
Stuart J. Kerry
Joonsuk Kim
Kyeongpyo Kim
Yongsun Kim
Youngsoo Kim
Yunjoo Kim
Jarkko Kneckt
Mark M. Kobayashi
Fumihide Kojima
Tom Kolze
Thomas M. Kurihara
Joseph Kwak
Hyoungjin Kwon
Ismail Lakkis
Paul Lambert
Zhou Lan
Jeremy A. Landt
Joseph P. Lauer
Wooyong Lee
Yuro Lee
Sheung Li
Hang Liu
Pei Liu
Peter Loc
Hui-Ling Lou
Bradley Lynch
Jakub Majkowski
Alastair Malarky
Jouni K. Malinen
Alexander Maltsev
Hiroshi Mano
Bill Marshall
Roman M. Maslennikov
Justin P. McNew
Sven Mesecke
Robert R. Miller
Michael Montemurro
Rajendra T. Moorti
Hitoshi Morioka
Yuichi Morioka
Daniel Camps Mur
Peter Murray
Andrew Myles
Yukimasa Nagai
Kengo Nagata
Hiroki Nakano
Sai Shankar Nandagopalan
Chiu Ngo
Paul Nikolich
Eero Nikula
Richard H. Noens
Jisung Oh
Jong-Ee Oh
Youko Omori
Satoshi Oyama

Richard H. Paine
Arul Durai Murugan Palanivelu
Changmin Park
Minyoung Park
Vijaykumar Patel
Bemini Hennadige Peiris
Eldad Perahia
James E. Petranovich
Albert Petrick
John Petro
Vishakan Ponnampalam
James D. Portaro
Henry S. Ptasinski
Rene Purnadi
Ivan Pustogarov
Emily H. Qi
Huyu Qu
Jim E. Raab
Mohammad Rahman
Vinuth Rai
Ali Raissinia
Harish Ramamurthy
Stephen G. Rayment
Ivan Reede
Alex Reznik
Randal Roebuck
Richard Roy
Alexander Safonov
Kazuyuki Sakoda
Hemanth Sampath

Donald Schultz
Jean Schwoerer
Yongho Seok
Huairong Shao
Stephen J. Shellhammer
Ian Sherlock
Kai Shi
Francois Simon
Graham Kenneth Smith
Matt Smith
Kapil Sood
Vinay Sridhara
Robert Stacey
Dorothy Stanley
David S. Stephenson
Carl R. Stevenson
John Stine
Guenael T. Strutt
Chin-Sean Sum
Arash Tabibiazar
Eiji Takagi
Mineo Takai
Yasushi Takatori
Teik-Kheong Tan
Allan Thomson
Jerry Thrasher
Eric Tokubo
Ichihiko Toyoda
Jason Trachewsky
Solomon B. Trainin

Jean Tsao
Masahiro Umehira
Richard D. J. Van Nee
Allert Van Zelst
Prabodh Varshney
Ganesh Venkatesan
Dalton T. Victor
George A. Vlantis
Jesse R. Walker
Chao-Chun Wang
Junyi Wang
Qi Wang
Craig D. Warren
Fujio Watanabe
Menzo M. Wentink
Frank Whetten
James Worsham
Harry R. Worstell
Fonchi Wu
Takeshi Yamamoto
James Yee
Peter Yee
Su Khiong Yong
Seiji Yoshida
Christopher Young
Artur Zaks
Hongyuan Zhang
Shiwei Zhao
Chunhui Zhu

The following were officers of Task Group s:

Dee Denteneer, *Chair*
Guido R. Hiertz*, Vice Chair*
(**Stuart J. Kerry,** *Pro-tem Chair*)
(**Donald E. Eastlake 3rd***, Former Chair*)

Guenael Strutt, *Secretary*
(**Stephen Rayment**, *Former Secretary*)

Kazuyuki Sakoda, *Technical Editor*
(**Anthony F. Maida**, *Former Technical Editor*)
(**Avinash Joshi***, Former Pro-tem Technical Editor*)
(**W. Steven Conner**, *Former Technical Editor*)

Major contributions were received from the following individuals:

Osama S. Aboul-Magd
Santosh P. Abraham
Jonathan Agre
Hidenori Aoki
Malik Audeh
Michael Bahr
Dennis Baker
Mathilde Benveniste
Nehru Bhandaru
Tony Braskich
Javier Cardona
Narasimha Chari
Ray-Guang Cheng
Liwen Chu
W. Steven Conner
Dee Denteneer

Srinivasa Duvvurri
Donald E. Eastlake, III
Steohen P. Emeott
Stefano M. Faccin
Don Fedyk
Atsushi Fujiwara
Sandesh Goel
Michelle Gong
Hrishikesh Gossain
David Gurevich
Jyunichiro Hagiwara
Liu Hang
Susan Hares
Dan N. Harkins
Vann Hasty
Jim Hauser

Kevin Hayes
Guido Hiertz
David E. Hudak
David Hunter
Jorjeta Jetcheva
Avinash Joshi
Tyan-Shu Jou
Thomas Junge
Youiti Kado
Shantanu Kangude
Mika Kasslin
Stuart J. Kerry
Andrew Khieu
Joseph Kim
Jarkko Kneckt
Jan Kruys

Ted Kuo
Jua-Ru Li
Catherine Livet
Andrey Lyakhov
Anthony Maida
Vijay Mantri
Janne Marin
Yoichi Matsumoto
Sebastian Max
Stephen McCann
Minseok Oh
Youko Omori
John Petro
Rene Purnadi
Huyu Qu
Shah Rahman
Harish Ramamurthy
Stephen G. Rayment
Edward Reuss
Vincent Roy
Marian Rudolf
Bahareh Sadeghi
Alexander Safonov
Shin Saito
Kazuyuki Sakoda
Yongho Seok
Oyunchimeg Shagdar
Ming Sheu
Ashish Shukla
D. J. Shyy
Tricci So
Peter Stanforth
Lothar Stibor
Guenael T. Strutt
Sheng Sun
Rakesh Taori
John Tomici
George A. Vlantis
Jesse R. Walker
Carl Wijting
Jack Winters
James Woodyatt
Zhen Xie
Akiyoshi Yagi
Akira Yamada
Lily Yang
Jen-Shun Yang
Zhonghui Yao
Peter Yee
Yunpeng Zang
Amy Zhang
Bing Zhang
Meiyuan Zhao
Juan Carlos Zuniga

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

Tomoko Adachi
Roberto Aiello
Thomas Alexander
Richard Alfvin
Mark Anderson
Butch Anton
Lee Armstrong
Torrey Atcitty
Taehan Bae
Michael Bahr
Raja Banerjea
Hugh Barrass
Klaus Bender
Mathilde Benveniste
Harry Bims
Gennaro Boggia
Nancy Bravin
William Byrd
Ruben Salazar Cardozo
James Carlo
Youngbin Chang
Clint Chaplin
Yi-Ming Chen
Ray-Guang Cheng
Keith Chow
Michael Coddington
Charles Cook
Todor Cooklev
Joseph Decuir
Theodorus Denteneer
Wael Diab
Russell Dietz
Thomas Dineen
Roger Durand
Sourav Dutta
Donald E. Eastlake, III
Peter Ecclesine
Richard Eckard
Marc Emmelmann
Joseph Epstein
Shulan Feng
C. Fitzgerald
Prince Francis
Devon Gayle
Pieter-Paul Giesberts
Gregory Gillooly
Stephen Glass
Reinhard Gloger
Patrick Gonia
Sudheer Grandhi
Randall Groves
Michael Gundlach
C. Guy
Rainer Hach
David Halasz
Christopher Hansen
Robert F. Heile
Marco Hernandez
Guido Hiertz
Ronald Hochnadel
Oliver Hoffmann
Russell Housley
Chun-Yen Hsu
David Hunter
Yasuhiko Inoue
Akio Iso
Atsushi Ito
Tetsuya Ito
Raj Jain
Junghoon Jee
Anthony Jeffree
Bobby Jose
Tal Kaitz
Naveen Kakani
Shinkyo Kaku
Masahiko Kaneko
Tae-Gyu Kang
Piotr Karocki
Stuart J. Kerry
Yongbum Kim
Youhan Kim
Jarkko Kneckt
Bruce Kraemer
Joseph Kubler
Thomas Kurihara
Joseph Kwak
Paul Lambert
David Landry
Jeremy Landt
Yeou Song Lee
Michael Lerer
Daniel Levesque
Zexian Li
Jan-Ray Liao
Arthur Light
Lu Liru
William Lumpkins
Greg Luri
Kaiying Lv
Bradley Lynch
Chris Lyttle
Elvis Maculuba
Jouni Malinen
Stephen McCann
Michael McInnis
Steven Methley
Jochen Miroll
Emmanuel Monnerie
Michael Montemurro
Matthew Mora
Jose Morales
Ronald Murias
Rick Murphy
Peter Murray
Andrew Myles
Michael S. Newman
Charles Ngethe
Paul Nikolich
Satoshi Obara
Knut Odman

Robert O’Hara
Chris Osterloh
Satoshi Oyama
James Petranovich
Subburajan Ponnuswamy
Venkatesha Prasad
Michael Probasco
Henry Ptasinski
Emily Qi
Sridhar Rajagopal
Jayaram Ramasastry
Maximilian Riegel
Robert Robinson
Randal Roebuck
Jon Rosdahl
Herbert Ruck
Charles Rush
Rashid Saeed
Randall Safier
Kazuyuki Sakoda
Naotaka Sato

Ryo Sawai
Bartien Sayogo
Cristina Seibert
Yongho Seok
Suman Sharma
Yang Shi
Shusaku Shimada
Gil Shultz
Jae-Hyung Song
Kapil Sood
Amjad Soomro
Robert Soranno
Manikantan Srinivasan
Dorothy Stanley
Kenneth Stanwood
Thomas Starai
Adrian Stephens
Rene Struik
Walter Struppler
Guenael Strutt
Mark Sturza

Bo Sun
Jun Ichi Takada
David Tepen
Solomon Trainin
Mark-Rene Uchida
Prabodh Varshney
Ganesh Venkatesan
Bhupender Virk
George Vlantis
Hongbo Wang
Stanley Wang
Stephen Webb
Hung-Yu Wei
Menzo Wentink
Ludwig Winkel
James Worsham
Harry Worstell
Yunsong Yang
James Yee
Tan Pek Yew
Oren Yuen
Juan Zuniga

When the IEEE-SA Standards Board approved this standard on 10 September 2011, it had the following membership:

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

Masayuki Ariyoshi
William Bartley
Ted Burse
Clint Chaplin
Wael Diab
Jean-Philippe Faure
Alexander Gelman
Paul Houzé

Jim Hughes
Joseph L. Koepfinger*
David J. Law
Thomas Lee
Hung Ling
Oleg Logvinov
Ted Olsen

Gary Robinson
Jon Walter Rosdahl
Sam Sciacca
Mike Seavey
Curtis Siller
Phil Winston
Howard L. Wolfman
Don Wright

*Member Emeritus

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

Satish Aggarwal, NRC Representative
Richard DeBlasio, DOE Representative
Michael Janezic, NIST Representative

Catherine Berger
IEEE Project Editor

Kathryn Bennett
IEEE Standards Program Manager, Technical Program Development

Contents

List of figures

List of tables

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

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

Amendment 10: Mesh Networking

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

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

(This amendment specifies enhancements to IEEE Std 802.11™. It is based on IEEE Std 802.11™-2007, as amended by IEEE Std 802.11k™-2008, IEEE Std 802.11r™-2008, IEEE Std 802.11y™-2008, IEEE Std 802.11w™-2009, IEEE Std 802.11n™-2009, IEEE Std 802.11p™-2010, IEEE Std 802.11z™-2010, IEEE Std 802.11v™-2011, and IEEE Std 802.11u™-2011.)

NOTE—The editing instructions are shown in ***bold italic***. Four editing instructions are used: change, delete, insert, and replace. ***Change*** is used to make corrections in existing text or tables. The editing instruction specifies the location of the change and describes what is being changed by using ~~strikethrough~~ (to remove old material) and underscore (to add new material). ***Delete*** removes existing material. ***Insert*** adds new material without disturbing the existing material. Insertions may require renumbering. If so, renumbering instructions are given in the editing instruction. ***Replace*** is used to make changes in figures or equations by removing the existing figure or equation and replacing it with a new one. Editing instructions, change markings, and this NOTE will not be carried over into future editions because the changes will be incorporated into the base standard.[1]

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

1. Overview

1.2 Purpose

Insert the following dash list item to the end of 1.2:

— Defines the MAC procedures that are necessary for wireless multi-hop communication to support wireless LAN mesh topologies.

2. Normative references

Insert the following new normative references:

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

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

3. Definitions

Insert the following new subclause title:

3.1 General definitions

Change the definition for basic service set (BSS) as follows:

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

Change the definition for distribution system service (DSS) as follows:

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

NOTE—DSSs are provided between pairs of IEEE 802.11 MACs.

Change the definition for distribution system medium (DSM) as follows:

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

Change the definition for wireless distribution system (WDS) as follows:

wireless distribution system (WDS): ~~A mechanism for wireless communication using a four address frame format specified in this standard. This standard describes such a frame format, but does not describe how such a mechanism or frame format would be used.~~ Often used as a vernacular term for a mechanism for wireless communication among non-mesh stations (STAs) using a four address frame format.

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

Insert the following new definitions in alphabetical order:

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

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

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

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

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

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

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

mesh basic service set (MBSS): A basic service set (BSS) that forms a self-contained network of mesh stations (STAs). An MBSS contains zero or more mesh gates.

mesh facility: The set of enhanced functions, channel access rules, frame formats, mutual authentication methods, and managed objects used to provide data transfer among autonomously operating stations (STAs) that may not be in direct communication with each other over a single instance of the wireless medium.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

neighbor station (STA): A STA that is in direct communication range over a single instance of the wireless medium.

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

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

password: A shared, secret, and potentially low-entropy word, phrase, code, or key used as a credential for authentication purposes.

NOTE—The method of distribution of a password to the units in the system is outside the scope of this standard.

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

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

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

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

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

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

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

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

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

Insert the following new subclause and new definitions:

3.2 Definitions specific to IEEE Std 802.11

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

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

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

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

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

Mesh Data frame: A unicast Data frame with both the FromDS and ToDS bits set to 1 and that is transmitted from a mesh station (STA) to a peer mesh STA, or a group addressed Data frame that has FromDS set to 1 and ToDS set to 0 that is transmitted by a mesh STA.

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

4. Abbreviations and acronyms

Insert the following new acronym in alphabetical order:

AMPE	authenticated mesh peering exchange
FSM	finite state machine
GANN	gate announcement
HWMP	hybrid wireless mesh protocol
MAF	MCCA access fraction
MBCA	mesh beacon collision avoidance
MBSS	mesh basic service set
MCCA	MCF controlled channel access
MCCAOP	MCF controlled channel access opportunity
MCF	mesh coordination function
MGTK	mesh group temporal key
MICE	MIC element
MPM	mesh peering management
MPSP	mesh peer service period
MTK	mesh temporal key
PERR	path error
PREP	path reply
PREQ	path request
PXU	proxy update
PXUC	proxy update confirm
RANN	root announcement
RAV	resource allocation vector
RSPI	receiver service period initiated
SAE	simultaneous authentication of equals
TTL	time to live

5. General description

5.2 Components of the IEEE 802.11 architecture

5.2.2 STA membership in a BSS is dynamic

Change the first paragraph in 5.2.2 as follows:

A STA's membership in a BSS is dynamic (STAs turn on, turn off, come within range, and go out of range). To become a member of an infrastructure BSS or an IBSS, a STA joins the BSS using the synchronization procedure described in 11.1.3.4. To start a new mesh BSS or to become a member of a mesh BSS, a STA starts beaconing and the synchronization maintenance procedure described in 11C.12. To access all the services of an infrastructure BSS, a STA becomes "associated." These associations are dynamic and involve the use of the distribution system service (DSS), which is described in 5.3.2. A mesh STA does not become associated as there is no central entity in a mesh BSS (MBSS). Instead, a mesh STA peers with other mesh STAs and thereby they form the MBSS.

5.2.3 Distribution system (DS) concepts

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

Change the first paragraph in 5.2.3.1 as follows:

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

Insert the following paragraph after the second paragraph in 5.2.3.1:

Owing to its distributed nature, a mesh BSS (MBSS) has no central entity like the AP of an infrastructure BSS. Instead, an MBSS forms a single set of independent mesh STAs. This set is undivided and cannot be further unified. Therefore, the ESS concept does not apply to the MBSS.

5.2.6 QoS BSS: The QoS network

Change the first paragraph in 5.2.6 as follows:

The IEEE 802.11 QoS facility provides MAC enhancements to support LAN applications with QoS requirements. The QoS enhancements are available to QoS STAs associated with a QoS access point in a QoS BSS. A subset of the QoS enhancements is available for use between STAs that are members of the same QoS IBSS. Similarly, a subset of the QoS enhancements is available for use between neighbor peer mesh STAs. A mesh BSS is one type of QoS BSS and it is described in 5.2.14. Because a non-mesh QoS STA implements a superset of STA functionality, as defined in this standard, the STA might ~~may~~ associate with a non-QoS access point in a non-QoS BSS, to provide non-QoS MAC data service when there is no QoS BSS with which to associate. As a mesh STA does not implement the necessary service, the mesh STA does not associate with any access point.

Change the third paragraph in 5.2.6 as follows:

For infrastructure BSS and IBSS, t~~T~~his standard provides two mechanisms for the support of applications with QoS requirements.

5.2.9 High-throughput (HT) station (STA)

Insert the following text to the end of the first paragraph in 5.2.9:

Similarly, a subset of the HT features is available for use between two HT STAs that have established mesh peering (see 7.3.2.56 for details).

Insert the following new subclause after 5.2.13:

5.2.14 Mesh BSS: IEEE 802.11 wireless mesh network

5.2.14.1 General

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

5.2.14.2 Overview of the mesh BSS

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

STAs in a mesh BSS might be sources, sinks, or propagators of traffic; some mesh STAs might only propagate traffic for other STAs. As described in 5.2.14.4, a mesh BSS might have interfaces to external networks and can be utilized as a backhaul for infrastructure BSSs.

Within a mesh BSS, STAs utilize the mesh coordination function (MCF) to access the channel. MCF is based on the core QoS facilities specified in 5.2.6, and a mesh BSS is categorized as one type of QoS BSS. MCF is described in 9.9a.

5.2.14.3 Mesh STA

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

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

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

- HCCA
- Traffic specifications (TSPECs)
- Traffic stream (TS) management
- Admission control

- Automatic power save delivery (APSD)
- Direct-link setup (DLS)
- Tunneled direct-link setup (TDLS)

5.2.14.4 IEEE 802.11 components and mesh BSS

Example mesh and infrastructure BSSs are illustrated in Figure 5-6b. Only mesh STAs participate in mesh functionalities such as formation of the mesh BSS, path selection, and forwarding. Accordingly, a mesh STA is not a member of an IBSS or an infrastructure BSS. Consequently, mesh STAs do not communicate with non-mesh STAs.

However instead of existing independently, an MBSS might also access the distribution system (DS). The MBSS interconnects with other BSSs through the DS. Then, mesh STAs can communicate with non-mesh STAs. Therefore, a logical architectural component is introduced in order to integrate the MBSS with the DS—the mesh gate. Data move between an MBSS and the DS via one or more mesh gates. Thus, the mesh gate is the logical point at which MSDUs from an MBSS enter the IEEE 802.11 DS. Once an MBSS contains a mesh gate that connects it to the IEEE 802.11 DS, the MBSS can be integrated with other infrastructure BSSs too, given that their APs connect to the same DS. Several mesh gates are shown in Figure 5-6b connecting different MBSSs to the DS.

When an MBSS accesses the IEEE 802.11 DS through its mesh gate, the MBSS can be integrated with a non-IEEE-802.11 LAN. To integrate the IEEE 802.11 DS to which this MBSS connects, the DS needs to contain a portal. See 5.2.5. Consequently, mesh gate and portal are different entities. The portal integrates the IEEE 802.11 architecture with a non-IEEE-802.11 LAN (e.g., a traditional wired LAN), whereas the mesh gate integrates the MBSS with the IEEE 802.11 DS.

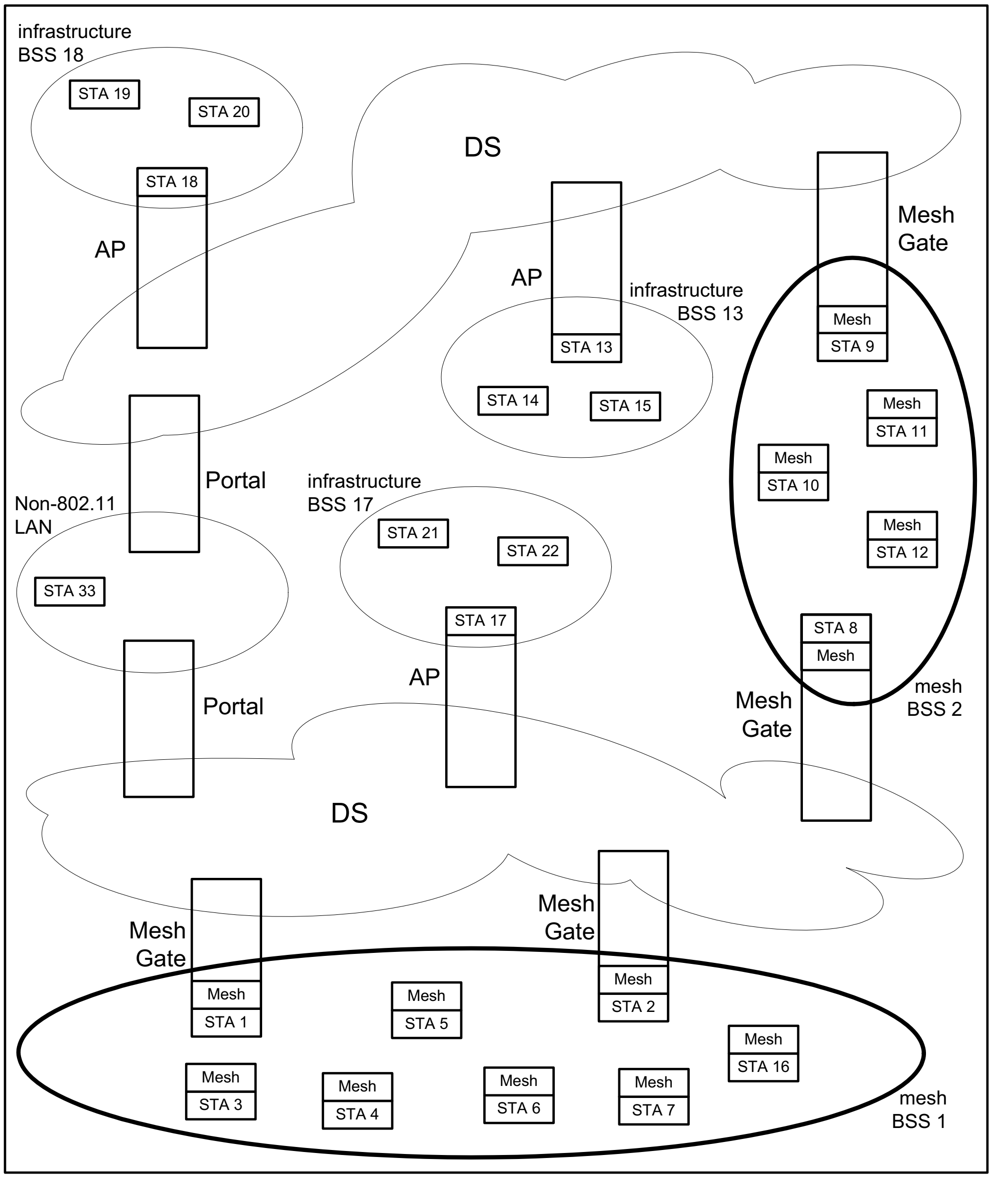


Figure 5-6b—Example MBSS containing mesh STAs, mesh gates, APs, and portals

It is possible for one device to offer any combination of the functions of an AP, a portal, and a mesh gate, see 11C.10.5. An example device combining the functions of an AP and a mesh gate is shown in Figure 5-6c. The implementation of such collocated entities is beyond the scope of this standard. The configuration of a mesh gate that is collocated with an access point allows the utilization of the mesh BSS as a distribution system medium. In this case, two different entities (mesh STA and access point) exist in the collocated device and the mesh BSS is hidden to STAs that associate to the access point. The mesh STA collocation is outlined in 11C.10.5, which states that the usage of a distinct MAC address for each collocated STA avoids ambiguities.

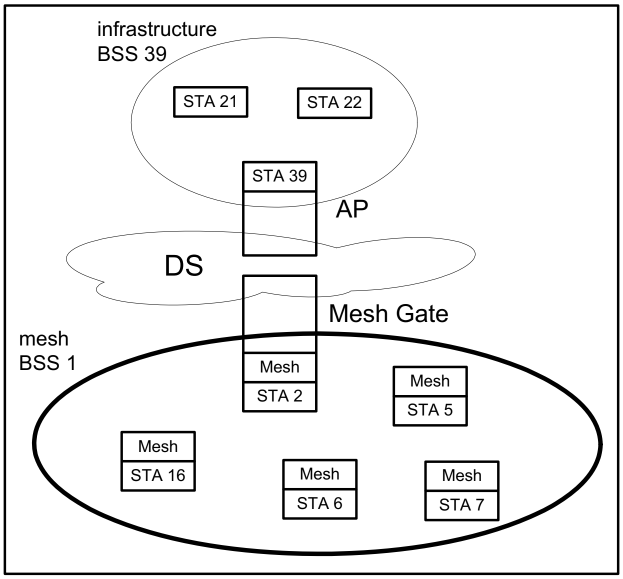


Figure 5-6c—Example device consisting of mesh STA and AP STA to connect an MBSS and an infrastructure BSS

5.2.14.5 Introduction to mesh functions

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

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

5.2.14.5.1 Mesh discovery

A mesh STA performs either active scanning or passive scanning to discover an operating mesh BSS. Each mesh STA transmits Beacon frames periodically, and responds with Probe Response frames when a Probe Request frame is received, so that neighbor mesh STAs can perform mesh discovery appropriately. The identification of the mesh BSS is given by the Mesh ID element contained in the Beacon and the Probe Response frames. The details for the mesh discovery facility are described in 11C.2.

5.2.14.5.2 Mesh peering management (MPM)

Within a mesh BSS, direct communication between neighbor mesh STAs is allowed only when they are peer mesh STAs. After mesh discovery, two neighbor mesh STAs agree to establish a mesh peering to each other, and, after successfully establishing the mesh peering, they become peer mesh STAs. A mesh STA can establish a mesh peering with multiple neighbor mesh STAs. The mesh peering management (MPM) facilitates the mesh peering establishment and closure of the mesh peerings. The details of MPM are described in 11C.3.

5.2.14.5.3 Mesh security

In an MBSS, mesh link security protocols are used to authenticate a pair of mesh STAs and to establish session keys between them. Mesh authentication protocols establish a shared, common pairwise master key (PMK), and authenticate a peer mesh STA. The authenticated mesh peering exchange protocol relies on the existence of the PMK between the two mesh STAs to establish an authenticated peering and derive session keys. The details of mesh security are described in 8.2a, 11C.3.3, 11C.5, and 11C.6.

5.2.14.5.4 Mesh beaconing and synchronization

In order to assist mesh discovery, mesh power management, and synchronization in a mesh BSS, all mesh STAs periodically transmit Beacon frames. Synchronization in a mesh BSS is maintained by an active synchronization method. The default synchronization method is the neighbor offset synchronization method. Mesh beacon collision avoidance (MBCA) mitigates collisions of Beacon frames among hidden nodes. The details of mesh beaconing and synchronization are described in 11C.12.

5.2.14.5.5 Mesh coordination function (MCF)

A mesh STA uses the mesh coordination function (MCF) for channel access. MCF consists of EDCA (contention-based channel access defined in 9.9a.2) and MCCA (controlled channel access defined in 9.9a.3). MCCA is a reservation based channel access method and aims to optimize the efficiency of frame exchanges in a mesh BSS.

5.2.14.5.6 Mesh power management

A mesh STA can manage the activity level of its links per mesh peering. A mesh STA sets the activity level of each of its mesh peerings to either active mode, light sleep mode, or deep sleep mode. The mesh STA performs mesh power mode tracking for each of its neighbor peer mesh STAs, and delivers the frames based on the rules defined in 11C.13.

5.2.14.5.7 Mesh channel switching

When a mesh STA switches the operating channel, it uses the channel switch protocol defined in 11.9.7 and 11.9a.3. The channel switch protocol enables the propagation of channel switching messages throughout the mesh BSS, prior to the channel switch execution.

5.2.14.5.8 Frame addressing in an MBSS

Three address, four address, and extended address frame formats enable the distribution of messages over multiple instances of the wireless medium within a mesh BSS and integration to the ESS. Frame format details are described in Clause 7 and 9.22.3.

5.2.14.5.9 Mesh path selection and forwarding

Mesh path selection enables path discovery over multiple instances of the wireless medium within a mesh BSS. The overview of the mesh path selection framework is described in 11C.7. The hybrid wireless mesh protocol (HWMP) is defined as the default path selection protocol for the mesh BSS. HWMP provides both proactive path selection and reactive path selection. The details of HWMP are described in 11C.9. The path selection protocol uses link metrics in the assessment of a mesh path to the destination. The airtime link metric is the default link metric. It is defined in 11C.8.

Once the mesh path of a particular pair of the source mesh STA and the destination mesh STA is found through the mesh path selection function, mesh STAs propagate the data by the forwarding function. The details of the forwarding function are described in 9.22.

As a result of the mesh path selection and forwarding, MSDUs are transmitted among all the mesh STAs in a mesh BSS, even if the mesh STAs are not neighbor STAs of each other. Figure 5-6d depicts the MSDU transfer within a mesh BSS.

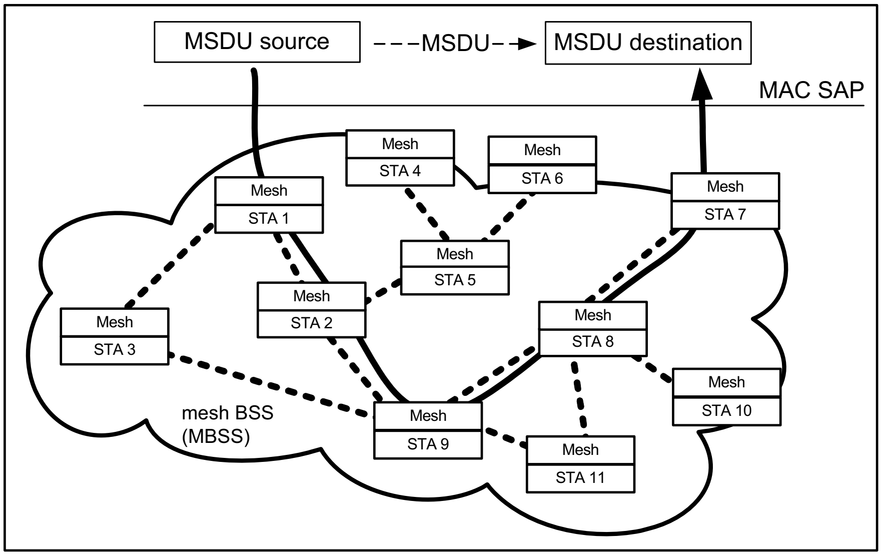


Figure 5-6d—MAC data transport over an MBSS

5.2.14.5.10 Interworking with the DS

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

When a mesh gate has access to IEEE 802 STAs outside the mesh BSS, the mesh gate acts as a proxy for the IEEE 802 STAs outside the MBSS. Such a mesh gate is called a proxy mesh gate. The details of the proxy functionality are described in 11C.10.3.

The details of the mesh BSS interworking are described in 11C.10.

5.2.14.5.11 Intra-mesh congestion control

Intra-mesh congestion control is used to provide flow control over the multi-hop communication. Intra-mesh congestion control is useful to mitigate wasteful wireless medium utilization caused by buffer overflow at mesh STAs. Intra-mesh congestion control consists of three main mechanisms: local congestion monitoring and congestion detection, congestion control signaling, and local rate control. The details of the intra-mesh congestion control are described in 11C.11.

5.2.14.5.12 Emergency service support in mesh BSS

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

5.3 Logical service interfaces

5.3.2 DSS

Change the second and the third paragraphs 5.3.2 as follows:

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

The services that comprise the DSS are as follows:

a) Association (not mesh facility)

b) Disassociation (not mesh facility)

c) Distribution

d) Integration

e) Reassociation (not mesh facility)

f) QoS traffic scheduling (QoS facility only)

g) Interworking with the DS (mesh facility only)

5.4 Overview of the services

5.4.1 Distribution of messages within a DS

5.4.1.1 Distribution

Change the sixth paragraph in 5.4.1.1 as follows:

While IEEE Std 802.11 does not specify DS implementations, it does recognize and support the use of the WM as ~~the~~ one possible DSM. This is specifically supported by the IEEE 802.11 frame formats. (Refer to Clause 7 for details.) A mesh BSS might form an entire DS or a part of a DS using the WM, as shown in

Figure 5-6b. Mesh services are used to form a mesh BSS and distribute messages. Clause 11C defines how mesh BSSs are formed and how messages are distributed through a mesh BSS.

5.4.3 Access control and data confidentiality services

5.4.3.1 Authentication

Change the third and fourth paragraph in 5.4.3.1 as follows:

IEEE Std 802.11 defines ~~three~~ four authentication methods: Open System authentication, Shared Key authentication, ~~and~~ FT authentication, and simultaneous authentication of equals (SAE). Open System authentication admits any STA to the DS. Shared Key authentication relies on WEP to demonstrate knowledge of a WEP encryption key. FT authentication relies on keys derived during the initial mobility domain association to authenticate the stations as defined in Clause 11A. SAE authentication uses finite field cryptography to prove knowledge of a shared password. The IEEE 802.11 authentication mechanism also allows definition of new authentication methods.

An RSNA might support SAE authentication. An RSNA also supports authentication based on IEEE Std 802.1X-2004, or preshared keys (PSKs) after Open System authentication. IEEE 802.1X authentication utilizes the EAP to authenticate STAs and the AS with one another. This standard does not specify an EAP method that is mandatory to implement. See 8.4.4 (RSNA policy selection in an IBSS and for DLS) for a description of the IEEE 802.1X authentication and PSK usage within an IEEE 802.11 IBSS.

Change the sixth paragraph in 5.4.3.1 as follows:

Either SAE authentication or ~~T~~the Open System authentication algorithm is used in RSNs based on infrastructure BSS and IBSS, although Open System authentication is optional in an RSN based on an IBSS. SAE authentication is used in an MBSS. RSNA disallows the use of Shared Key authentication.

Change the eighth and ninth paragraph in 5.4.3.1 as follows:

Because the IEEE 802.1X authentication process could be time-consuming (depending on the authentication protocol in use), the authentication service can be invoked independently of the association service.

This type of ~~P~~preauthentication is typically done by a STA while it is already associated with an AP (with which it previously authenticated). IEEE Std 802.11 does not require that STAs preauthenticate with APs. However, authentication is required before an association can be established.

Insert the following paragraph to the end of 5.4.3.1:

SAE authentication is performed prior to association and a STA can take advantage of the fact that it can be IEEE 802.11 authenticated to many APs simultaneously by completing the SAE protocol with any number of APs while still being associated to another AP. RSNA security can be established after association using the resulting shared key.

5.4.3.2 Deauthentication

Change the first paragraph in 5.4.3.2 as follows:

The deauthentication service is invoked when an existing Open System ~~or~~, Shared Key, or SAE authentication is to be terminated. Deauthentication is an SS.

Insert the following paragraph after the fist paragraph in 5.4.3.2:

When the deauthentication service is terminating SAE authentication any PTKSA, GTKSA, mesh TKSA, or mesh GTKSA related to this SAE authentication is destroyed. If PMK caching is not enabled, deauthentication also destroys any PMKSA created as a result of this successful SAE authentication.

5.4.3.3 Data confidentiality

Insert the following paragraph after the fourth paragraph in 5.4.3.3:

IEEE Std 802.11 provides one security protocol, Counter mode with Cipher-block chaining Message authentication code Protocol (CCMP), for protection of individually addressed and group addressed data frames between mesh STAs.

Insert the following new subclause after 5.6:

5.6a Differences between ESS and MBSS LANs

In 5.2.14, the concept of the MBSS LAN was introduced. It was noted that using the multi-hop capability it appears as if all mesh STAs are directly connected at the MAC layer even if the STAs are not within range of each other. This is different from an IBSS network, where STAs cannot communicate if they are not within range of each other.

Unlike the IBSS, an MBSS might have access to the DS. An MBSS connects through one or more mesh gates to the DS. Since in an MBSS it appears as if all mesh STAs are directly connected at the MAC layer, the MBSS can be used as a DSM. APs, portals, and mesh gates might use the MBSS as a DSM to provide the DSS. Thus, different infrastructure BSSs can unite over the MBSS to form an ESS for example.

An AP identifies the infrastructure BSS that it forms. This is different from the MBSS where no such central entity exists. Whereas infrastructure BSSs need the ESS and thus the DS to unite, the MBSS network appears the same to an LLC layer without the need for access to a DS. However, if an MBSS has one or more mesh gates providing access to the DS, the MBSS might exist in disjointed areas and yet form a single network.

5.8 IEEE Std 802.11 and IEEE Std 802.1X-2004

5.8.2 Infrastructure functional model overview

Change the first paragraph in 5.8.2 as follows:

This subclause summarizes the system setup and operation of an RSN, in three ~~two~~ cases: when a password or PSK is used during IEEE 802.11 authentication, when an IEEE 802.1X AS is used after Open System authentication, and when a PSK is used after Open System authentication. For an ESS, the AP includes an Authenticator, and each associated STA includes a Supplicant.

Insert the following new subclause after 5.8.2.1:

5.8.2.1a AKM Operations with a Password or PSK

The following AKM operations are carried out when authentication is accomplished using a Password or PSK:

— A STA discovers the AP's security policy through passively monitoring the Beacon frames or through active probing. After discovery the STA performs SAE authentication using IEEE 802.11 Authentication frames with the AP (see Figure 5-14a).

— Upon the successful conclusion of SAE, both the STA and AP generate a PMK. The STA then associates to an AP and negotiate security policy. The AKM confirmed in the Association Request and Response is the AKM of SAE or Fast BSS Transition.

— The PMK generated by SAE is used in a 4-Way Handshake using EAPOL-Key frames, just as with IEEE 802.1X authentication when an AS is present. See Figure 5-13.

— The GTK and GTK sequence number are sent from the Authenticator to the Supplicant just as in the AS case. See Figure 5-13 and Figure 5-14.

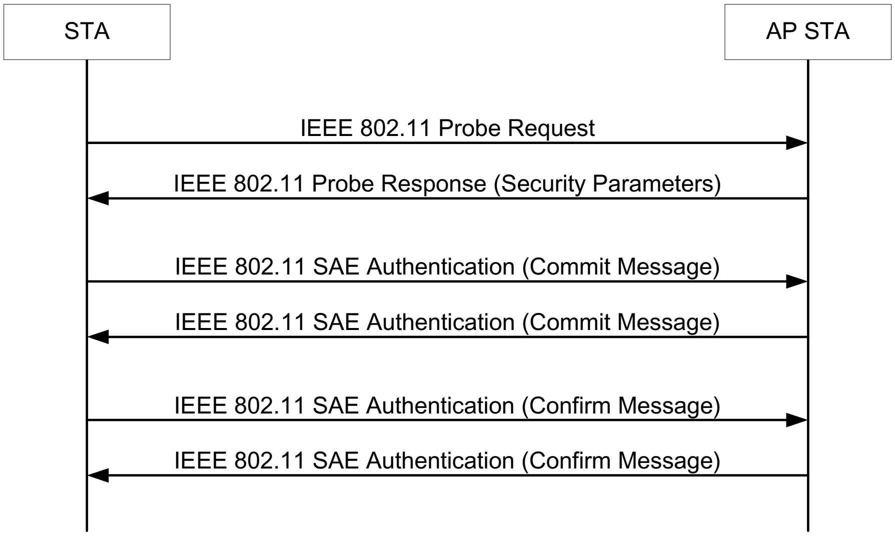

STA
AP STA
IEEE 802.11 Probe Request
IEEE 802.11 Probe Response (Security Parameters)
IEEE 802.11 SAE Authentication (Commit Message)
IEEE 802.11 SAE Authentication (Commit Message)
IEEE 802.11 SAE Authentication (Confirm Message)
IEEE 802.11 SAE Authentication (Confirm Message)

Figure 5-14a—Example using SAE Authentication

Change the title of 5.8.2.2 as follows:

5.8.2.2 Alternate ~~O~~operations with PSK

Change the first paragraph in 5.8.2.2 as follows:

The following AKM operations ~~are~~ represent an alternate operation of using a PSK. This operation has security vulnerabilities when used with a low-entropy key and is recommended to be used only after taking that into account. When this operation is carried out, ~~when~~ the PMK is a PSK.

5.8.3 IBSS functional model description

5.8.3.2 Sample IBSS 4-Way Handshake

Change the ninth paragraph in 5.8.3.2 as follows:

If a fourth STA comes within range and its SME decides to initiate a security association with the three peers, its Authenticator initiates 4-Way Handshakes with each of the other three Supplicants. Similarly, the original three STA Authenticators in the IBSS need to initiate 4-Way Handshakes to the fourth STA Supplicant. A STA learns that a peer STA is RSNA-enabled and the peer's security policy [e.g., whether the Authentication and Key Management Protocol (AKMP) is SAE, PSK, or IEEE 802.1X authentication] from the Beacon or Probe Response frame. The initiation might ~~may~~ start for a number of reasons:

6. MAC service definition

6.1 Overview of MAC services

6.1.2 Security services

Insert the following paragraph to the end of 6.1.2:

A mesh STA with dot11MeshSecurityActivated equal to true shall not use the pairwise cipher suite selectors WEP-40, WEP-104, and TKIP.

6.1.5 MAC data service architecture

Replace Figure 6-1 with the following figure:

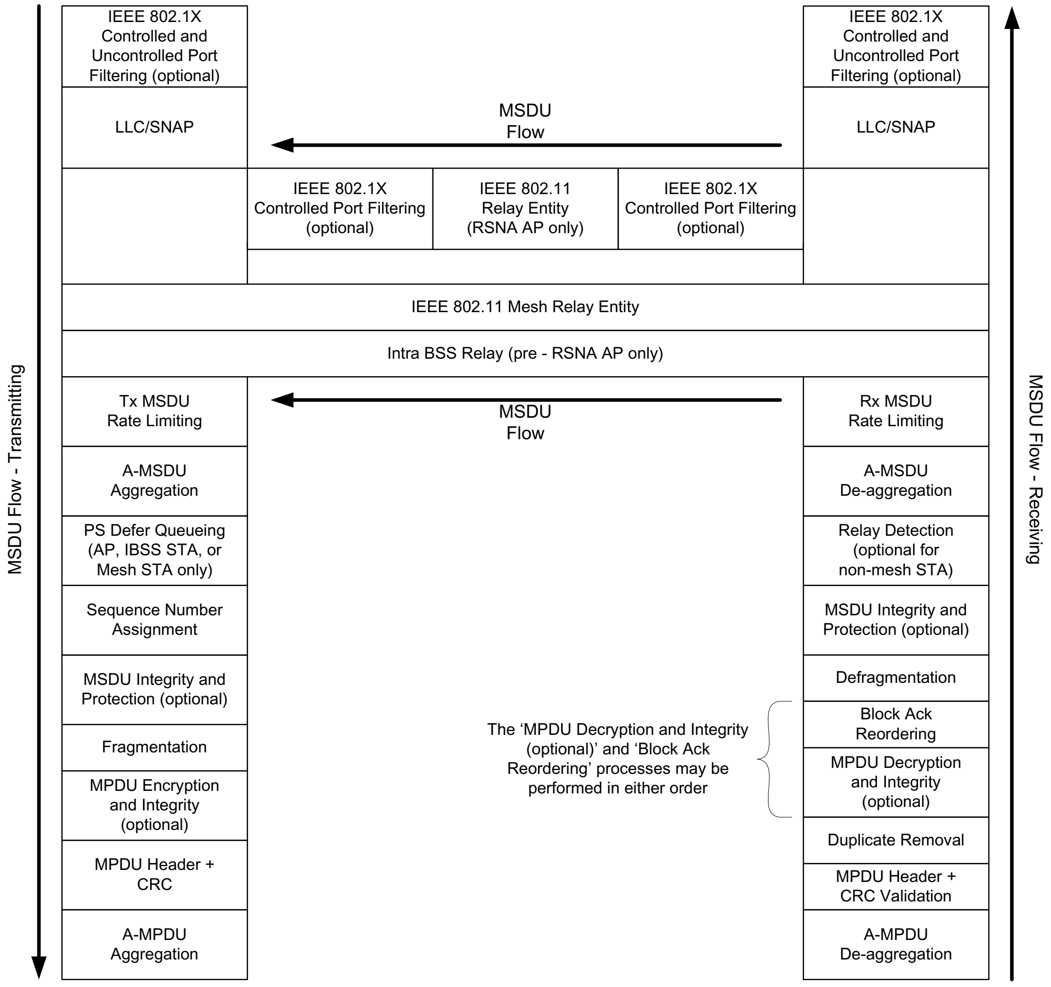

IEEE 802.1X Controlled and Uncontrolled Port Filtering (optional)
IEEE 802.1X Controlled and Uncontrolled Port Filtering (optional)
MSDU Flow
LLC/SNAP
LLC/SNAP
IEEE 802.1X Controlled Port Filtering (optional)
IEEE 802.11 Relay Entity (RSNA AP only)
IEEE 802.1X Controlled Port Filtering (optional)
IEEE 802.11 Mesh Relay Entity
Intra BSS Relay (pre - RSNA AP only)
MSDU Flow - Transmitting
MSDU Flow - Receiving
Tx MSDU Rate Limiting
MSDU Flow
Rx MSDU Rate Limiting
A-MSDU Aggregation
A-MSDU De-aggregation
PS Defer Queueing (AP, IBSS STA, or Mesh STA only)
Relay Detection (optional for non-mesh STA)
Sequence Number Assignment
MSDU Integrity and Protection (optional)
MSDU Integrity and Protection (optional)
Defragmentation
Block Ack Reordering
The 'MPDU Decryption and Integrity (optional)' and 'Block Ack Reordering' processes may be performed in either order
Fragmentation
MPDU Encryption and Integrity (optional)
MPDU Decryption and Integrity (optional)
Duplicate Removal
MPDU Header + CRC
MPDU Header + CRC Validation
A-MPDU Aggregation
A-MPDU De-aggregation

Figure 6-1—MAC data plane architecture

7. Frame formats

7.1 MAC frame formats

7.1.1 Conventions

Insert the following paragraph after the seventh paragraph in 7.1.1:

A QoS Data frame that is transmitted by a mesh STA is referred to as a Mesh Data frame.

7.1.2 General frame format

Change the second paragraph in 7.1.2 as follows:

The Frame Body field is of variable size. The maximum frame body size is determined by the maximum MSDU size (2304 octets) plus the length of the Mesh Control field (6, 12, or 18 octets) if present, or the maximum A-MSDU size (3839 or 7935 octets, depending upon the STA's capability), plus any overhead from security encapsulation.

7.1.3 Frame fields

7.1.3.1 Frame Control field

7.1.3.1.3 To DS and From DS fields

Change third and fourth row of Table 7-2 as follows:

Table 7-2—To/From DS combinations in data frames

To DS and From DS values	Meaning
To DS = 0 From DS = 1	A data frame exiting the DS or being sent by the Port Access Entity in an AP, or a group addressed Mesh Data frame with Mesh Control field present using the three-address MAC header format
To DS = 1 From DS = 1	A data frame using the four-address MAC header format. This standard ~~does not~~ defines procedures for using this combination of field values only in a mesh BSS

7.1.3.1.6 Power Management field

Change 7.1.3.1.6 as follows:

The Power Management field is 1 bit in length and is used to indicate the power management mode of a STA. The value of this field remains constant in each frame from a particular STA within a frame exchange sequence (see Annex S). The value indicates the mode in which the station will be after the successful completion of the frame exchange sequence.

In an infrastructure BSS or in an IBSS the following applies:

A value of 1 indicates that the STA will be in PS mode. A value of 0 indicates that the STA will be in active mode. This field is always set to 0 in frames transmitted by an AP.

In an MBSS the following applies:

A value of 0 in group addressed frames, in management frames transmitted to non-peer STAs, and in Probe Response frames indicates that the mesh STA will be in active mode towards all neighbor mesh STAs. A value of 1 in group addressed frames, in management frames transmitted to non-peer STAs, and in Probe Response frames indicates that the mesh STA will be in deep sleep mode towards all non-peer mesh power STAs.

A value of 0 in individually addressed frames transmitted to a peer mesh STA indicates that the mesh STA will be in active mode towards this peer mesh STA A value of 1 in individually addressed frames transmitted to a peer mesh STA, except Probe Response frames, indicates that the mesh STA will be in either light sleep mode or deep sleep mode towards this peer mesh STA. When the QoS Control field is present in the frame, the Mesh Power Save Level subfield in the QoS Control field indicates whether the mesh STA will be in light sleep mode or in deep sleep mode for the recipient mesh STA as specified in 7.1.3.5.10.

The mesh power mode transition rules are described in 11C.13.3.

7.1.3.1.7 More Data field

Insert the following paragraph after the third paragraph (starting with "For a STA with TDLS peer PSM enabled", which is added by IEEE 802.11z) of 7.1.3.1.7:

The More Data field is 1 in individually addressed frames transmitted by a mesh STA to a peer mesh STA that is either in light sleep mode or in deep sleep mode for the corresponding mesh peering, when additional BUs remain to be transmitted to this peer mesh STA.

Insert the following paragraph to the end of 7.1.3.1.7:

The More Data field is 1 in group addressed frames transmitted by a mesh STA when additional group addressed BUs remain to be transmitted. The More Data field is 0 in group addressed frames transmitted by a mesh STA when no more group addressed BUs remain to be transmitted.

7.1.3.5 QoS Control field

Change the first paragraph of 7.1.3.5 as follows:

The QoS Control field is a 16-bit field that identifies the TC or TS to which the frame belongs ~~and~~ as well as various other QoS-related, A-MSDU related, and mesh-related information about the frame that varies by frame type, ~~and~~ subtype, and type of transmitting STA. The QoS Control field is present in all data frames in which the QoS subfield of the Subtype field is set to 1 (see 7.1.3.1.2). Each QoS Control field comprises five or eight subfields, as defined for the particular sender (HC, ~~or~~ non-AP STA, or mesh STA) and frame type and subtype. The usage of these subfields and the various possible layouts of the QoS Control field are described in 7.1.3.5.1 to ~~7.1.3.5.8~~ 7.1.3.5.11 and illustrated in Table 7-4.

Change Table 7-4 as follows:

Table 7-4—QoS Control field

Applicable frame (sub) types	Bits 0–3	Bit 4	Bits 5–6	Bit 7	Bit 8	Bit 9	Bit 10	Bits 11–15
QoS CF-Poll and QoS CF-Ack+CF-Poll frames sent by HC	TID	EOSP	Ack Policy	Reserved	TXOP Limit			
QoS Data+CF-Poll and QoS Data+CF-Ack+CF-Poll frames sent by HC	TID	EOSP	Ack Policy	A-MSDU Present	TXOP Limit			
QoS Data and QoS Data+CF-Ack frames sent by HC	TID	EOSP	Ack Policy	A-MSDU Present	AP PS Buffer State			
QoS Null frames sent by HC	TID	EOSP	Ack Policy	Reserved	AP PS Buffer State			
QoS Data and QoS Data+CF-Ack frames sent by non-AP STAs that are not a PU buffer STA or a PU sleep STA in a non-mesh BSS	TID	0	Ack Policy	A-MSDU Present	TXOP duration requested			
	TID	1	Ack Policy	A-MSDU Present	Queue size			
QoS Null frames sent by non-AP STAs that are not a PU buffer STA or a PU sleep STA in a non-mesh BSS	TID	0	Ack Policy	Reserved	TXOP duration requested			
	TID	1	Ack Policy	Reserved	Queue Size			
QoS Data and QoS Data+CF-Ack frames sent by PU buffer STAs in a non-mesh BSS	TID	EOSP	Ack Policy	A-MSDU Present	Reserved			
QoS Null frames sent by PU buffer STAs in a non-mesh BSS	TID	EOSP	Ack Policy	Reserved	Reserved			
QoS Data and QoS Data+CF-Ack frames sent by PU sleep STAs in a non-mesh BSS	TID	Reserved	Ack Policy	A-MSDU Present	Reserved			
QoS Null frames sent by PU sleep STAs in a non-mesh BSS	TID	Reserved	Ack Policy	Reserved	Reserved			
All frames sent by mesh STAs in a mesh BSS	TID	EOSP	Ack Policy	A-MSDU Present	Mesh Control Present	Mesh Power Save Level	RSPI	Reserved

7.1.3.5.2 EOSP (end of service period) subfield

Insert the following paragraph to the end of 7.1.3.5.2:

The mesh STA uses the EOSP subfield to indicate the end of the current mesh peer service period (MPSP) in which it operates as the owner. The mesh STA sets the EOSP subfield to 1 in its transmission and retransmissions of the MPSP's final frame to end an MPSP, and sets it to 0 otherwise. See 11C.13.9.4 for details.

7.1.3.5.3 Ack Policy subfield

Change the first (Bit5:0, Bit6:0) and the second (Bit5:1, Bit6:0) row of the Table 7-6 as follows:

Table 7-6—Ack Policy subfield in QoS Control field of QoS data frames

Bits in QoS Control field		Meaning
Bit 5	Bit 6	
0	0	Normal Ack or Implicit Block Ack Request. In a frame that is a non-A-MPDU frame: The addressed recipient returns an ACK or QoS +CF-Ack frame after a short interframe space (SIFS) period, according to the procedures defined in 9.2.0b.9 and 9.9.2.3. If dot11MCCAActivated is false ~~For QoS Null (no data) frames~~, this is the only permissible value for the Ack Policy subfield for QoS Null (no data) frames. In a frame that is part of an A-MPDU: The addressed recipient returns a BlockAck MPDU, either individually or as part of an A-MPDU starting a SIFS after the PPDU carrying the frame, according to the procedures defined in 9.2.0b.10, 9.10.7.5, 9.10.8.3, 9.15.4, and 9.19.3.
1	0	No Ack The addressed recipient takes no action upon receipt of the frame. More de-tails are provided in 9.11. The Ack Policy subfield is set to this value in all directed frames in which the sender does not require acknowledgment. This combination is also used for group addressed frames that use the QoS frame format. This combination is not used for QoS data frames with a TID for which a Block Ack agreement exists. If dot11MCCAActivated is true this value is permissible for the Ack Policy subfield for group addressed QoS Null (no data) frames.

Insert the following new subclauses after 7.1.3.5.8:

7.1.3.5.9 Mesh Control Present subfield

The Mesh Control Present subfield is 1 bit in length, and indicates the presence of a Mesh Control field in the frame body. When the Mesh Control Present subfield is 1, the Frame Body field contains a Mesh Control field as defined in 7.1.3.6.3. The mesh STA sets the Mesh Control Present subfield to 1 in the Mesh Data frame containing an unfragmented MSDU, an A-MSDU, or the first fragment of an MSDU.

7.1.3.5.10 Mesh Power Save Level subfield

The Mesh Power Save Level subfield is 1 bit in length and indicates whether the mesh STA's peer-specific mesh power mode will be deep sleep mode or light sleep mode after the successful completion of the frame exchange sequence.

When the Power Management field in the Frame Control field in the frame is 1, the following applies:

In individually addressed Mesh Data frames, a value of 0 indicates that the mesh STA's peer-specific mesh power mode for the recipient mesh STA will be light sleep mode (see 11C.13.8.4). In individually addressed Mesh Data frames, a value of 1 indicates that the mesh STA's peer-specific mesh power mode for the recipient mesh STA will be deep sleep mode (see 11C.13.8.5).

In group addressed Mesh Data frames, a value of 0 indicates that none of the peer-specific mesh power modes of the mesh STA will be deep sleep mode. In group addressed Mesh Data frames, a value of 1 indicates that at least one of the peer-specific mesh power modes of the mesh STA is deep sleep mode.

The Mesh Power Save Level subfield is reserved if the Power Management field in the Frame Control field is 0.

7.1.3.5.11 Receiver Service Period Initiated (RSPI) subfield

The Receiver Service Period Initiated (RSPI) subfield is 1 bit in length. The subfield is set to 0 to indicate that the mesh peer service period, of which the receiver of this frame is the owner, is not initiated. The subfield is set to 1 to indicate that the mesh peer service period, of which the receiver of this frame is the owner, is initiated. The use of the RSPI subfield is described in 11C.13.9.2. The RSPI subfield is reserved in group addressed frames.

7.1.3.6 Frame Body field

Insert the following heading (7.1.3.6.1) immediately after the heading 7.1.3.6:

7.1.3.6.1 General

Change 7.1.3.6 as follows:

The Frame Body is a variable length field that contains information specific to individual frame types and subtypes. The minimum length of the frame body is 0 octets. The maximum length of the frame body is defined by the maximum length MSDU plus the length of Mesh Control field as defined in 7.1.3.6.3, if present, ~~or A-MSDU~~ plus any overhead for encryption as defined in Clause 8, or by the maximum length A-MSDU plus any overhead for encryption as defined in Clause 8.

Insert the following new subclauses:

7.1.3.6.2 Overhead for encryption

The overhead for encryption is described in Clause 8. When the Mesh Control field is present in the frame body, the Mesh Control field is encrypted as a part of data.

7.1.3.6.3 Mesh Control field

The Mesh Control field is present in the unfragmented Mesh Data frame, in the first fragment of the Mesh Data frame, and in the management frame of subtype Action, Category Multihop Action (Multihop Action frame) transmitted by a mesh STA.

In Mesh Data frames, when the Mesh Control Present subfield in the QoS Control field is 1, the Mesh Control field is prepended to the MSDU and located as follows:

- When the frame body contains an MSDU (or a fragment thereof) and the frame is not encrypted, the Mesh Control field is located in the first octets of the frame body.
- When the frame body contains an MSDU (or a fragment thereof) and the frame is encrypted, the Mesh Control field is located in the first octets of the encrypted data portion.
- When the frame body contains an A-MSDU, the Mesh Control field is located in the Aggregate MSDU subframe header as shown in Figure 7-17c.

In the Multihop Action frame, the Mesh Control field is present as specified in 7.4.16.

The Mesh Control field is of variable length (6, 12, or 18 octets). The structure of the Mesh Control field is defined in Figure 7-4e.

Mesh Flags	Mesh TTL	Mesh Sequence Number	Mesh Address Extension
Octets: 1	1	4	0, 6, or 12

Figure 7-4e—Mesh Control field

The Mesh Flags subfield is 1 octet in length and contains the Address Extension Mode subfield. The structure of the Mesh Flags subfield is shown in Figure 7-4f.

B0 B1	B2 B7
Address Extension Mode	Reserved
Bits: 2	6

Figure 7-4f—Mesh Flags subfield

The Address Extension Mode subfield indicates the contents of the Mesh Address Extension subfield. Table 7-6g1 defines valid values for the Address Extension Mode and describes the corresponding contents of the Mesh Address Extension subfield. If the Address Extension Mode is 00, the Mesh Address Extension subfield is not present. For values 01 and 10, the Mesh Address Extension subfield is present following the Mesh Sequence Number subfield.

Table 7-6g1—Valid values for the Address Extension Mode

Address Extension Mode value (binary)	Address Extension Mode description	Mesh Address Extension subfield length (octets)	Applicable frame types
00	No Mesh Address Extension subfield	0	Data, Management (Multihop Action, group addressed)
01	Mesh Address Extension subfield contains Address 4	6	Management (Multihop Action, individually addressed), Data (proxied, group addressed)
10	Mesh Address Extension subfield contains Address 5 and Address 6	12	Data (proxied, individually addressed)
11	Reserved	—	—

The Mesh TTL subfield is 1 octet in length and contains an unsigned integer corresponding to the remaining number of hops the MSDU/MMPDU is forwarded. How the Mesh TTL is used in both individually and group addressed frames is described in 9.22.4 and 9.22.5.

The Mesh Sequence Number subfield is 4 octets in length and contains an unsigned integer sequence number counter value. Source mesh STAs assign mesh sequence numbers from a single modulo-2^{32} counter, starting at 0 and incrementing by 1 for each MSDU or MMPDU that is transmitted with a Mesh Control field. Usage of the Mesh Sequence Number is described in 9.22.7.

NOTE—It is believed that a 32-bit sequence number is sufficient as the rollover would occur after a period of 5 days assuming a source continuously transmitting at a rate of 10^4 frames per second.

The Mesh Address Extension subfield, shown in Figure 7-4g, is 6 or 12 octets in length and is present only when the Address Extension Mode subfield of the Mesh Flags subfield is a non-zero non-reserved value. The Mesh Address Extension subfield provides additional address fields for mesh address extension as defined in Table 7-6g1. The interpretation of the extended Address fields is described in 9.22.3.

	Address 4	Address 5	Address 6
Octets:	6	6	6

Figure 7-4g—Mesh Address Extension subfield

The Address 4 subfield is present when the Address Extension Mode subfield in the Mesh Flags subfield is 01 (binary). It carries a fourth address that is not included as a part of the MAC header for these frames.

The Address 5 subfield and Address 6 subfield are present when the Address Extension Mode subfield in the Mesh Flags subfield is 10 (binary). It carries the addresses of source and destination end station of the end-to-end IEEE 802 communication in cases where either (or both) of the end stations are not mesh STAs at the beginning or end of a single mesh path. (See Figure 9-38.)

NOTE—This is useful, for example, when the end stations of IEEE 802 communication are non-mesh, external STAs that communicate over a mesh BSS via proxy mesh gates.

Details on the usage of these optional address fields are given in 9.22.3.

7.2 Format of individual frame types

7.2.1 Control frames

7.2.1.4 PS-Poll frame format

Insert the following paragraph to the end of 7.2.1.4:

PS-Poll frame is not used in MBSSs.

7.2.2 Data frames

7.2.2.1 Data frame format

Change the fifth paragraph of 7.2.2.1 as follows:

A STA uses the contents of the Address 1 field to perform address matching for receive decisions. A mesh STA also uses the address matching rules described in 9.22.4, when it receives an individually addressed frame. ~~In cases where the Address 1 field contains a group address, the BSSID also is validated~~ When a STA other than mesh STA (non-mesh STA) receives a frame with the Address 1 field equal to a group address, the STA also validates the BSSID to ensure either that the group addressed frame originated from a STA in the BSS of which the receiving STA is a member, or that it contains the wildcard BSSID value, indicating a data frame sent outside the context of a BSS (dot11OCBEnabled is true in the transmitting STA). When a mesh STA receives a frame with the Address 1 field equal to a group address, the mesh STA also validates the TA to ensure that the group addressed frame originated from one of its peer mesh STA. A mesh STA also uses the address matching rules described in 9.22.5.

Change the thirteenth paragraph of 7.2.2.1 as follows:

The BSSID of the Data frame is determined as follows:

a) If the STA is an AP or is associated with an AP, the BSSID is the address currently in use by the STA contained in the AP.

b) If the STA is a member of an IBSS, the BSSID is the BSSID of the IBSS.

c) If the STA is transmitting a data frame when dot11OCBEnabled is true, the BSSID shall be the wildcard BSSID.

d) If the STA is a member of an MBSS, the BSSID is the address of the transmitter and is equal to the Data frame's TA.

Change the seventeenth to nineteenth paragraph of 7.2.2.1 as follows:

~~The frame body consists of the MSDU (or a fragment thereof) or A-MSDU and a security header and trailer (if and only if the Protected Frame subfield in the Frame Control field is 1).~~ The frame body consists of either

— The MSDU (or a fragment thereof), the Mesh Control field (if and only if the frame is transmitted by a mesh STA and the Mesh Control Present subfield of the QoS Control field is 1), and a security header and trailer (if and only if the Protected Frame subfield in the Frame Control field is 1)

— The A-MSDU and a security header and trailer (if and only if the Protected Frame subfield in the Frame Control field is 1)

The presence of an A-MSDU in the frame body is indicated by setting the A-MSDU Present subfield of the QoS Control field to 1, as shown in Table 7-4.

For data frames of subtype Null (no data), CF-Ack (no data), CF-Poll (no data), and CF-Ack+CF-Poll (no data) and for the corresponding QoS data frame subtypes, the Frame Body field is null (i.e., has a length of 0 octets); these subtypes are used for MAC control purposes. For data frames of subtypes Data, Data+CF-Ack, Data+CF-Poll, and Data+CF-Ack+CF-Poll the Frame Body field contains all of, or a fragment of, an MSDU after any encapsulation for security. For data frames of subtypes QoS Data, QoS Data+CF-Ack, QoS Data+CF-Poll, and QoS Data+CF-Ack+CF-Poll, the Frame Body field contains an MSDU (or fragment thereof) or A-MSDU after any encapsulation for security. For data frames of subtype QoS Data that are transmitted by a mesh STA, the Frame Body field also contains a Mesh Control field, as described in 7.1.3.6.3.

The maximum length of the Frame Body field can be determined from the maximum MSDU length plus the length of the Mesh Control field (if present) plus any overhead from encapsulation for encryption (i.e., it is always possible to send a maximum length MSDU, with any encapsulations provided by the MAC layer within a single data MPDU). When the frame body carries an A-MSDU, the size of the frame body field is ~~may be~~ limited by

— The PHY's maximum PLCP service data unit (PSDU) length

— If A-MPDU aggregation is used, a maximum MPDU length of 4095 octets (see 7.4a)

7.2.2.2 A-MSDU format

Insert the following paragraph to the end of 7.2.2.2:

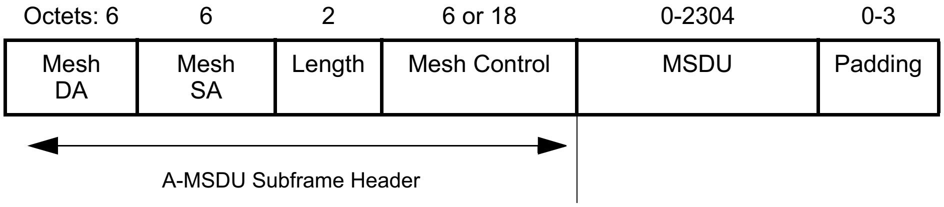
When Mesh Data frames are aggregated, the Aggregate MSDU subframe header includes Mesh DA, Mesh SA, Length, and Mesh Control. The A-MSDU subframe structure for Mesh Data is defined in Figure 7-17c.

The Mesh DA and Mesh SA fields contain the addresses of the destination mesh STA and the source mesh STA, respectively, determined in 9.22.3.

The Length field contains the length in octets of the MSDU.

The format of the Mesh Control field is described in 7.1.3.6.3.

NOTE 4—It is possible to have different Mesh DA, Mesh SA and Mesh Control in Subframe Headers of the same A-MSDU as long as they all map to the same Address 1 and Address 2 values.



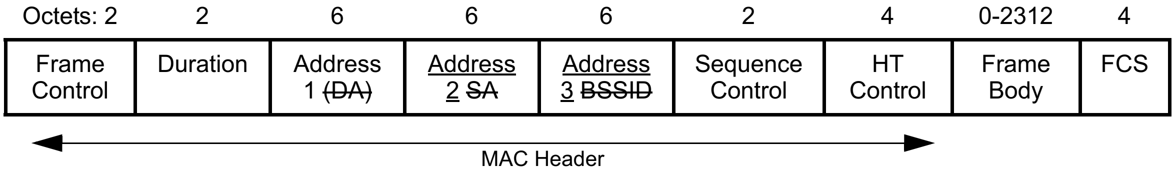
Octets: 6	6	2	6 or 18	0-2304	0-3
Mesh DA	Mesh SA	Length	Mesh Control	MSDU	Padding

Figure 7-17c—A-MSDU Subframe structure for Mesh Data

7.2.3 Management frames

Change the first to the sixth paragraphs of 7.2.3 and Figure 7-18 as follows:

The format of a management frame is defined in Figure 7-18. The Frame Control, Duration, Address 1 ~~(DA)~~, Address 2 ~~SA~~, Address 3, ~~BSSID~~ and Sequence Control fields are present in all management frame subtypes.



Octets: 2	2	6	6	6	2	4	0-2312	4
Frame Control	Duration	Address 1 ~~(DA)~~	Address 2 ~~SA~~	Address 3 ~~BSSID~~	Sequence Control	HT Control	Frame Body	FCS

Figure 7-18—Management frame format

A STA uses the contents of the Address 1 ~~(DA)~~ field to perform the address matching for receive decisions. In the case where the Address 1 ~~(DA)~~ field contains a group address and the frame subtype is other than Beacon or the frame subtype Action, Category Multihop Action (Multihop Action frame), the Address 3 field ~~BSSID~~ also is validated to ensure that the group addressed frame originated from a STA in the BSS of which the receiving STA is a member or from a mesh STA to which mesh peering is maintained. Details of addressing and forwarding of the group addressed frame in an MBSS are defined in 9.22.5. When the Address 1 ~~(DA)~~ field contains a group address and the frame subtype is either Probe Request or Action with Category Public, a wildcard BSSID value matches all receiving STA's BSSIDs. If the frame subtype is Beacon, other address matching rules apply, as specified in 11.1.2.3 (Beacon reception). Frames of subtype Probe Request with a group address in the Address 1 field are additionally processed as described in 11.1.3.2.1 (Sending a probe response). If the frame subtype is Action, the Category is Public, and the Action is 20/40 BSS Coexistence Management, then additional address matching rules for receive decisions apply as specified in 11.14 (20/40 MHz BSS operation) and 11.16 (20/40 BSS Coexistence Management frame usage).

The address fields for all management frames ~~do not vary by frame subtype~~ except Multihop Action frames are as follows:

- The Address 1 field of the management frame is the RA (=DA) and is determined as the destination of the frame.
- The Address 2 field of the management frame is the TA (=SA) and is determined as the address of the STA transmitting the frame.
- The Address 3 field ~~BSSID~~ of the management frame is set and determined as follows:
  a) In management frames of subtype Probe Request, the Address 3 field is the BSSID. T~~t~~he BSSID is either a specific BSSID as described in item c) below or the wildcard BSSID as defined in the procedures specified in 11.1.3.
  b) In management frames of subtype Action, Category Public, the Address 3 field is the BSSID. T~~t~~he BSSID value is set according to 11.18 (Public Action frame addressing).
  c) Otherwise:
     1) If the STA is an AP or is associated with an AP, the Address 3 field is the BSSID. T~~t~~he BSSID is the address currently in use by the STA contained in the AP.
     2) If the STA is an AP or is transmitting the management frame to an AP, the Address 3 field is the BSSID. T~~t~~he BSSID is the address currently in use by the STA contained in the AP.
     3) If the STA is transmitting the management frame to one or more members of an IBSS, the Address 3 field is the BSSID. T~~t~~he BSSID is the BSSID of the IBSS.
     4) If dot11OCBEnabled is true, the Address 3 field is ~~BSSID shall be~~ the wildcard BSSID.
     5) If the STA is a mesh STA, the Address 3 field is the TA.

~~The DA field is the destination of the frame.~~

~~The SA field is the address of the STA transmitting the frame.~~

The address fields for the Multihop Action frame are as follows:

- The Address 1 field is the RA and is determined as the address of the receiver of the frame.
- The Address 2 field is the TA and is determined as the address of the transmitter of the frame.
- The Address 3 field is the DA and is determined as the address of the destination mesh STA of the frame.

NOTE—Address 4 is included in the Mesh Control field.

7.2.3.1 Beacon frame format

Change the contents of the order 4 row, order 10 row, order 23 row, order 24 row, and order 38 row of Table 7-8 as follows:

Table 7-8—Beacon frame body

Order	Information	Notes
4	Service Set Identifier (SSID)	If dot11MeshActivated is true, the SSID element is the wildcard value as described in 7.3.2.1.
10	Traffic indication map (TIM)	The TIM element is present only within Beacon frames generated by APs or mesh STAs.
23	EDCA Parameter Set	The EDCA Parameter Set element is present if dot11QosOptionImplemented is true, and dot11MeshActivated is false, and the QoS Capability element is not present.
24	QoS Capability	The QoS Capability element is present if dot11QosOptionImplemented is true, and dot11MeshActivated is false, and EDCA Parameter Set element is not present.
38	HT Operation	The HT Operation element is included by an AP and a mesh STA when dot11HighThroughputOptionImplemented attribute is true.

Insert the following additional rows (preserving their order) in Table 7-8 just before the Vendor Specific element.

Table 7-8—Beacon frame body

Order	Information	Notes
49	Mesh ID	The Mesh ID element is present if dot11MeshActivated is true.
50	Mesh Configuration	The Mesh Configuration element is present if dot11MeshActivated is true.
51	Mesh Awake Window	The Mesh Awake Window element is optionally present if dot11MeshActivated is true.
52	Beacon Timing	The Beacon Timing element is optionally present if both dot11MeshActivated and dot11MBCAActivated are true.
53	MCCAOP Advertisement Overview	The MCCAOP Advertisement Overview element is optionally present if both dot11MeshActivated and dot11MCCAActivated are true.
54	MCCAOP Advertisement	One or more MCCAOP Advertisement elements are optionally present if both dot11MeshActivated and dot11MCCAActivated are true.
55	Mesh Channel Switch Parameters	The Mesh Channel Switch Parameters element is optionally present when dot11MeshActivated is true and either Channel Switch Announcement element or Extended Channel Switch Announcement element is present.

7.2.3.6 Reassociation Request frame format

Change the contents of the order 13 row of Table 7-12 as follows:

Table 7-12—Reassociation Request frame body

Order	Information	Notes
13	Fast BSS Transition	An FTIE is present in a Reassociation Request frame if dot11FastBSSTransitionEnabled is true and dot11RSNAAuthenticationSuiteSelected is 00-0F-AC:3, ~~or~~ 00-0F-AC:4 or 00-0F-AC:9 (i.e., part of a fast BSS transition in an RSN).

7.2.3.8 Probe Request frame format

Change the contents of the order 1 row of Table 7-14 as follows:

Table 7-14—Probe Request frame body

Order	Information	Notes
1	SSID	If dot11MeshActivated is true, the SSID element is the wildcard value as described in 7.3.2.1.

Insert the following additional rows (preserving their order) in before the last row of Table 7-14 just before the Vendor Specific element:

Table 7-14—Probe Request frame body

Order	Information	Notes
13	Mesh ID	The Mesh ID element is present if dot11MeshActivated is true.

7.2.3.9 Probe Response frame format

Change the contents of the order 4 row, order 22 row, and order 36 row of Table 7-15 as follows:

Table 7-15—Probe Response frame body

Order	Information	Notes
4	SSID	If dot11MeshActivated is true, the SSID element is the wildcard value as described in 7.3.2.1.
22	EDCA Parameter Set	The EDCA Parameter Set element is present if dot11QosOptionImplemented is true and dot11MeshActivated is false.

Table 7-15—Probe Response frame body *(continued)*

Order	Information	Notes
36	HT Operation	The HT Operation element is included by an AP and a mesh STA when dot11HighThroughputOptionImplemented attribute is true.

Insert the following additional rows (preserving their order) before the last row of Table 7-15 just before the Vendor Specific element:

Table 7-15—Probe Response frame body

Order	Information	Notes
48	Mesh ID	The Mesh ID element is present if dot11MeshActivated is true.
49	Mesh Configuration	The Mesh Configuration element is present if dot11MeshActivated is true.
50	Mesh Awake Window	The Mesh Awake Window element is optionally present if dot11MeshActivated is true.
51	Beacon Timing	The Beacon Timing element is optionally present if both dot11MeshActivated and dot11MBCAActivated are true.
52	MCCAOP Advertisement Overview	The MCCAOP Advertisement Overview element is optionally present if both dot11MeshActivated and dot11MCCAActivated are true.
53	MCCAOP Advertisement	One or more MCCAOP Advertisement elements are optionally present if both dot11MeshActivated and dot11MCCAActivated are true.
54	Mesh Channel Switch Parameters	The Mesh Channel Switch Parameters element is optionally present if dot11MeshActivated is true and either Channel Switch Announcement element or Extended Channel Switch Announcement element is present.

7.2.3.10 Authentication frame format

Insert the following sentence to the end of the first paragraph in 7.2.3.10:

SAE authentication is used when dot11MeshActiveAuthenticationProtocol is sae (1).

Insert the following new rows into the Table 7-16:

Table 7-16—Authentication frame body

Order	Information	Notes
10	Finite Cyclic Group	An unsigned integer indicating a finite cyclic group as described in 8.2a.4. This is present in SAE authentication frames as defined in Table 7-17.
11	Anti-Clogging Token	A random bit-string used for anti-clogging purposes as described in 8.2a.6. This is present in SAE authentication frames as defined in Table 7-17.
12	Send-Confirm	A binary encoding of an integer used for anti-replay purposes as described in 8.2a.7.5. This is present in SAE authentication frames as defined in Table 7-17.
13	Scalar	An unsigned integer encoded as described in 8.2a.7.4. This is present in SAE authentication frames as defined in Table 7-17.
14	Element	A field element from a finite field encoded as described in 8.2a.7.4. This is present in SAE authentication frames as defined in Table 7-17.
15	Confirm	An unsigned integer encoded as described in 8.2a.7.5. This is present in SAE authentication frames as defined in Table 7-17.

Change the title of Table 7-17 as follows:

Insert the following new rows into the Table 7-17 and change the title of the fourth column:

Table 7-17—Presence of fields and ~~information~~ elements in Authentication frames

Authentication Algorithm	Authentication transaction sequence number	Status Code	Presence of fields 4–~~9~~15
SAE	1	Status	Scalar is present if Status is zero. Element is present if Status is zero. Anti-Clogging Token is present if status is 76 or if frame is in response to a previous rejection with Status 76. Finite Cyclic Group is present if Status is zero or 76.
SAE	2	Status	Send-Confirm is present. Confirm is present.

7.2.3.12 Action frame format

Change the second row of the Table 7-19 as follows:

Table 7-19—Action frame body

Order	Information
2 – (Last – 1)	One or more vendor-specific elements are optionally present. These elements follow all other elements. These elements are absent when the Category subfield of the Action field is Vendor-Specific, ~~or~~ Vendor-Specific Protected, or Self-protected.

7.3 Management frame body components

7.3.1 Fields that are not information elements

7.3.1.1 Authentication algorithm number field

Insert the following text after "Authentication algorithm number = 2: Fast BSS Transition":

Authentication algorithm number = 3: simultaneous authentication of equals (SAE)

7.3.1.4 Capability Information field

Insert the following sentence to the end of fourth paragraph in 7.3.1.4:

A mesh STA sets the ESS and IBSS subfields to 0 in transmitted Beacon or Probe Response management frames.

Insert the following sentence to the end of fifth paragraph in 7.3.1.4:

A mesh STA sets the CF-Pollable and CF-Poll Request subfields to 0.

Insert the following paragraph after the 10th paragraph in 7.3.1.4:

A mesh STA sets the Privacy subfield to 1 in transmitted Beacon or Probe Response management frames if data confidentiality is required for all data frames exchanged within the MBSS. If data confidentiality is not required, a mesh STA sets the Privacy subfield to 0 within these management frames.

Insert the following paragraph after the 12th paragraph in 7.3.1.4:

A mesh STA sets the Short Preamble subfield to 1 when dot11ShortPreambleOptionImplemented is true. Otherwise, a mesh STA sets the Short Preamble subfield to 0.

Change the 24th paragraph in 7.3.1.4 as follows:

For IBSS and MBSS, the Short Slot Time subfield is set to 0.

7.3.1.7 Reason Code field

Change the first paragraph in 7.3.1.7 as follows:

This Reason Code field is used to indicate the reason that an unsolicited notification management frame of type Disassociation, Deauthentication, DELTS, DELBA, ~~or~~ DLS Teardown, or Mesh Peering Close was generated. It is contained in the Mesh Channel Switch Parameters element to indicate the reason for the channel switch. It is contained in the Path Error (PERR) element to indicate the reason for the path error.

Insert the following rows into Table 7-22 and change the last row (Reserved) as follows:

Table 7-22—Reason codes

Reason code	Meaning
52	"MESH-PEERING-CANCELLED". SME cancels the mesh peering instance with the reason other than reaching the maximum number of peer mesh STAs
53	"MESH-MAX-PEERS". The mesh STA has reached the supported maximum number of peer mesh STAs
54	"MESH-CONFIGURATION-POLICY-VIOLATION". The received information violates the Mesh Configuration policy configured in the mesh STA profile
55	"MESH-CLOSE-RCVD". The mesh STA has received a Mesh Peering Close message requesting to close the mesh peering.
56	"MESH-MAX-RETRIES". The mesh STA has re-sent dot11MeshMaxRetries Mesh Peering Open messages, without receiving a Mesh Peering Confirm message.
57	"MESH-CONFIRM-TIMEOUT". The confirmTimer for the mesh peering instance times out.
58	"MESH-INVALID-GTK". The mesh STA fails to unwrap the GTK or the values in the wrapped contents do not match
59	"MESH-INCONSISTENT-PARAMETERS". The mesh STA receives inconsistent information about the mesh parameters between Mesh Peering Management frames
60	"MESH-INVALID-SECURITY-CAPABILITY". The mesh STA fails the authenticated mesh peering exchange because due to failure in selecting either the pairwise ciphersuite or group ciphersuite
61	"MESH-PATH-ERROR-NO-PROXY-INFORMATION". The mesh STA does not have proxy information for this external destination.
62	"MESH-PATH-ERROR-NO-FORWARDING-INFORMATION". The mesh STA does not have forwarding information for this destination.
63	"MESH-PATH-ERROR-DESTINATION-UNREACHABLE". The mesh STA determines that the link to the next hop of an active path in its forwarding information is no longer usable.
64	"MAC-ADDRESS-ALREADY-EXISTS-IN-MBSS". The Deauthentication frame was sent because the MAC address of the STA already exists in the mesh BSS. See 11.3.3.
65	"MESH-CHANNEL-SWITCH-REGULATORY-REQUIREMENTS". The mesh STA performs channel switch to meet regulatory requirements.
66	"MESH-CHANNEL-SWITCH-UNSPECIFIED". The mesh STA performs channel switch with unspecified reason.
67~~52~~–65 535	Reserved

7.3.1.8 AID field

Change the first paragraph in 7.3.1.8 as follows:

~~The~~ In infrastructure BSS operation, the AID field is a value assigned by an AP during association that represents the 16-bit ID of a STA. In mesh BSS operation, the AID field is a value that represents the 16-bit ID of a neighbor peer mesh STA. An AID value is assigned by a mesh STA that receives and accepts a Mesh Peering Open frame to the transmitter of the Mesh Peering Open frame during the mesh peering establishment process (see 11C.3.1). The length of the AID field is 2 octets. The AID field is illustrated in Figure 7-26 (AID field).

7.3.1.9 Status Code field

Insert the following rows into Table 7-23 and change the last row (Reserved) as shown.

Table 7-23—Status codes

Status code	Meaning
76	Authentication is rejected because an Anti-Clogging Token is required.
77	Authentication is rejected because the offered finite cyclic group is not supported.
78	The TBTT adjustment request has not been successful because the STA could not find an alternative TBTT.
79~~76~~–65 535	Reserved

7.3.1.11 Action field

Insert the following new rows into Table 7-24, and change the first Reserved row as follows.

Table 7-24—Category values

Code	Meaning	See subclause	Robust
13	Mesh	7.4.15	Yes
14	Multihop	7.4.16	Yes
15	Self-protected	7.4.14	No
~~13~~ 16	Reserved	—	—

Insert the following new subclauses after the last subclause in 7.3.1:

7.3.1.35 Send-Confirm field

The Send-Confirm field is used with SAE authentication as an anti-replay counter as specified in 8.2a. See Figure 7-36t.

	Send-Confirm
Octets:	2

Figure 7-36t—Send-Confirm field

7.3.1.36 Anti-Clogging Token field

The Anti-Clogging Token field is used with SAE authentication for denial-of-service protection as specified in 8.2a. See Figure 7-36u.

	Anti-Clogging Token
Octets:	variable

Figure 7-36u—Anti-Clogging Token field

7.3.1.37 Scalar field

The Scalar field is used with SAE authentication to communicate cryptographic material as specified in 8.2a. See Figure 7-36v.

	Scalar
Octets:	variable

Figure 7-36v—Scalar field

7.3.1.38 Element field

The Element field is used with SAE authentication to communicate an element in a finite field as specified in 8.2a. See Figure 7-36w.

	Element
Octets:	variable

Figure 7-36w—Element field

7.3.1.39 Confirm field

The Confirm field is used with SAE authentication to authenticate and prove possession of a cryptographic key as specified in 8.2a. See Figure 7-36x.

	Confirm
Octets:	variable

Figure 7-36x—Confirm field

7.3.1.40 Finite Cyclic Group field

The Finite Cyclic Group is used in SAE to indicate which cryptographic group to use in the SAE exchange as specified in 8.2a. See Figure 7-36y.

	Finite Cyclic Group
Octets:	2

Figure 7-36y—Finite Cyclic Group field

7.3.2 Information elements

Insert the following rows (ignoring the header row and footer note) in Table 7-26 in the correct position to preserve ordering by the "Element ID" column and update the "Reserved" range of codes appropriately.

Table 7-26—Element IDs

Element	Element ID	Total length of element in octets including the Type and Length octets	Extensible
Mesh Configuration (see 7.3.2.98)	113	9	Yes
Mesh ID (see 7.3.2.99)	114	2 to 34	
Mesh Link Metric Report (see 7.3.2.100)	115	3 to 257	
Congestion Notification (see 7.3.2.101)	116	16	Yes
Mesh Peering Management (see 7.3.2.102)	117	5, 7, 9, 21, 23, or 25	Yes
Mesh Channel Switch Parameters (see 7.3.2.103)	118	8	Yes
Mesh Awake Window (see 7.3.2.104)	119	4	Yes
Beacon Timing (see 7.3.2.105)	120	3 to 255	
MCCAOP Setup Request (see 7.3.2.106)	121	8	Yes
MCCAOP Setup Reply (see 7.3.2.107)	122	4 or 9	
MCCAOP Advertisement (see 7.3.2.109)	123	4 to 257	Yes
MCCAOP Teardown (see 7.3.2.110)	124	3 or 9	Yes
Gate Announcement (GANN) (see 7.3.2.111)	125	17	Yes
Root Announcement (RANN) (see 7.3.2.112)	126	23	Yes
Path Request (PREQ) (see 7.3.2.113)	130	39 to 254	Yes
Path Reply (PREP) (see 7.3.2.114)	131	33 or 39	Yes
Path Error (PERR) (see 7.3.2.115)	132	17 to 251	Yes
Proxy Update (PXU) (see 7.3.2.116)	137	21 to 257	Yes
Proxy Update Confirmation (PXUC) (see 7.3.2.117)	138	9	Yes
Authenticated Mesh Peering Exchange (see 7.3.2.118)	139	86 to 257	
MIC (see 7.3.2.119)	140	18	
MCCAOP Advertisement Overview (see 7.3.2.108)	174	8	Yes

7.3.2.1 SSID element

Change the second paragraph of 7.3.2.1 as follows:

The length of the SSID information field is between 0 and 32 octets. A 0 length information field is used within Probe Request management frames to indicate the wildcard SSID. The wildcard SSID is also used in Beacon and Probe Response frames transmitted by mesh STAs.

7.3.2.2 Supported Rates element

Change the first sentence in the second paragraph in 7.3.2.2 as follows:

Within Beacon, Probe Response, Association Response, ~~and~~ Reassociation Response, Mesh Peering Open, and Mesh Peering Confirm management frames, each Supported Rate contained in the BSSBasicRateSet parameter is encoded as an octet with the MSB (bit 7) set to 1, and bits 6 through 0 are set to the data rate, if necessary rounded up to the next 500 kb/s, in units of 500 kb/s.

Change the first sentence in the third paragraph in 7.3.2.2 as follows:

Within Beacon, Probe Response, Association Response, ~~and~~ Reassociation Response, Mesh Peering Open, and Mesh Peering Confirm management frames, each BSS membership selector contained in the BSSMembershipSelectorSet parameter is encoded as an octet with the MSB (bit 7) set to 1, and bits 6 through 0 are set to the encoded value for the selector as found in Table 7-26a (e.g., an HT PHY BSS membership selector contained in the BSSMembershipSelectorSet parameter is encoded as X'FF').

7.3.2.6 TIM element

Change the fifth and sixth paragraph of 7.3.2.6 as follows:

The Bitmap Control field is a single octet. Bit 0 of the field contains the Traffic Indicator bit associated with ~~Association~~ AID 0. This bit is set to 1 in TIM elements with a value of 0 in the DTIM Count field when one or more group addressed MSDUs/MMPDUs are buffered at the AP or the mesh STA. The remaining 7 bits of the field form the Bitmap Offset.

The traffic-indication virtual bitmap, maintained by the AP or the mesh STA that generates a TIM, consists of 2008 bits, and is organized into 251 octets such that bit number N ($0 \leq N \leq 2007$) in the bitmap corresponds to bit number (N mod 8) in octet number (N / 8) where the low-order bit of each octet is bit number 0, and the high order bit is bit number 7. Each bit in the traffic-indication virtual bitmap corresponds to traffic buffered for a specific neighbor peer mesh STA within the MBSS that the mesh STA is prepared to deliver or STA within the BSS that the AP is prepared to deliver, at the time the Beacon frame is transmitted. Bit number N is 0 if there are no directed MSDUs/MMPDUs buffered for the STA whose ~~Association~~ AID is N. If any directed MSDUs/MMPDUs for that STA are buffered and the AP or the mesh STA is prepared to deliver them, bit number N in the traffic indication virtual bitmap is 1. A PC ~~may~~ might decline to set bits in the TIM for CF-Pollable STAs it does not intend to poll (see 11.2.1.6).

7.3.2.9 Country element

Change the third paragraph in 7.3.2.9 as follows:

The Country String field of the element shall be 3 octets in length. The AP and mesh STA shall set this field to the value contained in the dot11CountryString attribute before transmission in a Beacon or Probe Response frame. Upon reception of this element, a STA shall set the value of the dot11CountryString to the value contained in this field.

7.3.2.13 ERP Information element

Change the first paragraph of 7.3.2.13 as follows:

The ERP element contains information on the presence of Clause 15 or Clause 18 STAs in the BSS that are not capable of Clause 19 (ERP-OFDM) data rates. It also contains the requirement of the ERP element sender (AP in a BSS, ~~or~~ STA in an IBSS, or mesh STA in an MBSS) as to the use of protection mechanisms to optimize BSS performance and as to the use of long or short Barker preambles. See Figure 7-50 for a definition of the frame element.

7.3.2.14 Extended Supported Rates element

Change the first sentence in the second paragraph in 7.3.2.14 as follows:

Within Beacon, Probe Response, Association Response, ~~and~~ Reassociation Response, Mesh Peering Open, and Mesh Peering Confirm management frames, each supported rate contained in the BSSBasicRateSet parameter, as defined in 10.3.10.1, is encoded as an octet with the MSB (bit 7) set to 1 and bits 6 through 0 are set to the appropriate value from the valid range column of the DATA_RATE row of the table in 10.4.4.2 (e.g., a 1 Mb/s rate contained in the BSSBasicRateSet parameter is encoded as X'82').

Change the first sentence in the third paragraph in 7.3.2.14 as follows:

Within Beacon, Probe Response, Association Response, ~~and~~ Reassociation Response, Mesh Peering Open, and Mesh Peering Confirm management frames, each BSS membership selector contained in the BSSMembershipSelectorSet parameter is encoded as an octet with the MSB (bit 7) set to 1, and bits 6 through 0 are set to the encoded value for the selector as found in Table 7-26a (e.g., an HT PHY BSS membership selector contained in the BSSMembershipSelectorSet parameter is encoded as X'FF').

7.3.2.16 Power Capability element

Change the first sentence in the last paragraph in 7.3.2.16 as follows:

The Power Capability element is included in Association Request frames~~,~~ as described in 7.2.3.4 , ~~and~~ Reassociation Request frames~~,~~ as described in 7.2.3.6 , and Mesh Peering Open frame as described in 7.4.14.2.2.

7.3.2.19 Supported Channels element

Change the first sentence in the fifth paragraph in 7.3.2.19 as follows:

The Supported Channels element is included in Association Request frames~~,~~ as described in 7.2.3.4 , ~~and~~ Reassociation Request frames~~,~~ as described in 7.2.3.6 , and Mesh Peering Open frame as described in 7.4.14.2.2.

7.3.2.20 Channel Switch Announcement element

Change the first paragraph of 7.3.2.20 as follows:

The Channel Switch Announcement element is used by an AP in a BSS, ~~or~~ a STA in an IBSS, or a mesh STA in an MBSS to advertise when it is changing to a new channel and the channel number of the new channel. The format of the Channel Switch Announcement element is shown in Figure 7-57.

Change the fifth paragraph of 7.3.2.20 as follows:

For non-mesh STAs, t~~T~~he Channel Switch Count field either is set to the number of TBTTs until the STA sending the Channel Switch Announcement element switches to the new channel or is set to 0. A value of 1 indicates that the switch occurs immediately before the next TBTT. A value of 0 indicates that the switch occurs at any time after the frame containing the element is transmitted.

For mesh STAs, the Channel Switch Count field is encoded as an octet with bits 6 to 0 are set to the time, in units of 2TU when MSB (bit 7) set to 0, or in units of 100TU when MSB (bit 7) set to 1, until the mesh STA sending the Channel Switch Announcement element switches to the new channel. A value of 0 for bits 6 to 0 indicates that the switch occurs at any time after the frame containing the element is transmitted. For example, a 200 TU channel switch time is encoded as X'82' and a 10TU channel switch time is encoded as X'05'.

7.3.2.20a Secondary Channel Offset element

Change the first paragraph of 7.3.2.20a as follows:

The Secondary Channel Offset element is used by an AP in a BSS, ~~or~~ a STA in an IBSS, or a mesh STA in an MBSS together with the Channel Switch Announcement element when changing to a new 40 MHz channel. The format of the Secondary Channel Offset element is shown in Figure 7-57a.

7.3.2.25 RSN element

7.3.2.25.2 AKM Suites

Insert two new rows and change the existing 'Reserved' row in Table 7-34 as follows:

Table 7-34—AKM suite selectors

OUI	Suite Type	Meaning		
		Authentication type	**Key management type**	**Key derivation type**
00-0F-AC	8	SAE Authentication with SHA-256 or using PMKSA caching as defined in 8.4.6.2 with SHA-256 key derivation	RSNA key management as defined in 8.5, PMKSA caching as defined in 8.4.6.2 with SHA256 key derivation or authenticated mesh peering exchange as defined in 11C.5	Defined in 8.5.1.5.2
00-0F-AC	9	FT authentication over SAE with SHA-256	FT key management defined in 8.5.1.5	Defined in 8.5.1.5.2
00-0F-AC	~~8~~10–255	Reserved	Reserved	Reserved

Change the sixth paragraph in 7.3.2.25.2 as follows:

The AKM suite selector value 00-0F-AC:8 (i.e., SAE Authentication with SHA-256 or using PMKSA caching as defined in 8.4.6.2 with SHA-256 key derivation) is used when either a password or PSK is used with RSNA key management.

NOTE—Selector values 00-0F-AC:1 and 00-0F-AC:8 can simultaneously be enabled by an Authenticator.

The AKM suite selector value 00-0F-AC:2 (PSK) is used when an alternate form of PSK is used with RSNA key management.

7.3.2.25.4 PMKID

Change the second item after the first paragraph in 7.3.2.25.4 as follows:

b) A cached PMKSA from an EAP or SAE authentication

7.3.2.53 Extended Channel Switch Announcement element

Change the first paragraph of 7.3.2.53 as follows:

The Extended Channel Switch Announcement element is used by an access point (AP) in a basic service set (BSS), ~~or~~ a STA in an independent basic service set (IBSS), or a mesh STA in an MBSS to advertise when the BSS, or MBSS is changing to a new channel or a new channel in a new regulatory class. The announcement includes both the regulatory class and the channel number of the new channel. The element is present only when an extended channel switch is pending. The format of the Extended Channel Switch Announcement element is shown in Figure 7-95o12.

Change the third paragraph of 7.3.2.53 as follows:

The Channel Switch Mode field indicates any restrictions on transmission until a channel switch. An AP in a BSS or a STA in an IBSS sets the Channel Switch Mode field to either 0 or 1 on transmission as specified in 11.9.7.1 and 11.9.7.2. The Channel Switch Mode field is reserved in an MBSS.

Change the last paragraph of 7.3.2.53 as follows:

For non-mesh STAs, t~~T~~he Channel Switch Count field indicates either the number of target beacon transmission times (TBTTs) until the STA sending the Extended Channel Switch Announcement element switches to the new channel or a value of zero. A value of one indicates that the switch occurs immediately before the next TBTT. A value of zero indicates that the switch occurs anytime after the frame containing the element is transmitted.

For mesh STAs, the Channel Switch Count field is encoded as an octet with bits 6 to 0 are set to the time, in units of 2TU when MSB (bit 7) set to 0, or in units of 100TU when MSB (bit 7) set to 1, until the mesh STA sending the Channel Switch Announcement element switches to the new channel. A value of 0 for bits 6 to 0 indicates that the switch occurs at any time after the frame containing the element is transmitted. For example, a 200 TU channel switch time is encoded as X'82' and a 10TU channel switch time is encoded as X'05'.

7.3.2.56 HT Capabilities element

7.3.2.56.1 HT Capabilities element structure

Change the second sentence in the second paragraph in 7.3.2.56.1 as follows:

The HT Capabilities element is present in Beacon, Association Request, Association Response, Reassociation Request, Reassociation Response, Probe Request, ~~and~~ Probe Response frames, Mesh Peering Open frames, and Mesh Peering Close frames.

7.3.2.56.2 HT Capabilities Info field

Insert the following paragraph to the end of 7.3.2.56.2:

The following subfields are reserved for a mesh STA: Tx STBC, Rx STBC, PSMP Support.

7.3.2.56.5 HT Extended Capabilities field

Insert the following paragraph to the end of 7.3.2.56.5:

The following subfield is reserved for a mesh STA: PCO.

7.3.2.57 HT Operation element

Insert the following sentence to the end of the fourth paragraph in 7.3.2.57:

The “Reserved in MBSS?” column indicates whether each field is reserved (Y) or not reserved (N) when this element is present in a frame transmitted within an MBSS.

Change Table 7-43p in 7.3.2.57 as follows (note that not all table cells are shown here):

Table 7-43p—HT Operation element

Field	Definition	Encoding	Reserved in IBSS?	Reserved in MBSS?
Primary Channel	*The text in these table cells remains unchanged for these rows.*			N
Secondary Channel Offset				N
STA Channel Width				N
RIFS Mode				Y
HT Protection				N
Nongreenfield HT STAs Present	AP indicates ~~Indicates~~ if any HT STAs that are not HT-greenfield capable have associated. Mesh STA indicates if it establishes a mesh peering with a HT STA that is not HT-greenfield capable. Determines when a non-AP STA should use HT-greenfield protection. Present in Beacon and Probe response frames transmitted by an AP or a mesh STA. Otherwise reserved. See 9.13.3.1.	Set to 0 if all HT STAs that are associated are HT-greenfield capable or all HT peer mesh STAs are HT-greenfield capable. Set to 1 if one or more HT STAs that are not HT-greenfield capable are associated or one or more HT peer mesh STA are not HT-greenfield capable.	Y	N

Table 7-43p—HT Operation element *(continued)*

Field	Definition	Encoding	Reserved in IBSS?	Reserved in MBSS?
OBSS Non-HT STAs Present	*The text in these table cells remains unchanged for these rows.*			Y
Dual Beacon				Y
Dual CTS Protection				Y
STBC Beacon				Y
L-SIG TXOP Protection Full Support				Y
PCO Active				Y
PCO Phase				Y
Basic MCS Set	Indicates the MCS values that are supported by all HT STAs in the BSS. Present in Beacon/Probe Response/Mesh Peering Open/Mesh Peering Confirm frames. Otherwise reserved.	*The text in these table cells remains unchanged for these rows.*		N

7.3.2.92 Interworking ~~information~~ element

Change the fourth to eighth paragraphs of 7.3.2.92 as follows:

A non-AP STA or mesh STA sets Internet, ASRA and UESA fields to 0 when including the Interworking element in the Probe Request frame. A non-AP STA sets the Internet, and ASRA bits to 0 when including the Interworking element in (Re)association Request frames. A mesh STA sets the Internet bit to 0 when including the Interworking element in Mesh Peering Open frames. In (Re)association Request frames, a non-AP STA sets the UESA bit according to the procedures in 11.3.2. The Access Network Types are shown in Table 7-43bh. The Access Network Type field is set by the AP or the mesh STA to advertise its Access Network Type to non-AP STAs or mesh STAs. A non-AP STA or a mesh STA uses this field to indicate the desired Access Network Type in an active scan. See Annex X.1 for informative text on usage of fields contained within the Interworking element.

Bit 4 is the Internet field. The AP or mesh STA sets this field to 1 if the network provides connectivity to the Internet; otherwise it is set to 0 indicating that it is unspecified whether the network provides connectivity to the Internet.

Bit 5 is the Additional Step Required for Access (ASRA) field. It is set to 1 by the AP to indicate that the network requires a further step for access. It is set to 0 whenever dot11RSNAEnabled is true. For more information, refer to Network Authentication Type Information in 7.3.4.5. For a mesh STA the ASRA field is used as an emergency indicator. If a mesh STA requires emergency services, the ASRA field is set to 1, otherwise it is set to 0. See 11.23.6.

Bit 6 is the ESR (emergency services reachable) field. It is set to 1 by the AP or mesh STA to indicate that emergency services are reachable through the AP or mesh STA; otherwise, it is set to 0 indicating that it is ~~unspecified whether emergency services are reachable~~ unable to reach the emergency services.~~,~~ ~~s~~See 11.23.6.

Bit 7 is the UESA (unauthenticated emergency service accessible) field. When ~~the AP sets it~~ equal to 0, this field indicates that no unauthenticated emergency services are reachable through this AP or mesh STA. When ~~set~~ equal to 1, this field indicates that higher layer unauthenticated emergency services are reachable through this AP or mesh STA. A STA uses the Interworking ~~information~~ element with the UESA bit ~~set~~ equal to 1 to gain unauthenticated access to a BSS to access emergency services. A mesh STA uses the Interworking element with the UESA bit equal to 1 to gain unauthenticated access to another mesh STA to access emergency services. See 11.3.2.

Change the tenth paragraph of 7.3.2.92 as follows:

The HESSID field, which is the identifier for a homogeneous ESS, specifies the value of HESSID, see 11.23.2. A STA uses this field to indicate the desired HESSID in an active scan per 11.1.3. The HESSID field for an AP is set to the value of dot11HESSID. This optional field is not used by mesh STAs.

Insert the following new subclauses after 7.3.2.97:

7.3.2.98 Mesh Configuration element

7.3.2.98.1 General

The Mesh Configuration element shown in Figure 7-95o130 is used to advertise mesh services. It is contained in Beacon frames and Probe Response frames transmitted by mesh STAs, and is also contained in Mesh Peering Open and Mesh Peering Confirm frames.

	Element ID	Length	Active Path Selection Protocol Identifier	Active Path Selection Metric Identifier	Congestion Control Mode Identifier	Synchronization Method Identifier	Authentication Protocol Identifier	Mesh Formation Info	Mesh Capability
Octets:	1	1	1	1	1	1	1	1	1

Figure 7-95o130—Mesh Configuration element format

The Element ID is set to the value given in Table 7-26 for this element.

The Length field is set to 7.

The remainder of the fields are described in the following subclauses.

7.3.2.98.2 Active Path Selection Protocol Identifier

The Active Path Selection Protocol Identifier field indicates the path selection protocol that is currently activated in the MBSS. Table 7-43bj1 provides path selection protocol identifier values defined by this standard.

Table 7-43bj1—Active Path Selection Protocol Identifier field values

Value	Meaning
0	Reserved

Table 7-43bj1—Active Path Selection Protocol Identifier field values *(continued)*

Value	Meaning
1	Hybrid wireless mesh protocol (default path selection protocol) defined in 11C.9 (default path selection protocol)
2–254	Reserved
255	Vendor specific (The active path selection protocol is specified in a Vendor Specific element)

When the Active Path Selection Protocol Identifier field is 255, the active path selection protocol is specified by a Vendor Specific element that is present in the frame. The content of the Vendor Specific element is beyond the scope of this standard. (See 7.3.2.26.)

7.3.2.98.3 Active Path Selection Metric Identifier

The Active Path Selection Metric Identifier field indicates the path metric that is currently used by the active path selection protocol in the MBSS. Table 7-43bj2 provides the path selection metric identifier values defined by this standard.

Table 7-43bj2—Active Path Selection Metric Identifier field values

Value	Meaning
0	Reserved
1	Airtime link metric defined in 11C.8 (default path selection metric)
2–254	Reserved
255	Vendor specific (The active metric is specified in a Vendor Specific element)

When the Active Path Selection Metric Identifier field is 255, the active path metric is specified by a Vendor Specific element that is present in the frame. The content of the Vendor Specific element is beyond the scope of this standard. (See 7.3.2.26.)

7.3.2.98.4 Congestion Control Mode Identifier

The Congestion Control Mode Identifier field indicates the congestion control protocol that is currently activated in the MBSS. Table 7-43bj3 provides the congestion control mode identifier values defined by this standard.

The congestion mode identifier value of 0 indicates the mesh STA has no active congestion control protocol, and is set as the default value for the congestion control mode identifier in the MBSS.

When the Congestion Control Mode Identifier field is 255, the active congestion control protocol is specified by a Vendor Specific element that is present in the frame. The content of the Vendor Specific element is beyond the scope of this standard. (See 7.3.2.26.)

Table 7-43bj3—Congestion Control Mode Identifier field values

Value	Meaning
0	Congestion control is not activated (default congestion control mode)
1	Congestion control signaling protocol defined in 11C.11.2
2–254	Reserved
255	Vendor specific (The active congestion control protocol is specified in a Vendor Specific element)

7.3.2.98.5 Synchronization Method Identifier

The Synchronization Method Identifier field indicates the synchronization method that is currently activated in the MBSS. Table 7-43bj4 provides the synchronization method identifier values defined by this standard.

Table 7-43bj4—Synchronization Method Identifier field values

Value	Meaning
0	Reserved
1	Neighbor offset synchronization method defined in 11C.12.2.2 (default synchronization method)
2–254	Reserved
255	Vendor specific (The active synchronization method is specified in a Vendor Specific element)

The neighbor offset synchronization method is defined as the default synchronization method among mesh STAs. The details of the neighbor offset synchronization method are described in 11C.12.2.2.

When the Synchronization Method Identifier field is 255, the active synchronization method is specified by a Vendor Specific element that is present in the frame. The content of the Vendor Specific element is beyond the scope of this standard. (See 7.3.2.26.)

7.3.2.98.6 Authentication Protocol Identifier

The Authentication Protocol Identifier field indicates the type of authentication protocol that is currently used to secure the MBSS. Table 7-43bj5 provides the authentication protocol identifier values defined by this standard.

When the Authentication Protocol Identifier field is 255, the active authentication protocol is specified by a Vendor Specific element that is present in the frame. The content of the Vendor Specific element is beyond the scope of this standard. (See 7.3.2.26.)

7.3.2.98.7 Mesh Formation Info

The format of the Mesh Formation Info field is shown in Figure 7-95o131.

Table 7-43bj5—Authentication Protocol Identifier field values

Value	Meaning
0	No authentication method is required to establish mesh peerings within the MBSS
1	SAE defined in 8.2a
2	IEEE 802.1X authentication
3–254	Reserved
255	Vendor specific (The active authentication protocol is specified in a Vendor Specific element)

	B0	B1 B6	B7
	Connected to Mesh Gate	Number of Peerings	Connected to AS
Bits:	1	6	1

Figure 7-95o131—Mesh Formation Info field

The Connected to Mesh Gate subfield is set to 1, if the mesh STA has a mesh path to a mesh gate that announces its presence using GANN elements, RANN elements, or PREQ elements, and set to 0 otherwise.

The Number of Peerings subfield contains an unsigned integer that indicates the number of mesh peerings currently maintained by the mesh STA or 63, whichever is smaller.

The Connected to AS subfield is set to 1 if the Authentication Protocol Identifier field in the Mesh Configuration element is set to 2 (indicating IEEE 802.1X authentication) and the mesh STA has an active connection to an AS.

NOTE—When an AS is collocated with an IEEE 802.1X authenticator an active connection is implicitly true.

7.3.2.98.8 Mesh Capability

The Mesh Capability field comprises a set of values indicating whether a mesh STA is a possible candidate for mesh peering establishment. The details of the Mesh Capability field are shown in Figure 7-95o132.

	B0	B1	B2	B3	B4	B5	B6	B7
	Accepting Additional Mesh Peerings	MCCA Supported	MCCA Enabled	Forward-ing	MBCA Enabled	TBTT Adjusting	Mesh Power Save Level	Reserved
Bits:	1	1	1	1	1	1	1	1

Figure 7-95o132—Mesh Capability field

The Accepting Additional Mesh Peerings subfield is set to 1 if the mesh STA is willing to establish additional mesh peerings with other mesh STAs and set to 0 otherwise (i.e., the Accepting Additional Mesh Peerings subfield is set in accordance with dot11MeshAcceptingAdditionalPeerings). When the Mesh

Configuration element is included in the Mesh Peering Open frame and in the Mesh Peering Confirm frame, the Accepting Additional Mesh Peerings subfield is set to 1.

The MCCA Supported subfield is set to 1 if the mesh STA implements MCCA and set to 0 otherwise (i.e., the MCCA Supported subfield is set in accordance with dot11MCCAImplemented).

The MCCA Enabled subfield is set to 1 if the mesh STA is using the MCCA and set to 0 otherwise (i.e., the MCCA Enabled subfield is set in accordance with dot11MCCAActivated).

The Forwarding subfield is set to 1 if the mesh STA forwards MSDUs and set to 0 otherwise (i.e., the Forwarding subfield is set in accordance with dot11MeshForwarding).

The MBCA Enabled subfield is set to 1 if the mesh STA is using MBCA, and is set to 0 otherwise (i.e., the MBCA Enabled subfield is set in accordance with dot11MBCAActivated). (See 11C.12.4.)

The TBTT Adjusting subfield is set to 1 while the TBTT adjustment procedure is ongoing, and is set to 0 otherwise. (See 11C.12.4.4.3.)

The Mesh Power Save Level subfield is set to 1 if at least one of the peer-specific mesh power modes is deep sleep mode and set to 0 otherwise. The Mesh Power Save Level subfield is reserved when the Power Management field in the Frame Control Field is set to 0. See 7.1.3.5.10.

7.3.2.99 Mesh ID element

The Mesh ID element is used to advertise the identification of an MBSS and is described in 11C.2.2. The format of the Mesh ID element is shown in Figure 7-95o133. The Mesh ID element is transmitted in Mesh Peering Open frames, Mesh Peering Confirm frames, Mesh Peering Close frames, Beacon frames, and Probe Request and Response frames.

	Element ID	Length	Mesh ID
Octets:	1	1	0-32

Figure 7-95o133—Mesh ID element format

The Element ID is set to the value given in Table 7-26 for this element.

The length of the Mesh ID field is between 0 and 32 octets. A Mesh ID field of length 0 indicates the wildcard Mesh ID, which is used within Probe Request frame.

Detailed usage of the Mesh ID element is described in 11C.2.2.

7.3.2.100 Mesh Link Metric Report element

The Mesh Link Metric Report element is transmitted by a mesh STA to a neighbor peer mesh STA to indicate the quality of the link between the transmitting mesh STA and the neighbor peer mesh STA. The format of the Mesh Link Metric Report element is shown in Figure 7-95o134.

	Element ID	Length	Flags	Link Metric
Octets:	1	1	1	variable

Figure 7-95o134—Mesh Link Metric Report element format

The Element ID is set to the value given in Table 7-26 for this element.

The Length field indicates the number of octets in the Information field (fields following the Element ID and Length fields).

The format of the Flags field is shown in Figure 7-95o135.

B0	B1 B7
Request	Reserved
Bits: 1	7

Figure 7-95o135—Flags field

The Flags field is set as follows:

- Bit 0: Request subfield (0 = not a request, 1 = link metric report request). A Request subfield equal to 1 indicates that the recipient of Mesh Link Metric Report element is requested to send a link metric report to the transmitter of the Mesh Link Metric Report element.
- Bit 1–7: Reserved.

The Link Metric field indicates the value of the link metric associated with the mesh link between the peer mesh STA transmitting the Mesh Link Metric Report and the neighbor mesh STA receiving the Mesh Link Metric Report. The length and the data type of the Link Metric field are determined by the active path selection metric identifier (see 7.3.2.98.3). The length and the data type for the airtime link metric are given in Figure 11C-5 in 11C.8.

7.3.2.101 Congestion Notification element

The Congestion Notification element is used to indicate the congestion status of the mesh STA per mesh destination and AC, and the duration for which the STA expects the congestion to last. The format of the Congestion Notification element is shown in Figure 7-95o136. The Congestion Notification element is included in Congestion Control Notification frames as described in 7.4.15.5.

Element ID	Length	Destination Mesh STA Address	Congestion Notification Duration Timer (AC_BK)	Congestion Notification Duration Timer (AC_BE)	Congestion Notification Duration Timer (AC_VI)	Congestion Notification Duration Timer (AC_VO)
Octets: 1	1	6	2	2	2	2

Figure 7-95o136—Congestion Notification element format

The Element ID is set to the value given in Table 7-26 for this element.

The Length field is set to 14.

The Destination Mesh STA Address field is represented as a 48-bit MAC address and is set to the address of the mesh destination for which the intra-mesh congestion control is applied. It is set to the broadcast address if the intra-mesh congestion control is applied to all destinations.

The element contains four Congestion Notification Duration fields for the four EDCA access categories to indicate the estimated congestion duration per AC at the mesh STA transmitting the congestion notification. The congestion notification duration values are encoded as unsigned integers in units of 100 µs.

7.3.2.102 Mesh Peering Management element

The Mesh Peering Management element is used to manage a mesh peering with a neighbor mesh STA. The format of the Mesh Peering Management element is shown in Figure 7-95o137.

Element ID	Length	Mesh Peering Protocol Identifier	Local Link ID	Peer Link ID (conditional)	Reason Code (conditional)	Chosen PMK (optional)
Octets: 1	1	2	2	2	2	16

Figure 7-95o137—Mesh Peering Management element format

The Element ID is set to the value given in Table 7-26 for this element.

The Length field is set to the number of octets in the Mesh Peering Management element following the Length field itself.

The Mesh Peering Protocol Identifier field indicates the type of mesh peering protocol that is currently used to establish mesh peerings. Table 7-43bj6 provides the mesh peering protocol identifier values defined by this standard.

Table 7-43bj6—Mesh Peering Protocol Identifier field values

Value	Meaning
0	Mesh peering management protocol
1	Authenticated mesh peering exchange protocol
2–254	Reserved
255	Vendor specific (The active mesh peering protocol is specified in a Vendor Specific element)

When the Mesh Peering Protocol Identifier field is 255, the active mesh peering protocol is specified by a Vendor Specific element that is present in the frame. The content of the Vendor Specific element is beyond the scope of this standard. (See 7.3.2.26.)

The Local Link ID field is the unsigned integer value generated by the local mesh STA to identify the mesh peering instance.

The conditional components of the Mesh Peering Management element are present depending on the Action field value of the frame in which the Mesh Peering Management element is conveyed.

The Peer Link ID field is the unsigned integer value generated by the peer mesh STA to identify the mesh peering instance. This field is not present for the Mesh Peering Open frame, is present for the Mesh Peering

Confirm frame, and is optionally present for the Mesh Peering Close frame. The presence or absence of the Peer Link ID in a Mesh Peering Close is inferred by the Length field.

The Reason Code field enumerates reasons for sending a Mesh Peering Close. It is present for the Mesh Peering Close frame and is not present for Mesh Peering Open or Mesh Peering Confirm frames. The reason code is defined in 7.3.1.7.

The Chosen PMK field is present when dot11MeshSecurityEnabled is true and a PMK is shared between the transmitter and receiver of the frame containing the element. It contains the PMKID that identifies the PMK used to protect the Mesh Peering Management frame.

Detailed usage of the Mesh Peering Management element is described in 11C.3.6, 11C.3.7, 11C.3.8, and 11C.5.5.

7.3.2.103 Mesh Channel Switch Parameters element

The Mesh Channel Switch Parameters element is used together with Channel Switch Announcement element and Extended Channel Switch Announcement element by a mesh STA in an MBSS to advertise to other mesh STAs when it is changing to a new operating channel and/or regulatory class. The format of the Mesh Channel Switch Parameters element is shown in Figure 7-95o138.

	Element ID	Length	Time To Live	Flags	Reason Code	Precedence Value
Octets:	1	1	1	1	2	2

Figure 7-95o138—Mesh Channel Switch Parameters element format

The Element ID is set to the value given in Table 7-26 for this element.

The Length field is set to 6.

The Time To Live field is coded as an unsigned integer and indicates the remaining number of hops allowed for this element.

The Flags field indicates the attribute of this channel switch attempt. The format of the Flags field is shown in Figure 7-95o139.

	B0	B1	B2	B3 B7
	Transmit Restrict	Initiator	Reason	Reserved
Bits:	1	1	1	5

Figure 7-95o139—Flags field

The Transmit Restrict subfield is set to 1 when the mesh STA asks neighboring peer mesh STAs not to transmit further frames except frames containing Mesh Channel Switch Parameters element on the current channel until the scheduled channel switch. The Transmit Restrict subfield is set to 0 otherwise.

The Initiator subfield is set to 1 when the mesh STA initiates this channel switch attempt. The Initiator subfield is set to 0 when this channel switch attempt is initiated by another mesh STA and propagated by the current mesh STA.

The Reason subfield indicates the validity of the Reason Code field. It is set to 1 if the Reason Code field is valid, and is set to 0 otherwise. When the Reason subfield is 0, the content of the Reason Code field is reserved.

The Reason Code field specifies the reason for the mesh channel switch. The Reason Code is defined in 7.3.1.7. The content of the Reason Code field is valid only when Reason subfield of Flags field is set to 1, and is reserved otherwise.

The Precedence Value field is coded as unsigned integer and is set to a random value in the range 0 to 65535 determined by the initiator of this channel switch attempt.

The Mesh Channel Switch Parameters element is included in Channel Switch Announcement frames, as described in 7.4.1.5, and Extended Channel Switch Announcement frames, as described in 7.4.7.6. During MBSS Channel Switch, the Mesh Channel Switch Parameters element is included in Beacon frames, as described in 7.2.3.1, and Probe Response frames, as described in 7.2.3.9, until scheduled channel switch.

7.3.2.104 Mesh Awake Window element

The Mesh Awake Window element is present in DTIM Beacon frames and is optionally present in Beacon and Probe Response frames. The format of the Mesh Awake Window element is shown in Figure 7-95o140.

	Element ID	Length	Mesh Awake Window
Octets:	1	1	2

Figure 7-95o140—Mesh Awake Window element format

The Element ID is set to the value given in Table 7-26 for this element.

The Length field is set to 2.

The Mesh Awake Window field is 2 octets long and contains an unsigned integer that indicates the duration of the mesh awake window in TUs.

7.3.2.105 Beacon Timing element

The Beacon Timing element is used to advertise the beacon timing information of neighbor STAs (mesh STAs, APs, or STAs in an IBSS). The format of the Beacon Timing element is shown in Figure 7-95o141.

	Element ID	Length	Report Control	Beacon Timing Information #1	...	Beacon Timing Information #N
Octets:	1	1	1	6	...	6

Figure 7-95o141—Beacon Timing element format

The Element ID is set to the value given in Table 7-26 for this element.

The Length field indicates the number of octets in the Information field (fields following the Element ID and Length fields).

The Report Control field is used to signal information about the beacon timing information tuple contained in the Beacon Timing element. The structure of the Report Control field is defined in Figure 7-95o142.

	Status Number	Beacon Timing Element Number	More Beacon Timing Elements
Bits:	4	3	1

Figure 7-95o142—Report Control field

The Status Number subfield is set to the status number of the beacon timing information set. The status number is managed as described in 11C.12.4.2.4.

The Beacon Timing Element Number subfield is an unsigned integer that indicates the index of the beacon timing information tuple contained in this Beacon Timing element. The Beacon Timing Element Number is set to 0 in the Beacon Timing element for the first or only tuple of the beacon timing information and is incremented by one for each successive tuple of the beacon timing information. The beacon timing information tuples are managed as described in 11C.12.4.2.5.

The More Beacon Timing Element subfield is set to 1 if a successive tuple of beacon timing information exists, and set to 0 otherwise.

The Beacon Timing Information field contains the beacon timing information of a neighbor STA. When the mesh STA reports multiple beacon timing information, multiple Beacon Timing Information fields are included in the Beacon Timing element. The structure of the Beacon Timing Information field is defined in Figure 7-95o143.

	Neighbor STA ID	Neighbor TBTT	Neighbor Beacon Interval
Octets:	1	3	2

Figure 7-95o143—Beacon Timing Information field

The Neighbor STA ID subfield is an unsigned integer that indicates the identification of the neighbor STA corresponding to this beacon timing information. When a mesh peering is established with this neighbor STA, the MSB of this field is set to 0, and the rest of this field is set to the last 7 digits (7 LSBs) of the AID value assigned to this neighbor mesh STA. When a mesh peering is not established with this neighbor STA, the MSB of this field is set to 1, and the rest of this field is set to the last 7 digits (7 LSBs, taking the I/G bit as the MSB) of the 48-bit MAC address of this neighbor STA.

NOTE—Since the Neighbor STA ID subfield is provided in abbreviated form, it is possible that the same Neighbor STA ID value appears in multiple Beacon Timing Information fields.

The Neighbor TBTT subfield is an unsigned integer that indicates a TBTT of the corresponding neighbor STA, measured in the local TSF timer of the mesh STA. The value is indicated in multiples of 32 μs. When the active synchronization method is the neighbor offset synchronization method, the TBTT is calculated as described in 11C.12.4.2.2. The B5 to the B28 (taking the B0 as the LSB) of the calculated TBTT are contained in this subfield.

The Neighbor Beacon Interval subfield is an unsigned integer that indicates the beacon interval being used by the corresponding neighbor STA. The unit of the Neighbor Beacon Interval subfield is TU.

Detailed usage of the Beacon Timing element is described in 11C.12.4.2.

7.3.2.106 MCCAOP Setup Request element

7.3.2.106.1 General

The MCCAOP Setup Request element is used to make an MCCAOP reservation. This element is transmitted in individually addressed MCCA Setup Request frames or in group addressed MCCA Setup Request frames. The mesh STA transmitting the MCCA Setup Request element is the MCCAOP owner of the MCCAOPs that will be scheduled with this reservation setup request. The receivers of the MCCAOP Setup Request are the MCCAOP responders. The format of the element is shown in Figure 7-95o144.

	Element ID	Length	MCCAOP Reservation ID	MCCAOP Reservation
Octets:	1	1	1	5

Figure 7-95o144—MCCAOP Setup Request element format

The Element ID is set to the value given in Table 7-26 for this element.

The Length field is set to 6.

The MCCAOP Reservation ID field is an eight bit unsigned integer that represents the ID for the MCCAOP reservation. It is determined by the MCCAOP owner. When used in combination with the MAC address of the MCCAOP owner, the MCCAOP Reservation ID uniquely identifies the MCCAOP reservation. If this MCCAOP Setup Request is for an individually addressed transmission, the MCCAOP Reservation ID is between 0 and 127 and the MCCAOP Setup Request element is transmitted in an individually addressed frame to the intended responder. If this MCCAOP Setup Request is for a group addressed transmission, the MCCAOP Reservation ID is between 128 and 254 and the MCCAOP Setup Request element is transmitted in a group addressed frame. The value 255 is not used to identify a single MCCAOP reservation.

The MCCAOP Reservation field is described in 7.3.2.106.2.

7.3.2.106.2 MCCAOP Reservation field

The MCCAOP Reservation field is a 5 octet field specifying a schedule for frame transmissions called MCCAOPs. The MCCAOP Reservation field consists of three subfields and its format is shown in Figure 7-95o145.

	MCCAOP Duration	MCCAOP Periodicity	MCCAOP Offset
Octets:	1	1	3

Figure 7-95o145—MCCAOP Reservation field

The MCCAOP Duration subfield is one octet in length and contains an unsigned integer. It specifies the duration of the MCCAOPs in multiples of 32 µs.

The MCCAOP Periodicity subfield is one octet in length and contains a positive integer. It specifies the number of MCCAOPs scheduled in each DTIM interval.

The MCCAOP Offset subfield is three octets in length and contains an unsigned integer. It specifies the beginning of the first MCCAOP in each DTIM interval. The value is specified in multiples of 32 µs. The sum of MCCAOP Offset plus MCCAOP Duration is constrained to be smaller than the duration of the DTIM interval divided by MCCAOP Periodicity.

7.3.2.107 MCCAOP Setup Reply element

The MCCAOP Setup Reply element is used to reply to an MCCAOP Setup Request. This element is transmitted in individually addressed MCCA Setup Reply frames. The mesh STA transmitting the MCCA Setup Reply element is the MCCAOP responder of the MCCAOPs scheduled in this reservation setup. The receiver of the MCCAOP Setup Reply is the MCCAOP owner. The format of the element is shown in Figure 7-95o146.

	Element ID	Length	MCCAOP Reservation ID	MCCA Reply Code	MCCAOP Reservation
Octets:	1	1	1	1	0 or 5

Figure 7-95o146—MCCAOP Setup Reply element format

The Element ID is set to the value given in Table 7-26 for this element.

The Length field is set to 2 or 7 octets.

The MCCAOP Reservation ID field is an eight bit unsigned integer that represents the ID for the requested series of MCCAOPs. It is determined by the MCCAOP owner and copied from the MCCAOP Setup Request element. When used in combination with the MAC address of the MCCAOP owner, the MCCAOP Reservation ID uniquely identifies the MCCAOP reservation. If this MCCAOP Setup Request is for an individually addressed transmission, the MCCAOP Reservation ID is between 0 and 127. If this MCCAOP Setup Request is for a group addressed transmission, the MCCAOP Reservation ID is between 128 to 254. The value 255 is not used to identify a single MCCAOP reservation.

The MCCA Reply Code field is a one octet field that contains the reply code used in an MCCAOP Setup Reply element. The reply codes are defined in Table 7-43bj7.

Table 7-43bj7—MCCA Reply Code field values

MCCA reply code	Meaning
0	Accept
1	Reject: MCCAOP reservation conflict
2	Reject: MAF limit exceeded
3	Reject: MCCA track limit (dot11MCCAMaxTrackStates) exceeded
4–255	Reserved

The MCCAOP Reservation field includes an alternative to the MCCAOP reservation specified in the MCCAOP Setup Request message. Its format is described in 7.3.2.106.2. When the MCCA Reply Code is 1, the MCCAOP Reservation field might be present. When the MCCA Reply Code is set to other values, the MCCAOP Reservation field is not present.

7.3.2.108 MCCAOP Advertisement Overview element

The MCCAOP Advertisement Overview element is used by a mesh STA to advertise its MCCA Information and information about its MCCAOP Advertisement elements, representing its MCCAOP advertisement set, to its neighbors. This element is transmitted in MCCA Advertisement frames and optionally present in Beacon frames. The format of the MCCAOP Advertisement Overview element is shown in Figure 7-95o147.

Element ID	Length	Advertisement Set Sequence Number	Flags	MCCA Access Fraction	MAF Limit	Advertisement Elements Bitmap
Octets: 1	1	1	1	1	1	2

Figure 7-95o147—MCCAOP Advertisement Overview element format

The Element ID is set to the value given in Table 7-26 for this element.

The Length field is set to 6.

The Advertisement Set Sequence Number field is 1 octet in length and is coded as an unsigned integer. It is set to the advertisement set sequence number of the current MCCAOP advertisement set. The Advertisement Set Sequence Number, together with the MAC address of the transmitter of the MCCAOP Advertisement Overview element, identifies an MCCAOP advertisement set and provides an identifier and a chronological order of different MCCAOP advertisement sets of the same mesh STA.

The format of the Flags field is shown in Figure 7-95o148.

B0	B1 B7
Accept Reservations	Reserved
Bits: 1	7

Figure 7-95o148—Flags field format

The Flags field is set as follows:

- — Bit 0: Accept Reservations subfield. The Accept Reservations subfield is set to 1 if the mesh STA accepts additional reservations. It is set to 0 otherwise.
- — Bit 1–7: Reserved

The MCCA Access Fraction field is an eight bit unsigned integer. The MCCA Access Fraction field is set to the current value of the MCCA access fraction at the mesh STA rounded down (floor) to the nearest multiple of (1/255) of the DTIM interval length.

The MAF Limit field is an eight bit unsigned integer. The MAF Limit field is set to the maximum MCCA access fraction allowed at the mesh STA rounded down (floor) to the nearest multiple of (1/255) of the DTIM interval length.

The Advertisement Elements Bitmap field is 2 octets in length and indicates the MCCAOP Advertisement elements that are part of this MCCAOP advertisement set. The Advertisement Elements Bitmap field is a bitmap. Bit *i* in this bitmap equals 1 if the MCCAOP Advertisement element with MCCAOP Advertisement Element Index equal to *i* is part of this MCCAOP advertisement set, and it equals 0 otherwise.

7.3.2.109 MCCAOP Advertisement element

7.3.2.109.1 General

The MCCAOP Advertisement element is used by a mesh STA to advertise MCCAOP reservations to its neighbors. This element is transmitted in MCCA Advertisement frames and optionally present in Beacon frames. The format of the element is shown in Figure 7-95o149.

Element ID	Length	Advertisement Set Sequence Number	MCCAOP Advertisement Element Information	TX-RX Periods Report	Broadcast Periods Report	Interference Periods Report
Octets: 1	1	1	1	variable	variable	variable

Figure 7-95o149—MCCAOP Advertisement element format

The Element ID is set to the value given in Table 7-26 for this element.

The Length field is set to 2 to 255 octets.

The Advertisement Set Sequence Number field is 1 octet in length and is coded as an unsigned integer. It is set to the advertisement set sequence number of the current MCCAOP advertisement set. The Advertisement Set Sequence Number, together with the MAC address of the transmitter of the MCCAOP Advertisement element, identifies an MCCAOP advertisement set and provides an identifier and a chronological order of different MCCAOP advertisement sets of the same mesh STA.

The MCCAOP Advertisement Element Information field is 1 octet in length. It is described in 7.3.2.109.2.

The TX-RX Periods Report field is a variable length field that contains an MCCAOP Reservation Report field as described in 7.3.2.109.3. This field is only present when the TX-RX Report Present subfield of the MCCAOP Advertisement Element Information field is equal to 1. The TX-RX Periods Report field is described in 9.9a.3.7.2.

The Broadcast Periods Report field is a variable length field that contains an MCCAOP Reservation Report field as described in 7.3.2.109.3. This field is only present when the Broadcast Report Present subfield of the MCCAOP Advertisement Element Information field is equal to 1. The Broadcast Periods Report field is described in 9.9a.3.7.2.

The Interference Periods Report field is a variable length field that contains an MCCAOP Reservation Report field as described in 7.3.2.109.3. This field is only present when the Interference Report Present subfield of the MCCAOP Advertisement Element Information field is equal to 1. The Interference Periods Report field is described in 9.9a.3.7.2.

7.3.2.109.2 MCCAOP Advertisement Element Information field

The MCCA Information field is 1 octets in length and provides information on the MCCAOP reservations. The field consists of four subfields and its format is shown in Figure 7-95o150.

B0 B3	B4	B5	B6	B7
MCCAOP Advertisement Element Index	TX-RX Report Present	Broadcast Report Present	Interference Report Present	Reserved
Bits: 4	1	1	1	1

Figure 7-95o150—MCCAOP Advertisement Element Information field

The MCCAOP Advertisement Element Index subfield is a 4-bit unsigned integer. It identifies the MCCAOP Advertisement element.

The TX-RX Report Present subfield is 1 bit in length. It is set to 1 if the TX-RX Periods Report field is present in the MCCAOP Advertisement element and set to 0 if no TX-RX Periods Report field is present.

The Broadcast Report Present subfield is 1 bit in length. It is set to 1 if the Broadcast Periods Report field is present in the MCCAOP Advertisement element and set to 0 if no Broadcast Periods Report field is present.

The Interference Report Present subfield is 1 bit in length. It is set to 1 if the Interference Periods Report field is present in the MCCAOP Advertisement element and set to 0 if no Interference Periods Report field is present.

7.3.2.109.3 MCCAOP Reservation Report field

The MCCAOP Reservation Report field is of variable length and is used to report a number of MCCAOP reservations. The field consists of a variable number of subfields and its format is shown in Figure 7-95o151.

Number of Reported MCCAOP Reservations	MCCAOP Reservation 1	...	MCCAOP Reservation n
Octets: 1	5		5

Figure 7-95o151—MCCAOP Reservation Report field

The Number of Reported MCCAOP reservations is a field of one octet with an unsigned integer that specifies the number, n, of MCCAOP Reservations reported in this MCCAOP Reservation Report field.

The MCCAOP Reservation 1 through MCCAOP Reservation n fields specify the MCCAOP reservations reported. Each MCCAOP Reservation field is 5 octets in length and its format is shown in Figure 7-95o145 in 7.3.2.106.2.

7.3.2.110 MCCAOP Teardown element

The MCCAOP Teardown element is used to announce the teardown of an MCCAOP reservation. The MCCAOP Teardown element is transmitted in individually addressed MCCA Teardown frames or in group addressed MCCA Teardown frames. Its format is shown in Figure 7-95o152.

	Element ID	Length	MCCAOP Reservation ID	MCCAOP Owner
Octets:	1	1	1	0 or 6

Figure 7-95o152—MCCAOP Teardown element format

The Element ID is set to the value given in Table 7-26 for this element.

The Length is variable and set to 1 or 7 octets.

An MCCAOP Teardown element is transmitted by either the MCCAOP owner or the MCCAOP responder of a MCCAOP reservation to tear down the MCCAOP reservation.

The MCCAOP Reservation ID field is an eight bit unsigned integer that represents the ID for the MCCAOP reservation.

The MCCAOP Owner field is an optional field. It is 6 octets long and indicates the 48-bit MAC address of the MCCAOP owner. This field is only included if the element is transmitted by the MCCAOP responder.

7.3.2.111 GANN element

The Gate Announcement (GANN) element is used for announcing the presence of a mesh gate in the MBSS. The GANN element is transmitted in a Gate Announcement frame (see 7.4.15.4). The format of the GANN element is shown in Figure 7-95o153.

	Element ID	Length	Flags	Hop Count	Element TTL	Mesh Gate Address	GANN Sequence Number	Interval
Octets:	1	1	1	1	1	6	4	2

Figure 7-95o153—GANN element format

The Element ID is set to the value given in Table 7-26 for this element.

The Length field is set to 15.

The Flags field is reserved.

The Hop Count field is coded as an unsigned integer and indicates the number of hops from the originating mesh gate to the mesh STA transmitting this element.

The Element TTL field is coded as an unsigned integer and indicates the remaining number of hops allowed for this element.

The Mesh Gate Address field is represented as a 48-bit MAC address and is set to the MAC address of the mesh gate.

The GANN Sequence Number field is coded as an unsigned integer and is set to a GANN Sequence Number specific for the originating mesh gate.

The Interval field is coded as an unsigned integer and is set to the number of seconds between the periodic transmissions of Gate Announcements by the mesh gate.

Detailed usage of the GANN element is described in 11C.10.2.

7.3.2.112 RANN element

The Root Announcement (RANN) element is used for announcing the presence of a mesh STA configured as root mesh STA with dot11MeshHWMProotMode set to rann (4). RANN elements are sent out periodically by such a root mesh STA. The RANN element is transmitted in an HWMP Mesh Path Selection frame (see 7.4.15.3). The format of the RANN element is shown in Figure 7-95o154.

Element ID	Length	Flags	Hop Count	Element TTL	Root Mesh STA Address	HWMP Sequence Number	Interval	Metric
Octets: 1	1	1	1	1	6	4	4	4

Figure 7-95o154—RANN element format

The Element ID is set to the value given in Table 7-26 for this element.

The length is set to 21.

The format of the Flags field is shown in Figure 7-95o155.

B0	B1 B7
Gate Announcement	Reserved
Bits: 1	7

Figure 7-95o155—Flags field format

The Flags field is set as follows:

- Bit 0: Gate Announcement subfield (0 = gate announcement protocol not activated, 1 = gate announcement protocol activated). A Gate Announcement subfield equal to 1 indicates that the Root Mesh STA Address is a mesh gate with dot11MeshGateAnnouncementProtocol equal to true.
- Bit 1–7: Reserved.

The Hop Count field is coded as an unsigned integer and indicates the number of hops from the originating root mesh STA to the mesh STA transmitting this element.

The Element TTL field is coded as an unsigned integer and indicates the remaining number of hops allowed for this element.

The Root Mesh STA Address field is represented as a 48-bit MAC address and is set to the MAC address of the root mesh STA.

The HWMP Sequence Number is coded as an unsigned integer and is set to the HWMP sequence number specific to the root mesh STA.

The Interval field is coded as an unsigned integer and is set to the number of TUs between the periodic transmissions of Root Announcements.

The Metric field is set to the cumulative metric from the originating root mesh STA to the mesh STA transmitting the announcement.

Detailed usage of the RANN element is described in 11C.9.12.

7.3.2.113 PREQ element

The Path Request (PREQ) element is used for discovering a path to one or more target mesh STAs, path maintenance (optional), building a proactive (reverse) path selection tree to the root mesh STA, and confirming a path to a target mesh STA (optional). The PREQ element is transmitted in an HWMP Mesh Path Selection frame (see 7.4.15.3). The format of the PREQ element is shown in Figure 7-95o156.

Element ID	Length	Flags	Hop Count	Element TTL	Path Discovery ID	Originator Mesh STA Address	Originator HWMP Sequence Number	Originator External Address	Lifetime
Octets: 1	1	1	1	1	4	6	4	0 or 6	4

Metric	Target Count	Per Target Flags #1	Target Address #1	Target HWMP Sequence Number #1	...	Per Target Flags #N	Target Address #N	Target Sequence Number #N
4	1	1	6	4	...	1	6	4

Figure 7-95o156—PREQ element format

The Element ID is set to the value given in Table 7-26 for this element.

The length is set to 37 to 252 octets.

The format of the Flags field is shown in Figure 7-95o157.

The Flags field is set as follows:

— Bit 0: Gate Announcement subfield (0 = gate announcement protocol not activated, 1 = gate announcement protocol activated). A Gate Announcement subfield equal to 1 indicates that the Originator Mesh STA Address is a mesh gate with dot11MeshGateAnnouncementProtocol equal to true.

B0	B1	B2	B3 B5	B6	B7
Gate Announcement	Addressing Mode	Proactive PREP	Reserved	AE	Reserved
Bits: 1	1	1	3	1	1

Figure 7-95o157—Flags field format

- Bit 1: Addressing Mode subfield (0 = group addressed, 1 = individually addressed). When the Addressing Mode subfield is 0, the PREQ element is sent in an HWMP Mesh Path Selection frame that is group addressed to all neighbor peer mesh STAs. When the Addressing Mode subfield is 1, the PREQ element is sent in an HWMP Mesh Path Selection frame that is individually addressed to a neighbor peer mesh STA. Detailed addressing information is provided in 11C.9.7.
- Bit 2: Proactive PREP subfield (0 = off, 1 = on). The Proactive PREP subfield is only of relevance if the Target Address is the broadcast address (all ones). If equal to 1, every recipient of a PREQ with Target Address equal to the broadcast address replies with a PREP. If equal to 0, it will only reply under certain conditions (see 11C.9.4.2).
- Bit 3–5: Reserved.
- Bit 6: AE (Address Extension) subfield (1= external address present, 0 = otherwise). An AE subfield equal to 1 indicates that the field Originator External Address is present, and that the originator mesh STA is a proxy for this external address.
- Bit 7: Reserved.

The Hop Count field is coded as an unsigned integer and is set to the number of hops from the originator to the mesh STA transmitting this element.

The Element TTL field is coded as an unsigned integer and indicates the remaining number of hops allowed for this element.

The Path Discovery ID field is coded as an unsigned integer and is set to some unique ID for this PathDiscovery.

The Originator Mesh STA Address field is represented as a 48-bit MAC address and is set to the originator MAC address.

The Originator HWMP Sequence Number is coded as an unsigned integer and is set to the HWMP sequence number specific to the originator.

The Originator External Address field is the MAC address of an external STA proxied by the Originator. This field is only present if the AE subfield in the Flags field is set to 1 and is represented as a 48-bit MAC address.

The Lifetime field is coded as an unsigned integer and is set to the time for which mesh STAs receiving the PREQ consider the forwarding information to be valid. The lifetime is measured in TUs.

The Metric field is set to the cumulative metric from the originator to the mesh STA transmitting the PREQ.

The Target Count N field is coded as an unsigned integer and gives the number of targets (N) contained in this PREQ. The maximum value of N is 20. The Per Target Flags field, the Target Address field, and the Target HWMP Sequence Number field are repeated N times in the element.

The format of the Per Target Flags field is shown in Figure 7-95o158.

	B0	B1	B2	B3 B7
	TO	Reserved	USN	Reserved
Bits:	1	1	1	5

Figure 7-95o158—Per Target Flags field format

The Per Target Flags field is set as follows:

— Bit 0: TO (Target Only) subfield: The TO subfield defines which mesh STA responds with a PREP element to the PREQ element containing an individual target address. If TO = 1, only the target mesh STA responds with an individually addressed PREP. If TO = 0, intermediate mesh STAs with active forwarding information to the target mesh STA also respond.

— Bit 1: Reserved.

— Bit 2: USN (Unknown Target HWMP Sequence Number) subfield: The USN subfield indicates whether the Target HWMP Sequence Number field of the corresponding target is interpreted as HWMP sequence number (USN = 0) or not (USN = 1), the latter meaning that a target HWMP sequence number is unknown at the originator mesh STA.

— Bit 3–7: Reserved.

The Target Address field is represented as a 48-bit MAC address.

The Target HWMP Sequence Number field is coded as an unsigned integer and is the latest known HWMP sequence number received in the past by the originator mesh STA for any path towards the target. If such a target HWMP sequence number is not known, the USN subfield is set to 1 and Target HWMP Sequence Number field is reserved.

Detailed usage of the PREQ element is described in 11C.9.9.

7.3.2.114 PREP element

The Path Reply (PREP) element is used to establish a forward path to a target and to confirm that a target is reachable. The PREP is issued in response to a PREQ. The PREP element is transmitted in an HWMP Mesh Path Selection frame (see 7.4.15.3). The format of the PREP element is shown in Figure 7-95o159.

The Element ID is set to the value given in Table 7-26 for this element.

The length is set to 31 or 37 octets.

The format of the Flags field is shown in Figure 7-95o160.

The Flags field is set as follows:

— Bit 0–5: Reserved.

— Bit 6: AE (Address Extension) subfield (1 = external address present, 0 = otherwise). An AE subfield equal to 1 indicates that the field Target External Address is present, and that the target mesh STA is a proxy for this external address.

— Bit 7: Reserved.

Element ID	Length	Flags	Hop Count	Element TTL	Target Mesh STA Address	Target HWMP Sequence Number	Target External Address	Lifetime
Octets: 1	1	1	1	1	6	4	0 or 6	4

Metric	Originator Mesh STA Address	Originator HWMP Sequence Number
4	6	4

Figure 7-95o159—PREP element format

B0 B5	B6	B7
Reserved	AE	Reserved
Bits:6	1	1

Figure 7-95o160—Flags field format

The Hop Count field is coded as an unsigned integer and is set to the number of hops from the path target to the mesh STA transmitting this element.

The Element TTL field is coded as an unsigned integer and indicates the remaining number of hops allowed for this element.

The Target Mesh STA Address is the MAC address of the target mesh STA or target proxy mesh gate and is represented as a 48-bit MAC address.

The Target HWMP Sequence Number field is coded as an unsigned integer and is set to the HWMP sequence number of the target mesh STA (if the AE subfield in the Flags field is set to 0) or target proxy mesh gate (if the AE subfield in the Flags field is set to 1).

The Target External Address field is set to the external address on behalf of which the PREP is sent. This field is present only if Bit 6 (AE subfield) in Flags field equals 1 and is represented as a 48-bit MAC address.

The Lifetime field is coded as an unsigned integer and is set to the time for which mesh STAs receiving the PREP consider the forwarding information to be valid. The lifetime is measured in TUs.

The Metric field indicates the cumulative metric from the path target to the mesh STA transmitting this element.

The Originator Mesh STA Address field is represented as a 48-bit MAC address and is set to the MAC address of the originator, which is contained in the PREQ.

The Originator HWMP Sequence Number field is coded as an unsigned integer and is set to the HWMP sequence number of the originator mesh STA contained in the PREQ.

The detailed usage of the PREP element is described in 11C.9.10.

7.3.2.115 PERR element

The Path Error (PERR) element is used for announcing an unreachable destination. The PERR element is transmitted in an HWMP Mesh Path Selection frame (see 7.4.15.3). The format of the PERR element is shown in Figure 7-95o161.

	Element ID	Length	Element TTL	Number of Destinations N	Flags #1	Destination Address #1	HWMP Sequence Number #1	Destination External Address #1	Reason Code #1	...
Octets:	1	1	1	1	1	6	4	0 or 6	2	

Figure 7-95o161—PERR element format

The Element ID is set to the value given in Table 7-26 for this element.

The length is variable and set to (2 + 13 × Number of Destinations) or to (2 + 19 × Number of Destinations) octets.

The Element TTL field is coded as an unsigned integer and indicates the remaining number of hops allowed for this element.

The Number of Destinations N field is coded as an unsigned integer and indicates the number of announced destinations in PERR. The maximum value of N is 19. The Flags field, the Destination Address field, the HWMP Sequence Number field, the Destination External Address field, and the Reason Code field are repeated N times in the element.

The format of the Flags field is shown in Figure 7-95o162.

	B0 B5	B6	B7
	Reserved	AE	Reserved
Bits:	4	1	1

Figure 7-95o162—Flags field format

- Bit 0–5: Reserved.
- Bit 6: AE (Address Extension) subfield (1 = destination external address is present, 0 = otherwise).
- Bit 7: Reserved.

The Destination Address field is represented as a 48-bit MAC address and indicates the detected unreachable destination MAC address.

The HWMP Sequence Number field is coded as an unsigned integer and indicates the HWMP sequence number for the invalidated destination, if applicable. Otherwise, the HWMP Sequence Number field is reserved depending on the reason code.

The Destination External Address field is set to the external address, on behalf of which the PERR is sent. This field is present only if Bit 6 (AE subfield) in the Flags field equals 1 and is represented as a 48-bit MAC address.

The Reason Code field specifies the reason for sending a PERR element. The Reason Code is defined in 7.3.1.7.

The detailed usage of the PERR element is described in 11C.9.11.

7.3.2.116 PXU element

The Proxy Update (PXU) element is used to inform the destination mesh STA of the proxy information at the originator mesh STA. The PXU element is transmitted in a Proxy Update frame (see 7.4.16.2). The format of the PXU element is shown in Figure 7-95o163.

	Element ID	Length	PXU ID	PXU Originator MAC Address	Number of Proxy Information (N)	Proxy Information #1	...	Proxy Information #N
Octets:	1	1	1	6	1	11, 15, 17, or 21		11, 15, 17, or 21

Figure 7-95o163—Proxy Update element format

The Element ID is set to the value given in Table 7-26 for this element.

The Length is set to 8 + length of N Proxy Information fields.

The PXU ID field is coded as an unsigned integer and is set to the sequence number of the PXU. The source mesh STA sets the PXU ID field in the PXU element to a value from a single modulo-256 counter that is incremented by 1 for each new PXU element.

The PXU Originator MAC Address field is represented as a 48-bit MAC address and is the MAC address of the mesh STA that originates this Proxy Update element.

The Number of Proxy Information fields is coded as an unsigned integer and is set to the number N of Proxy Information field that follow this field and that are reported to the destination mesh STA. The maximum value of N is 22.

The Proxy Information field contains a single proxy information (see 11C.10.4.2). The length of the Proxy Information field depends on the settings of the subfields in the Flags subfield and is 11, 15, 17, or 21 octets.

The format of the Proxy Information field is defined in Figure 7-95o164.

The format of the Flags subfield is shown in Figure 7-95o165.

The Flags subfield is set as follows:

	Flags	External MAC Address	Proxy Information Sequence Number	Proxy MAC Address	Proxy Information Lifetime
Octets:	1	6	4	0 or 6	0 or 4

Figure 7-95o164—Proxy Information field

	B0	B1	B2	B3 B7
	Delete	Originator Is Proxy	Lifetime	Reserved
Bits:	1	1	1	5

Figure 7-95o165—Flags subfield

- Bit 0: The Delete subfield indicates whether this proxy information is to be deleted. It is set to 1 if the proxy information is to be deleted, and set to 0 otherwise.
- Bit 1: The Originator Is Proxy subfield indicates that the originator mesh STA of the PXU element is the proxy mesh gate of this proxy information when set to 1. In this case, there is no Proxy MAC Address subfield present in this Proxy Information field. When the Originator Is Proxy subfield is 0, the Proxy MAC Address subfield is present in this Proxy Information field.
- Bit 2: The Lifetime subfield indicates that the Proxy Information Lifetime subfield is present in this Proxy Information field when set to 1.
- Bit 3–7: Reserved.

The External MAC Address subfield is represented as a 48-bit MAC address and is the MAC address of the external STA proxied by the proxy mesh gate.

The Proxy Info Sequence Number field is coded as an unsigned integer and is set to the sequence number of the proxy information. The sequence number of the proxy information defines a chronological order of the proxy information for the external STA at this proxy mesh gate.

The Proxy MAC Address subfield is represented as a 48-bit MAC address and is set to the MAC address of proxy mesh gate. It is only present if the Originator Is Proxy subfield of the Flags subfield is 0.

The Proxy Information Lifetime subfield is coded as an unsigned integer and is set to the time for which the mesh STA receiving this PXU considers this proxy information to be valid. The proxy information lifetime is measured in TUs. It is only present if the Lifetime subfield of the Flags subfield is 1.

7.3.2.117 PXUC element

The Proxy Update Confirmation (PXUC) element is used to confirm the previously received PXU. The PXUC element is transmitted in a Proxy Update Confirmation frame (see 7.4.16.3). The format of PXUC element is shown in Figure 7-95o166.

The Element ID is set to the value given in Table 7-26 for this element.

The Length field is set to 7.

Element ID	Length	PXU ID	PXU Recipient MAC Address
Octets: 1	1	1	6

Figure 7-95o166—Proxy Update Confirmation element format

The PXU ID field is coded as an unsigned integer and is the PXU ID of the received PXU that is being confirmed.

The PXU Recipient MAC Address is represented as a 48-bit MAC address and is set to the MAC address of the recipient of the PXU, i.e., the originator of the PXUC element.

7.3.2.118 Authenticated Mesh Peering Exchange element

The Authenticated Mesh Peering Exchange element includes information needed to perform the authentication sequence during an authenticated mesh peering exchange. This element is shown in Figure 7-95o167.

Element ID	Length	Selected Pairwise Cipher Suite	Local Nonce	Peer Nonce	Key Replay Counter	GTKdata
Octets: 1	1	4	32	32	8	variable

Figure 7-95o167—Authenticated Mesh Peering Exchange element format

The Element ID is set to the value given in Table 7-26 for this element.

The Length field is variable and indicates the number of octets in the information field (fields following the Element ID and Length fields).

The Selected Pairwise Cipher Suite field contains a pairwise cipher suite selector, as defined in 7.3.2.25.1, indicating a cipher suite to be used to secure the link.

The Local Nonce field contains a nonce value chosen by the mesh STA that is sending the element. It is encoded following the conventions from 7.1.1.

The Peer Nonce field contains a nonce value that was chosen by the peer mesh STA or candidate peer mesh STA to which the element is being sent. It is encoded following the conventions from 7.1.1.

The Key Replay Counter field is optional. It is only used for the Mesh Group Key Inform frame (see 11C.6.3) and the Mesh Group Key Acknowledge frame (see 11C.6.4). It is represented as an unsigned binary number.

The GTKdata field is optional. When present, it contains the bit string of {GTK || Key RSC || GTKExpirationTime} as the GTK data material. When present, the GTKdata field is protected by the exchange in which it is contained (see 11C.5). The Key RSC denotes the last frame sequence number sent using the GTK and is specified in Table 8-3 of 8.5.2. GTKExpirationTime denotes the key lifetime of the GTK in seconds and the format is specified in Figure 8-31 of 8.5.2.

Detailed usage of the Authenticated Mesh Peering Exchange element is described in 11C.5.5 and in 11C.6.

7.3.2.119 MIC element

The MIC element (MICE) provides message integrity to Mesh Peering Management frames. The format of the MIC element is shown in Figure 7-95o168.

	Element ID	Length	MIC
Octets:	1	1	16

Figure 7-95o168—MIC element format

The Element ID is set to the value given in Table 7-26 for this element.

The Length field is set to 16.

The MIC field contains a message integrity code calculated over the Mesh Peering Management frame (as specified in 11C.5) and the mesh group key handshake frame (as specified in 11C.6).

7.4 Action frame format details

7.4.1 Spectrum management action details

7.4.1.5 Channel Switch Announcement frame format

Change the first paragraph of 7.4.1.5 as follows:

The Channel Switch Announcement frame uses the Action frame body format and is transmitted by an AP in a BSS, ~~or~~ a STA in an IBSS, or a mesh STA in an MBSS to advertise a channel switch. The format of the Channel Switch Announcement Action field is shown in Figure 7-100.

Replace Figure 7-100 with the following figure:

	Category	Spectrum Management Action	Channel Switch Announcement element	Secondary Channel Offset element	Mesh Channel Switch Parameters element
Octets:	1	1	5	3	6

Figure 7-100—Channel Switch Announcement frame Action field format

Insert the following paragraph to the end of 7.4.1.5:

The Mesh Channel Switch Parameters element is defined in 7.3.2.103. This element is present when a mesh STA performs MBSS channel switch. The Mesh Channel Switch Parameters element is not included for channel switch other than MBSS.

7.4.7 Public Action details

7.4.7.6 Extended Channel Switch Announcement frame format

Change the first paragraph of 7.4.7.6 as follows:

The Extended Channel Switch Announcement frame is transmitted by an AP in an infrastructure BSS, ~~or~~ a STA in an IBSS, or a mesh STA in an MBSS to advertise a channel switch. The format of the Extended Channel Switch Announcement frame Action field is shown in Figure 7-101h4.

Replace Figure 7-101h4 with the following figure:

	Category	Public Action	Channel Switch Mode	New Regulatory Class	New Channel Number	Channel Switch Count	Mesh Channel Switch Parameters element
Octets:	1	1	1	1	1	1	6

Figure 7-101h4—Extended Channel Switch Announcement frame Action field format

Change the last paragraph of 7.4.7.6 as follows:

The Channel Switch Mode, New Regulatory Class, New Channel Number, and Channel Switch Count ~~remaining~~ fields are as described in the Extended Channel Switch Announcement element (see 7.3.2.53).

Mesh Channel Switch Parameters element is defined in 7.3.2.103. This element is present when a mesh STA performs MBSS channel switch. The Mesh Channel Switch Parameters element is not used for channel switch other than the MBSS channel switch.

Insert the following new subclauses after 7.4.13:

7.4.14 Self-protected Action frame details

7.4.14.1 Self-protected Action fields

The Self-protected Action frame is defined to allow robust STA-STA communications of the Action frames that are not robust (see 7.3.1.11). The protocols that use these Action frames are responsible for deciding whether to protect these frames and supporting protection mechanisms for these frames as needed.

Self-protected Action frames have a different nature than Public Action frames and Robust Action frames. Robust Action frames assume the existence of a completely established security association. Self-protected Action frames typically exist to manage the creation and destruction of security associations, whether or not they are completely established.

Public Action frames are defined as public for all STAs, including those that are not in the BSS and MBSS. Self-protected Action frames, however, are used for relationship creation and maintenance between two specific STAs. Their public nature is incidental.

A Self-protected Action field, in the octet field immediately after the Category field, differentiates the formats. The defined Self-protected Action frames are listed in Table 7-57v24.

Table 7-57v24—Self-protected Action field values

Self-protected Action field value	Description
0	Reserved
1	Mesh Peering Open
2	Mesh Peering Confirm
3	Mesh Peering Close
4	Mesh Group Key Inform
5	Mesh Group Key Acknowledge
6–255	Reserved

The Mesh Peering Open frame, the Mesh Peering Confirm frame, and the Mesh Peering Close frame are referred to as “Mesh Peering Management frames”.

7.4.14.2 Mesh Peering Open frame format

7.4.14.2.1 Mesh Peering Open frame self protection

Protection of this frame is provided when authenticated mesh peering exchange (AMPE) is enabled. AMPE provides integrity protection of Mesh Peering Open frames.

When the Mesh Peering Open frame is used by the mesh peering management (MPM) protocol, integrity protection on the frame is not enabled.

7.4.14.2.2 Mesh Peering Open frame details

The Mesh Peering Open frame is used to open a mesh peering using the procedures defined in 11C.3.6 and in 11C.5.5. The Mesh Peering Open frame is also, together with Mesh Peering Confirm and Mesh Peering Close frames, referred to as a Mesh Peering Management frame. The format of the Mesh Peering Open frame Action field is shown in Table 7-57v25.

Table 7-57v25—Mesh Peering Open frame Action field format

Order	Information	Notes
1	Category	
2	Self-protected Action	
3	Capability	
4	Supported Rates	
5	Extended Supported Rates	The Extended Supported Rates element is present if there are more than eight supported rates, and is optionally present otherwise.

Table 7-57v25—Mesh Peering Open frame Action field format *(continued)*

Order	Information	Notes
6	Power Capability	The Power Capability element is present if dot11SpectrumManagementRequired is true.
7	Supported Channels	The Supported Channels element is present if dot11SpectrumManagementRequired is true and dot11ExtendedChannelSwitchEnabled is false.
8	RSN	The RSN element is present only if dot11MeshSecurityActivated is true.
9	Mesh ID	The Mesh ID element is set as described in 7.3.2.99.
10	Mesh Configuration	The Mesh Configuration element is set as described in 7.3.2.98.
11	Mesh Peering Management	The Mesh Peering Management element is set as described in 7.3.2.102.
12	ERP Information	The ERP Information element is present if ERP mesh STA detects NonERP STAs in its vicinity, and is optionally present otherwise.
13	Supported Regulatory Classes	The Supported Regulatory Classes element is present if dot11ExtendedChannelSwitchEnabled is true.
14	HT Capabilities	The HT Capabilities element is present when dot11HighThroughputOptionImplemented is true.
15	HT Operation	The HT Operation element is included when dot11HighThroughputOptionImplemented is true.
16	20/40 BSS Coexistence element	The 20/40 BSS Coexistence element is optionally present when the dot112040BSSCoexistenceManagementSupport is true.
17	Extended Capabilities element	The Extended Capabilities element is optionally present if any of the fields in this element are nonzero.
18	Interworking	The Interworking element is present if dot11InterworkingServiceEnabled is true.
Last – 2	Vendor Specific	One or more vendor-specific elements are optionally present. These elements follow all other elements except MIC element and Authenticated Mesh Peering Exchange element.
Last – 1	MIC element	MIC element is present when dot11MeshSecurityActivated is true and a PMK exists between the sender and recipient of this frame.
Last	Authenticated Mesh Peering Exchange	The Authenticated Mesh Peering Exchange element is present when dot11MeshSecurityActivated is true and a PMK exists between the sender and recipient of this frame.

The Category field is set to the value in Table 7-24 for category Self-protected.

The Self-protected Action field is set to the value in Table 7-57v24 representing Mesh Peering Open.

The MIC element (MICE) appears prior to the Authenticated Mesh Peering Exchange element in the Mesh Peering Open frame. The information following the MIC element through to the end of the Mesh Peering Open frame body is encrypted and authenticated (see 11C.5).

7.4.14.3 Mesh Peering Confirm frame format

7.4.14.3.1 Mesh Peering Confirm frame self protection

Protection of this frame is provided when authenticated mesh peering exchange (AMPE) is enabled. AMPE provides integrity protection of Mesh Peering Confirm frames.

When the Mesh Peering Confirm frame is used by the mesh peering management (MPM) protocol, integrity protection on the frame is not enabled.

7.4.14.3.2 Mesh Peering Confirm frame details

The Mesh Peering Confirm frame is used to confirm a mesh peering using the procedures defined in 11C.3.7 and 11C.5.5. The Mesh Peering Confirm frame is also, together with Mesh Peering Open and Mesh Peering Close frames, referred to as a Mesh Peering Management frame. The format of the Mesh Peering Confirm frame Action field is shown in Table 7-57v26.

Table 7-57v26—Mesh Peering Confirm frame Action field format

Order	Information	Notes
1	Category	
2	Self-protected Action	
3	Capability	
4	AID	
5	Supported Rates	
6	Extended Supported Rates	The Extended Supported Rates element is present if there are more than eight supported rates, and is optionally present otherwise.
7	RSN	The RSN element is present only when dot11MeshSecurityActivated is true.
8	Mesh ID	The Mesh ID element is set as described in 7.3.2.99.
9	Mesh Configuration	The Mesh Configuration element is set as described in 7.3.2.98.
10	Mesh Peering Management	The Mesh Peering Management element is set as described in 7.3.2.102.
11	HT Capabilities	The HT Capabilities element is present when dot11HighThroughputOptionImplemented is true.
12	HT Operation	The HT Operation element is included when dot11HighThroughputOptionImplemented is true.
13	20/40 BSS Coexistence element	The 20/40 BSS Coexistence element is optionally present when the dot112040BSSCoexistenceManagementSupport is true.
14	Extended Capabilities element	The Extended Capabilities element is optionally present if any of the fields in this element are nonzero.
Last – 2	Vendor Specific	One or more vendor-specific elements are optionally present. These elements follow all other elements except MIC element and Authenticated Mesh Peering Exchange element.
Last – 1	MIC element	MIC element is present when dot11MeshSecurityActivated is true and a PMK exists between the sender and recipient of this frame.

Table 7-57v26—Mesh Peering Confirm frame Action field format *(continued)*

Order	Information	Notes
Last	Authenticated Mesh Peering Exchange	The Authenticated Mesh Peering Exchange element is present when dot11MeshSecurityActivated is true and a PMK exists between the sender and recipient of this frame.

The Category field is set to the value in Table 7-24 for category Self-protected.

The Self-protected Action field is set to the value in Table 7-57v24 representing Mesh Peering Confirm.

The MIC element (MICE) appears prior to the Authenticated Mesh Peering Exchange element in the Mesh Peering Open frame. The information following the MIC element through to the end of the Mesh Peering Confirm frame body is encrypted and authenticated (see 11C.5).

7.4.14.4 Mesh Peering Close frame format

7.4.14.4.1 Mesh Peering Close frame self protection

Protection of this frame is provided when authenticated mesh peering exchange (AMPE) is enabled. AMPE provides integrity protection of Mesh Peering Close frames.

When the Mesh Peering Close frame is used by the mesh peering management (MPM) protocol, integrity protection on the frame is not enabled.

7.4.14.4.2 Mesh Peering Close frame details

The Mesh Peering Close frame is used to close a mesh peering using the procedures defined in 11C.3.8 and in 11C.5.5. The Mesh Peering Close frame is also, together with Mesh Peering Open and Mesh Peering Confirm frames, referred to as a Mesh Peering Management frame. The format of the Mesh Peering Close frame Action field is shown in Table 7-57v27.

Table 7-57v27—Mesh Peering Close frame Action field format

Order	Information	Notes
1	Category	
2	Self-protected Action	
3	Mesh ID	The Mesh ID element is set as described in 7.3.2.99.
4	Mesh Peering Management	The Mesh Peering Management element is set as described in 7.3.2.102.
Last – 2	Vendor Specific	One or more vendor-specific elements are optionally present. These elements follow all other elements except MIC element and Authenticated Mesh Peering Exchange element.
Last – 1	MIC element	MIC element is present when dot11MeshSecurityActivated is true and a PMK exists between the sender and recipient of this frame.
Last	Authenticated Mesh Peering Exchange	The Authenticated Mesh Peering Exchange element is present when dot11MeshSecurityActivated is true and a PMK exists between the sender and recipient of this frame.

The Category field is set to the value in Table 7-24 for category Self-protected.

The Self-protected Action field is set to the value in Table 7-57v24 representing Mesh Peering Close.

The MIC element (MICE) appears prior to the Authenticated Mesh Peering Exchange element in the Mesh Peering Open frame. The information following the MIC element through to the end of the Mesh Peering Close frame body is encrypted and authenticated (see 11C.5).

7.4.14.5 Mesh Group Key Inform frame format

7.4.14.5.1 Mesh Group Key Inform frame self protection

The protection of the frames is provided by the mesh group key handshake protocol (see 11C.6) that uses Mesh Group Key Inform frames.

7.4.14.5.2 Mesh Group Key Inform frame details

The Mesh Group Key Inform frame is used to update a mesh GTK (MGTK) with a peer. The format of the Mesh Group Key Inform frame Action field is shown in Table 7-57v28.

Table 7-57v28—Mesh Group Key Inform frame Action field format

Order	Information	Notes
1	Category	
2	Self-protected Action	
3	MIC element	
4	Authenticated Mesh Peering Exchange	

The Category field is set to the value in Table 7-24 for category Self-protected.

The Self-protected Action field is set to the value in Table 7-57v24 representing Mesh Group Key Inform.

The MIC element is set as defined in 7.3.2.119.

The Authenticated Mesh Peering Exchange element is set according to 11C.6. The information following the MIC element through to the end of the Mesh Group Key Inform frame body is encrypted and authenticated (see 11C.6.2).

7.4.14.6 Mesh Group Key Acknowledge frame format

7.4.14.6.1 Mesh Group Key Acknowledge frame self protection

The protection of the frames is provided by the mesh group key handshake protocol (see 11C.6) that uses Mesh Group Key Acknowledge frames.

7.4.14.6.2 Mesh Group Key Acknowledge frame details

The Mesh Group Key Acknowledge frame is used to acknowledge receipt and processing of a Mesh Group Key Inform frame. The format of the Mesh Group Key Acknowledge frame Action field is shown in Table 7-57v29.

Table 7-57v29—Mesh Group Key Acknowledge frame Action field format

Order	Information	Notes
1	Category	
2	Self-protected Action	
3	MIC element	
4	Authenticated Mesh Peering Exchange	

The Category field is set to the value in Table 7-24 for category Self-protected.

The Self-protected Action field is set to the value in Table 7-57v24 representing Mesh Group Key Acknowledge.

The MIC element is set as defined in 7.3.2.119.

The Authenticated Mesh Peering Exchange element is set according to 11C.6. The information following the MIC element through to the end of the Mesh Group Key Acknowledge frame body is encrypted and authenticated (see 11C.6.2).

7.4.15 Mesh Action frame details

7.4.15.1 Mesh Action fields

Several Mesh Action frame formats are defined for mesh BSS operation. A Mesh Action field, in the octet field immediately after the Category field, differentiates the formats. The Mesh Action field values associated with each frame format are defined in Table 7-57v30.

Table 7-57v30—Mesh Action field values

Mesh Action field value	Description
0	Mesh Link Metric Report
1	HWMP Mesh Path Selection
2	Gate Announcement
3	Congestion Control Notification
4	MCCA Setup Request
5	MCCA Setup Reply
6	MCCA Advertisement Request

Table 7-57v30—Mesh Action field values *(continued)*

Mesh Action field value	Description
7	MCCA Advertisement
8	MCCA Teardown
9	TBTT Adjustment Request
10	TBTT Adjustment Response
11–255	Reserved

7.4.15.2 Mesh Link Metric Report frame format

The Mesh Link Metric Report frame is transmitted by a mesh STA to a neighbor peer mesh STA to report metric information on the link between the two mesh STAs. It is also transmitted by a mesh STA to a neighbor peer mesh STA to request metric information on the link between the two mesh STAs from the recipient. This frame is transmitted using an individually addressed frame. The format of the Mesh Link Metric Report frame Action field is shown in Table 7-57v31.

Table 7-57v31—Mesh Link Metric Report frame Action field format

Order	Information	Notes
1	Category	
2	Mesh Action	
3	Mesh Link Metric Report element	

The Category field is set to the value in Table 7-24 for category Mesh Action.

The Mesh Action field is set to the value in Table 7-57v30 representing Mesh Link Metric Report.

The Mesh Link Metric Report element is set as described in 7.3.2.100.

7.4.15.3 HWMP Mesh Path Selection frame format

The HWMP Mesh Path Selection frame is transmitted by a mesh STA to establish, update or delete paths to other mesh STAs using the HWMP defined in 11C.9. This frame is transmitted in an individually or group addressed frame depending on the contained elements and as defined in 11C.9.7. The format of the HWMP Mesh Path Selection frame Action field is shown in Table 7-57v32.

HWMP Mesh Path Selection frame contains one or more of the elements indicated in Table 7-57v32.

The Category field is set to the value in Table 7-24 for category Mesh Action.

The Mesh Action field is set to the value in Table 7-57v30 representing HWMP Mesh Path Selection.

The Path Request element is set as described in 7.3.2.113.

Table 7-57v32—HWMP Mesh Path Selection frame Action field format

Order	Information	Notes
1	Category	
2	Mesh Action	
3	Path Request element (optional)	
4	Path Reply element (optional)	
5	Path Error element (optional)	
6	Root Announcement element (optional)	

The Path Reply element is set as described in 7.3.2.114.

The Path Error element is set as described in 7.3.2.115.

The Root Announcement element is set as described in 7.3.2.112.

7.4.15.4 Gate Announcement frame format

The Gate Announcement frame is transmitted by a mesh gate to announce its presence in the MBSS. This frame is transmitted using group addresses. The format of the Gate Announcement frame Action field is shown in Table 7-57v33.

Table 7-57v33—Gate Announcement frame Action field format

Order	Information	Notes
1	Category	
2	Mesh Action	
3	Gate Announcement element	

The Category field is set to the value in Table 7-24 for category Mesh Action.

The Mesh Action field is set to the value in Table 7-57v30 representing Gate Announcement.

The Gate Announcement element is set as described in 7.3.2.111.

7.4.15.5 Congestion Control Notification frame format

A mesh STA uses the Congestion Control Notification frame to indicate its congestion status to its neighbor peer mesh STA(s). This frame is transmitted using individual addresses or group addresses. The format of the Congestion Control Notification frame Action field is shown in Table 7-57v34.

The Category field is set to the value in Table 7-24 for category Mesh Action.

The Mesh Action field is set to the value in Table 7-57v30 representing Congestion Control Notification.

Table 7-57v34—Congestion Control Notification frame Action field format

Order	Information	Notes
1	Category	
2	Mesh Action	
3 to (N+2)	Congestion Notification element	One or more Congestion Notification elements (7.3.2.101). Repeated N times (N is a number of Congestion Notification element contained in the frame).

The Congestion Notification element is set as described in 7.3.2.101.

7.4.15.6 MCCA Setup Request frame format

The MCCA Setup Request frame is used to set up an MCCAOP reservation. It is transmitted by a mesh STA with dot11MCCAActivated equal to true to one or more neighbor peer mesh STA with dot11MCCAActivated equal to true. This frame is transmitted using individual addresses or group addresses. The format of the MCCA Setup Request frame Action field is shown in Table 7-57v35.

Table 7-57v35—MCCA Setup Request frame Action field format

Order	Information	Notes
1	Category	
2	Mesh Action	
3	MCCAOP Setup Request element	

The Category field is set to the value in Table 7-24 for category Mesh Action.

The Mesh Action field is set to the value in Table 7-57v30 representing MCCA Setup Request.

The MCCAOP Setup Request element is described in 7.3.2.106.

7.4.15.7 MCCA Setup Reply frame format

The MCCA Setup Reply frame is used to reply to an MCCA Setup Request frame. It is transmitted by a mesh STA with dot11MCCAActivated equal to true to a neighbor peer mesh STA with dot11MCCAActivated equal to true. This frame is transmitted using individual addresses. The format of the MCCA Setup Reply frame Action field is shown in Table 7-57v36.

The Category field is set to the value in Table 7-24 for category Mesh Action.

The Mesh Action field is set to the value in Table 7-57v30 representing MCCA Setup Reply.

The MCCAOP Setup Reply element is described in 7.3.2.107.

Table 7-57v36—MCCA Setup Reply frame Action field format

Order	Information	Notes
1	Category	
2	Mesh Action	
3	MCCAOP Setup Reply element	

7.4.15.8 MCCA Advertisement Request frame format

The MCCA Advertisement Request frame is transmitted by a mesh STA with dot11MCCAActivated equal to true to a neighbor peer mesh STA with dot11MCCAActivated equal to true in order to request MCCAOP advertisements from the neighbor peer mesh STA. This frame is transmitted using individual addresses. The format of the MCCA Advertisement Request frame Action field is shown in Table 7-57v37.

Table 7-57v37—MCCA Advertisement Request frame Action field format

Order	Information	Notes
1	Category	
2	Mesh Action	
3	MCCAOP Advertisement Overview element	An MCCAOP Advertisement Overview element is optionally present.

The Category field is set to the value in Table 7-24 for category Mesh Action.

The Mesh Action field is set to the value in Table 7-57v30 representing MCCA Advertisement Request.

The MCCAOP Advertisement Overview element is described in 7.3.2.108.

7.4.15.9 MCCA Advertisement frame format

The MCCA Advertisement frame is transmitted by a mesh STA with dot11MCCAActivated equal to true to one or more neighbor peer mesh STAs with dot11MCCAActivated equal to true. This frame is transmitted using group addresses or individual addresses. The format of the MCCA Advertisement frame Action field is shown in Table 7-57v38.

The Category field is set to the value in Table 7-24 for category Mesh Action.

The Mesh Action field is set to the value in Table 7-57v30 representing MCCA Advertisement.

The MCCAOP Advertisement Overview element is described in 7.3.2.108.

The MCCAOP Advertisement element is described in 7.3.2.109.

Table 7-57v38—MCCA Advertisement frame Action field format

Order	Information	Notes
1	Category	
2	Mesh Action	
3	MCCAOP Advertisement Overview element	An MCCAOP Advertisement Overview element is optionally present.
4	MCCAOP Advertisement elements	In case the MCCAOP Advertisement Overview element is present, zero or more MCCAOP Advertisement elements are present. In case the MCCAOP Advertisement Overview element is not present, one or more MCCAOP Advertisement elements are present.

7.4.15.10 MCCA Teardown frame format

The MCCA Teardown frame is transmitted by a mesh STA with dot11MCCAActivated equal to true to one or more neighbor peer mesh STAs with dot11MCCAActivated equal to true. This frame is transmitted using group addresses or individual addresses. The format of the MCCA Teardown frame Action field is shown in Table 7-57v39.

Table 7-57v39—MCCA Teardown frame Action field format

Order	Information	Notes
1	Category	
2	Mesh Action	
3	MCCAOP Teardown element	

The Category field is set to the value in Table 7-24 for category Mesh Action.

The Mesh Action field is set to the value in Table 7-57v30 representing MCCA Teardown.

The MCCAOP Teardown element is described in 7.3.2.110.

7.4.15.11 TBTT Adjustment Request frame format

The TBTT Adjustment Request frame is used to request a particular neighbor peer mesh STA to adjust its TBTT. This frame is transmitted using individual addresses. The format of the TBTT Adjustment Request frame Action field is shown in Table 7-57v40.

The Category field is set to the value in Table 7-24 for category Mesh Action.

The Mesh Action field is set to the value in Table 7-57v30 representing TBTT Adjustment Request.

The Beacon Timing element is set as described in 7.3.2.105. When not all beacon timing information is included in a Beacon Timing element due to the maximum information element size limit, multiple Beacon Timing elements are present. The elements are present in the order of Beacon Timing Element Number field value in the Report Control field of the Beacon Timing element.

Table 7-57v40—TBTT Adjustment Request frame Action field format

Order	Information	Notes
1	Category	
2	Mesh Action	
3 to (N_Info +2)	Beacon Timing element	Repeated N_Info times (N_Info is a number of beacon timing information tuples as described in 11C.12.4.2.5).

7.4.15.12 TBTT Adjustment Response frame format

The TBTT Adjustment Response frame is used to respond to a TBTT adjustment request. This frame is transmitted using individual addresses. The format of the TBTT Adjustment Response frame Action field is shown in Table 7-57v41.

Table 7-57v41—TBTT Adjustment Response frame Action field format

Order	Information	Notes
1	Category	
2	Mesh Action	
3	Status Code	
4 to (N_Info +3)	Beacon Timing element (optional)	Repeated N_Info times (N_Info is a number of beacon timing information tuples as described in 11C.12.4.2.5).

The Category field is set to the value in Table 7-24 for category Mesh Action.

The Mesh Action field is set to the value in Table 7-57v30 representing TBTT Adjustment Response.

The Status Code field is set as described in 11C.12.4.4.2.

The Beacon Timing element is set as defined in 7.3.2.105. It is present only if the Status Code is set to 78 (i.e., when the request is not successful due to the neighbor constraint). When not all beacon timing information is included in a Beacon Timing element due to the maximum information element size limit, multiple Beacon Timing elements are present. The elements are present in the order of Beacon Timing Element Number field value in the Report Control field of the Beacon Timing element.

7.4.16 Multihop Action frame details

7.4.16.1 Multihop Action fields

Several Multihop Action frame formats are defined for mesh BSS operation. A Multihop Action field, in the octet field immediately after the Category field, differentiates the formats. The Multihop Action field values associated with each frame format are defined in Table 7-57v42. The Mesh Control field is present immediately after the Multihop Action field in all Multihop Action frames.

Table 7-57v42—Multihop Action field values

Multihop Action field value	Description
0	Proxy Update
1	Proxy Update Confirmation
2–255	Reserved

7.4.16.2 Proxy Update frame format

The Proxy Update frame is used to inform the recipient about new, updated, or deleted proxy information. This frame is transmitted using individual addresses. The format of the Proxy Update frame Action field is shown in Table 7-57v43.

Table 7-57v43—Proxy Update frame Action field format

Order	Information	Notes
1	Category	
2	Multihop Action	
3	Mesh Control	
4 to (N+3)	Proxy Update element	Repeated N times (N is the number of Proxy Update elements contained in the frame).

The Category field is one octet and is set to the value in Table 7-24 for category Multihop Action.

The Multihop Action field is set to the value in Table 7-57v42 representing Proxy Update.

The Mesh Control field is set as defined in 7.1.3.6.3.

The Proxy Update element is described in 7.3.2.116. The Proxy Update frame allows the inclusion of multiple Proxy Update elements.

7.4.16.3 Proxy Update Confirmation frame format

The Proxy Update Confirmation frame is transmitted by a mesh STA in response to a Proxy Update frame. This frame is used to inform that the corresponding Proxy Update element has been properly received, and is transmitted using individual addresses. The format of the Proxy Update Confirmation frame Action field is shown in Table 7-57v44.

The Category field is one octet and is set to the value in Table 7-24 for category Multihop Action.

The Multihop Action field is set to the value in Table 7-57v42 representing Proxy Update Confirmation.

Table 7-57v44—Proxy Update Confirmation frame Action field format

Order	Information	Notes
1	Category	
2	Multihop Action	
3	Mesh Control	
4 to (N+3)	Proxy Update Confirmation element	Repeated N times (N is the number of Proxy Update Confirmation elements contained in the frame).

The Mesh Control field is set as defined in 7.1.3.6.3.

The Proxy Update Confirmation element is described in 7.3.2.117. The Proxy Update Confirmation frame allows the inclusion of multiple Proxy Update Confirmation elements.

8. Security

8.1 Framework

8.1.1 Security methods

Change the list item starting with "RSNA establishment" in 8.1.1 as follows:

— RSNA establishment and termination procedures, including use of IEEE 802.1X authentication, described in 8.4 and SAE authentication described in 8.2a.

8.1.3 RSNA Establishment

Change the contents of item b) and c) in 8.1.3 as follows:

b) If an RSNA is based on a PSK or password in an ESS, the SME establishes an RSNA as follows:

1) It identifies the AP as RSNA-capable from the AP's Beacon or Probe Response frames.

2) If the RSNA-capable AP advertises support for SAE authentication in its Beacon or Probe Response frames, and the STA has a group defined in the dot11RSNAConfigDLCGroupTable and a password for the AP in the dot11RSNAConfigPasswordValueTable, the STA shall invoke SAE authentication to establish a PMK. If the RSNA-capable AP does not advertise support for SAE authentication in its Beacon and Probe Response frames but advertises support for the alternate form of PSK authentication (see 5.8.2.2), and the STA also supports the alternate form of PSK authentication, the STA may ~~It shall~~ invoke Open System authentication and use the PSK as the PMK with the key management algorithm in step 4) below.

3) It negotiates cipher suites during the association process, as described in 8.4.2 and 8.4.3.

4) It establishes temporal keys by executing a key management algorithm, using the protocol defined by 8.5. ~~It uses the PSK as the PMK.~~

5) It protects the data link by programming the negotiated cipher suites and the established temporal key into the MAC and then invoking protection.

6) If the STAs negotiate Management Frame Protection, the STA programs the TK and pairwise cipher suite into the MAC for protection of unicast Robust Management frames. It also installs the IGTK and the IPN for protection of group addressed Robust Management frames.

c) If an RSNA is based on a PSK or password in an IBSS, the SME executes the following sequence of procedures:

1) It identifies the peer as RSNA-capable from the peer's Beacon and Probe Response frames.

NOTE—STAs can respond to a data MPDU from an unrecognized STA by sending a Probe Request frame to find out whether the unrecognized STA is RSNA-capable.

2) If the RSNA-capable peer advertises support for SAE authentication in its Beacon and Probe Response frames and the STA has a group defined in the dot11RSNAConfigDLCGroupTable and a password for the peer in the dot11RSNAConfigPasswordValueTable, the STA shall invoke SAE authentication and establish a PMK. If the RSNA-capable peer does not advertise support for SAE authentication but advertises support for the alternate form of PSK authentication (see 5.8.2.2), and the STA also supports the alternate form of PSK authentication the STA ~~It~~ may optionally invoke Open System authentication and use a PSK as the PMK with the alternate form of PSK authentication.

3) Each STA uses the procedures in 8.5, to establish temporal keys and to negotiate cipher suites. ~~It uses a PSK as the PMK.~~ Note that two peers may follow this procedure simultaneously. See 8.4.9.

4) It protects the data link by programming the negotiated cipher suites and the established temporal key and then invoking protection.

8.1.6 Emergency service establishment in an RSN

Change the first paragraph in 8.1.6 as follows:

An AP or a mesh STA that supports RSNAs and has the ESR bit set to 1 and the UESA bit set to 1 in the Interworking element in Beacon and Probe Response frames, supports both RSNAs and emergency services associations (see 11.3.2.1) simultaneously.

NOTE—In an infrastructure BSS, STAs with emergency services association are advised to discard all group addressed frames they receive, as they do not possess the Group Key and will not be able to decrypt group addressed frames. In an RSNA enabled BSS that has ~~having~~ one or more STAs associated with an emergency services association, it is recommended that an AP ~~should~~ avoids transmitting unprotected group addressed frames in order not to ~~not~~ disturb the operation of STAs that are in possession of Group Key. One possible way of achieving this is to support Proxy-ARP in the AP, as defined in 11.22.13. In addition, it is recommended that an AP supporting emergency services association should also support DMS to convert group addressed frames to individually addressed frames and transmit them to STAs associated using the emergency services association. STAs using emergency services association might ~~could~~ request ~~for~~ DMS, if needed.

8.2 Pre-RSNA security methods

Insert the following sentence to the end of the first paragraph in 8.2:

Open System Authentication and Open System Deauthentication shall not be used between mesh STAs.

Insert the following new subclause immediately after 8.2:

8.2a Authentication using a password

8.2a.1 SAE overview

STAs, both AP STAs and non-AP STAs, may authenticate each other by proving possession of a password. Authentication protocols that employ passwords need to be resistant to off-line dictionary attacks.

Simultaneous authentication of equals (SAE) is a variant of *Dragonfly*, a password-authenticated key exchange based on a zero-knowledge proof. SAE is used by STAs to authenticate with a password; it has the following security properties:

- The successful termination of the protocol results in a PMK shared between the two STAs.
- An attacker is unable to determine either the password or the resulting PMK by passively observing an exchange or by interposing itself into the exchange by faithfully relaying messages between the two STAs.
- An attacker is unable to determine either the password or the resulting shared key by modifying, forging, or replaying frames to an honest, uncorrupted STA.
- An attacker is unable to make more than one guess at the password per attack. This implies that the attacker cannot make one attack and then go offline and make repeated guesses at the password until successful. In other words, SAE is resistant to dictionary attack.
- Compromise of a PMK from a previous run of the protocol does not provide any advantage to an adversary attempting to determine the password or the shared key from any other instance.
- Compromise of the password does not provide any advantage to an adversary in attempting to determine the PMK from the previous instance.

Unlike other authentication protocols SAE does not have a notion of an “initiator” and “responder” or of a “supplicant” and “authenticator.” The parties to the exchange are equals, with each side being able to initiate the protocol. Each side may initiate the protocol simultaneously such that each side views itself as the "initiator" for a particular run of the protocol. Such a peer-to-peer protocol may be used in a traditional client-server (or supplicant/authenticator) fashion but the converse does not hold. This requirement is necessary to address the unique nature of MBSSs.

The parties involved will be called STA-A and STA-B. They are identified by their MAC addresses, STA-A-MAC and STA-B-MAC, respectively. STAs begin the protocol when they discover a peer through Beacons and Probe Responses, or when they receive an IEEE 802.11 authentication frame indicating SAE authentication from a peer.

SAE is an RSNA authentication protocol and is selected according to 8.4.2.

SAE shall be implemented on all mesh STAs to facilitate and promote interoperability.

8.2a.2 Assumptions on SAE

SAE uses various functions and data to accomplish its task and assumes certain properties about each function. These are as follows:

- H is an "extractor" function (see IETF RFC 5869) that concentrates potentially dispersed entropy from an input to create an output that is a cryptographically strong, pseudo-random key. This function takes as input a non-secret "salt" and a secret input and produces a fixed-length output.
- CN is a confirmation function that takes a secret key and data to confirm and bind to the exchange.
- A finite cyclic group is negotiated for which solving the discrete logarithm problem is computationally infeasible.

When used with AKMs 00-0F-AC:8 or 00-0F-AC:9 from Table 7-34, H is instantiated as HMAC-SHA256:

$$H(salt, ikm) = \text{HMAC-SHA256}(salt, ikm)$$

When used with AKMs 00-0F-AC:8 or 00-0F-AC:9 from Table 7-34, CN is instantiated as a function that takes a key and a sequence of data. Each piece of data is converted to an octet string and concatenated together before being passed, along with the key, to HMAC-SHA256:

$$CN(key, X, Y, Z, \ldots) = \text{HMAC-SHA256}(key, \text{D2OS}(X) \,\|\, \text{D2OS}(Y) \,\|\, \text{D2OS}(Z) \,\|\, \ldots)$$

where D2OS() represents the data to octet string conversion functions in 8.2a.7.2.

Other instantiations of functions H and CN require creation of a new AKM identifier.

8.2a.3 Representation of a password

Passwords are used in SAE to deterministically compute a secret element in the negotiated group, called a "password element." The input to this process needs to be in the form of a binary string. For the protocol to successfully terminate, it is necessary for each side to produce identical binary strings for a given password, even if that password is in character format. There is no canonical binary representation of a character and ambiguity exists when the password is a character string. To eliminate this ambiguity, a compliant STA shall represent a character-based password as an ASCII string. Representation of a character-based password in another character set or use of a password pre-processing technique (to map a character string to a binary string) may be agreed upon, in an out-of-band fashion, prior to beginning SAE. If the password is already in binary form (e.g., it is a binary pre-shared key) no character set representation is assumed. The binary representation of the password, after being transformed from a character representation or directly if it is already in binary form, is stored in the dot11RSNASAEPasswordValueTable. When a "password" is called for in the description of SAE that follows the credential from the dot11RSNASAEPasswordValueTable is used.

8.2a.4 Finite cyclic groups

8.2a.4.1 General

SAE uses discrete logarithm cryptography to achieve authentication and key agreement. Each party to the exchange derives ephemeral public and private keys with respect to a particular set of domain parameters that define a finite cyclic group. Groups may be based on either Finite Field Cryptography (FFC) or on Elliptic Curve Cryptography (ECC). Each component of a group is referred to as an "element." Groups are negotiated using an identifying number from a repository maintained by IANA as "Group Description" attributes for IETF RFC 2409 (IKE) [B52]. The repository maps an identifying number to a complete set of domain parameters for the particular group. For the purpose of interoperability, conformant STAs shall support group nineteen (19), an ECC group defined over a 256-bit prime order field.

More than one group may be configured on a STA for use with SAE by using the dot11RSNAConfigDLCGroup table. Configured groups are prioritized in ascending order of preference. If only one group is configured, it is, by definition, the most preferred group.

NOTE—The preference of one group over another is a local policy issue.

SAE uses three arithmetic operators defined for both FFC and ECC groups, an operation that takes two elements to produce a third element (called the "element operation"), an operation that takes an integer (called "scalar") and an element to produce a second element (called the "scalar operation"), and an operation that takes an element to produce a second element (called the "inverse operation"). The convention used here is to represent group elements in uppercase bold italic and scalar values in lowercase italic. The element operation takes two elements, ***X*** and ***Y***, to produce a third element, ***Z***, and is denoted ***Z*** = elem-op(***X***,***Y***); the scalar operation takes a scalar, *x*, and an element, ***Y***, to produce a second element **Z** and is denoted ***Z*** = scalar-op(*x*,***Y***); the inverse operation takes an element, ***X***, to produce a second element, ***Z***, and is denoted ***Z*** = inverse-op(***X***).

scalar-op(*x*,***Y***) is defined as successive iterations of elem-op(***Y***, ***Y***). That is, it is possible to define scalar-op*(*1, ***Y***) = ***Y*** and for *x* > 1, scalar-op(*x*, ***Y***) = elem-op(scalar-op(*x*-1, ***Y***), ***Y****)*. The specific definition of elem-op(**X***,* ***Y***) depends on the type of group, either ECC or FFC.

8.2a.4.2 Elliptic curve cryptography (ECC) groups

8.2a.4.2.1 ECC group definition

ECC groups used by SAE are defined by the sextuple (*p*, *a*, *b*, ***G***, *r*, *h*) where *p* is a prime number, *a* and *b* specify the elliptic curve defined by the equation, $y^2 = x^3 + ax + b$ modulo *p*, ***G*** is a generator (a base point on the elliptic curve), *r* is the prime order of ***G***, and *h* is the co-factor. Elements in ECC groups are the points on the elliptic curve defined by their coordinates—(*x*, *y*)—that satisfy the equation for the curve and the identity element, the so-called "point at infinity."

The IANA registry used to map negotiated numbers to group domain parameters includes some ECC groups defined over a characteristic 2 finite field and may include some ECC groups with a co-factor greater than one (1). These groups shall not be used with SAE. Only ECC groups defined over an odd prime finite field with a co-factor equal to one (1) shall be used with SAE.

The element operation in an ECC group is addition of two points on the curve resulting in a third point on the curve. For example, the point ***X*** is added to the point ***Y*** to produce the point ***Z***:

$$\boldsymbol{Z} = \boldsymbol{X} + \boldsymbol{Y} = \text{elem-op}(\boldsymbol{X},\boldsymbol{Y})$$

The scalar operation in an ECC group is multiplication of a point on the curve by a scalar resulting in a second point on the curve. For example, the point ***Y*** is multiplied by the scalar *x* to produce the point ***Z***:

$\boldsymbol{Z} = x\boldsymbol{Y} = \text{scalar-op}(x, \boldsymbol{Y})$

The inverse operation in an ECC group is inversion of a point on a curve resulting in a second point on the curve. A point on an elliptic curve is the inverse of a different point if their sum is the "point at infinity." In other words:

elem-op(***X***, inverse(***X***)) = "point at infinity"

ECC groups make use of a mapping function, F, that maps a point (x, y) that satisfies the curve equation to its x-coordinate—i.e., if $\boldsymbol{P} = (x, y)$ then $F(\boldsymbol{P}) = x$. Function F is not defined with the identity element as input.

NOTE—SAE protocol operations preclude function F from ever being called with the identity element, i.e., the "point at infinity."

8.2a.4.2.2 Generation of the Password Element with ECC groups

The Password Element of an ECC group (***PWE***) shall be generated in a random hunt-and-peck fashion. The password and a counter, represented as a single octet and initially set to one (1), are used with the peer identities to generate a password seed. The password seed shall then be stretched using the key derivation function (KDF) from 8.5.1.5.2 to a length equal to the bit length of the prime number, *p*, from the elliptic curve domain parameters with the Label being the string "SAE Hunting and Pecking" and with the Context being the prime number. If the resulting password value is greater than or equal to the prime number, the counter shall be incremented, a new password seed shall be derived and the hunting-and-pecking shall continue. Otherwise, it shall be used as the x-coordinate of a candidate point (x, y) on the curve satisfying the curve equation, if such a point exists. If no solution exists, the counter shall be incremented, a new password-seed shall be derived and the hunting-and-pecking shall continue. Otherwise, there will be two possible solutions: (x, y) and $(x, p - y)$. The password seed shall be used to determine which one to use: if the least-significant bit (LSB) of the password seed is equal to that of *y*, the ***PWE*** shall be set to (x, y); otherwise, it shall be set to $(x, p - y)$.

NOTE—The probability that one requires more than *n* iterations of the "hunting and pecking" loop to find ***PWE*** is roughly $(1-(r/2p))^n$, which rapidly approaches zero (0) as *n* increases.

Algorithmically this process is described as follows:

```
found = 0;
counter = 1
z = len(p)
do {
    pwd-seed = H(MAX(STA-A-MAC, STA-B-MAC) || MIN(STA-A-MAC, STA-B-MAC),
        password || counter)
    pwd-value = KDF-z(pwd-seed, "SAE Hunting and Pecking", p)
    if (pwd-value < p)
    then
        x = pwd-value
        if the equation $y^2 = x^3 + ax + b$ modulo p has a solution y
        then
            determine a solution, y, to be the equation $y^2 = x^3 + ax + b$ modulo p
            if LSB(pwd-seed) = LSB(y)
            then
                PWE = (x, y)
            else
                PWE = (x, p – y)
            fi
            found = 1
        fi
    fi
```

```
        counter = counter + 1
    } while (found=0)
```

8.2a.4.3 Finite field cryptography (FFC) groups

8.2a.4.3.1 FFC group definition

FFC groups used by SAE are defined by the triple (p, $\boldsymbol{G}$, r), where p is a prime number, $\boldsymbol{G}$ is a generator, and r is the prime order of $\boldsymbol{G}$ modulo p. An element, $\boldsymbol{B}$, in an FFC group satisfies $\boldsymbol{B} = \boldsymbol{G}^i$ modulo p for some integer i. This special property differentiates elements from scalars, even though both elements and scalars can be represented as non-negative integers less than the prime modulus p. The notation convention of 8.2a.4 signifies this difference between an element and a scalar in an FFC group. The identity element for an FFC group is the value one (1) modulo p.

The element operation in an FFC group is modular multiplication of two elements of this group resulting in a third element of this group. For example, the element $\boldsymbol{X}$ is multiplied by the element $\boldsymbol{Y}$ to product the element $\boldsymbol{Z}$:

$$\boldsymbol{Z} = (\boldsymbol{XY}) \text{ modulo } p = \text{elem-op}(\boldsymbol{X},\boldsymbol{Y})$$

The scalar operation in an FFC group is modular exponentiation of an element of this group by a scalar resulting in a second element of this group. For example, the point $\boldsymbol{Y}$ is raised to the power x to produce the element $\boldsymbol{Z}$:

$$\boldsymbol{Z} = \boldsymbol{Y}^x \text{ modulo } p = \text{scalar-op}(x,\boldsymbol{Y})$$

Some FFC groups in the IANA repository are based on *safe primes*, i.e., a prime, p, of the form $p = 2q + 1$, where q is also a prime number. For these FFC groups, the group generated by $\boldsymbol{G}$ always has order $r = (p - 1)/2$ and thus is uniquely derived from context. For other FFC groups, the parameter r shall be explicitly stated as part of the domain parameters.

The inverse operation in a FFC group is modular inversion of an element of this group producing a second element in this group. An element $\boldsymbol{Z}$ is the inverse of a second element $\boldsymbol{X}$ of this group if their modular product is the identity element of the FCC group. In other words:

$$\text{elem-op}(\boldsymbol{X}, \text{inverse}(\boldsymbol{X})) = 1 \text{ modulo } p$$

In contrast to ECC groups, FFC groups do not need a mapping function that maps an element of the FFC group to an integer (since those elements are already non-negative integers less than the prime number, p). However, for sake of uniform protocol definition, function F with FFC groups is defined as the identity function—i.e., if x is an element of the FFC group then $\text{F}(x) = x$.

8.2a.4.3.2 Generation of the Password Element with FFC groups

The Password Element of an FFC group ($\boldsymbol{PWE}$) shall be generated in a random hunt-and-peck fashion similar to the technique for an ECC group. The password and a counter, represented as a single octet and initially set to one (1), are used with the two peer identities to generate a password seed. The password seed shall then be stretched using the key derivation function (KDF) from 8.5.1.5.2 to a length equal to the bit length of the prime number, p, from the group domain parameters with the Label being the string "SAE Hunting and Pecking" and the Content being the prime number. If the resulting password value is greater than or equal to the prime number, the counter shall be incremented, a new password seed shall be derived, and the hunting-and-pecking shall continue. Otherwise, it shall be raised to the power $(p - 1) / r$ (where p is the prime number and r is the order) modulo the prime number to produce a candidate $\boldsymbol{PWE}$. If the candidate

PWE is greater than one (1), the candidate ***PWE*** becomes the ***PWE***; otherwise, the counter shall be incremented, a new password seed shall be derived, and the hunting-and-pecking shall continue.

Algorithmically this process is described as follows:

```
found = 0;
counter = 1
z = len(p)
do {
        pwd-seed = H(MAX(STA-A-MAC, STA-B-MAC) || MIN(STA-A-MAC, STA-B-MAC),
                password || counter)
        pwd-value = KDF-z(pwd-seed, “SAE Hunting and Pecking”, p)
        if (pwd-value < p)
        then
                PWE = $pwd\text{-}value^{(p-1)/r}$ modulo p
                if (PWE > 1)
                then
                        found = 1
                fi
        fi
        counter = counter + 1
} while (found=0)
```

8.2a.5 SAE protocol

8.2a.5.1 Message exchanges

The protocol consists of two message exchanges, a commitment exchange and a confirmation exchange. The commitment exchange is used to force each party to the exchange to commit to a single guess of the password. The confirmation exchange is used to prove that the password guess was correct. Authentication frames are used to perform these exchanges (see 7.2.3.10 and 8.2a.7.3). The rules for performing these exchanges are specified by the finite state machine in 8.2a.8.

When a party has sent its message in the commit exchange it is said to have *committed* and when it has sent its message in the confirmation exchange it has *confirmed*. The following rules are ascribed to the protocol:

- A party may *commit* at any time
- A party *confirm*s after it has *committed* and its peer has *committed*
- A party *accept*s authentication after a peer has *confirmed*
- The protocol successfully *terminates* after each peer has *accepted*

8.2a.5.2 PWE and secret generation

Prior to beginning the protocol message exchange, the secret element ***PWE*** and two secret values are generated. First, a group is selected, either the most preferred group if the STA is initiating SAE to a peer, or the group from a received Commit Message if the STA is responding to a peer. The ***PWE*** shall be generated for that group (according to 8.2a.4.2.2 or 8.2a.4.3.2, depending on whether the group is ECC or FFC, respectively) using the identities of the two STAs and the configured password.

After generation of the ***PWE***, each STA shall generate a secret value, *rand*, and a temporary secret value, *mask*, each of which shall be chosen randomly such that $1 < rand < r$ and $1 < mask < r$, where *r* is the (prime) order of the group. The values *rand* and *mask* shall be random numbers produced from a quality random number generator. These values shall never be reused on distinct protocol runs.

8.2a.5.3 Construction of a Commit Message

A Commit Message consists of a scalar and an element that shall be produced using the ***PWE*** and secrets generated in 8.2a.5.2, as follows:

$$commit\text{-}scalar = (rand + mask) \text{ modulo } r$$
$$\boldsymbol{COMMIT\text{-}ELEMENT} = \text{inverse}(\text{scalar-op}(mask, \boldsymbol{PWE}))$$

This message shall be transmitted to the peer as described in 8.2a.7. The temporary secret *mask* may be destroyed at this point.

8.2a.5.4 Processing of a peer's Commit Message

Upon receipt of a peer's Commit Message both the scalar and element shall be verified.

If the scalar value is greater than zero (0) and less than the order, *r*, of the negotiated group, scalar validation succeeds, otherwise, it fails. Element validation depends on the type of group. For FFC groups, the element shall be an integer greater than zero (0) and less than the prime number *p*, and the scalar operation of the element and the order of the group, *r*, shall equal one (1) modulo the prime number *p*. If either of these conditions does not hold, element validation fails; otherwise, it succeeds. For ECC groups, both the x- and y-coordinates of the element shall be nonnegative integers less than the prime number *p*, and the two coordinates shall produce a valid point on the curve satisfying the group's curve definition, not being equal to the "point at the infinity." If either of those conditions does not hold, element validation fails; otherwise, element validation succeeds.

If either scalar validation or element validation fails, the STA shall reject the peer's authentication. If both the scalar and element from the peer's Commit Message are successfully validated, a shared secret element, *K*, shall be derived using the scalar and element (*peer-commit-scalar* and ***PEER-COMMIT-ELEMENT***, respectively) from the peer's Commit Message and the STA's secret value.

$$\boldsymbol{K} = \text{scalar-op}(rand, (\text{elem-op}(\text{scalar-op}(peer\text{-}commit\text{-}scalar, \boldsymbol{PWE}),$$
$$\boldsymbol{PEER\text{-}COMMIT\text{-}ELEMENT})))$$

If the shared secret element, ***K***, is the identity element for the negotiated group (the value one for an FFC group or the point-at-infinity for an ECC group) the STA shall reject the peer's authentication. Otherwise, a secret value, *k*, shall be computed as:

$$k = \text{F}(\boldsymbol{K})$$

The entropy of *k* shall then be extracted using H to produce *keyseed*. The key derivation function from 8.5.1.5.2 shall then be used to derive a key confirmation key, KCK, and a pairwise master key, PMK, from *keyseed*. When used with AKMs 8 or 9, the salt shall consist of thirty-two (32) octets of the value zero (0) (indicated below as <0>32) and both the KCK and PMK shall be 256-bits in length. Use of other AKMs require definition of the lengths of the salt, the KCK, and the PMK.

$$keyseed = \text{H}(<0>32, k)$$
$$KCK \parallel PMK = \text{KDF-512}(keyseed, \text{“SAE KCK and PMK”},$$
$$(commit\text{-}scalar + peer\text{-}commit\text{-}scalar) \text{ modulo } r)$$

The PMK identifier is defined as follows:

$$\text{PMKID} = \text{L}((commit\text{-}scalar + peer\text{-}commit\text{-}scalar) \text{ modulo } r, 0, 128)$$

8.2a.5.5 Construction of a Confirm Message

A peer generates a Confirm Message by passing the KCK, the current value of the *send-confirm* counter (see 7.3.1.35), the scalar and element from the sent Commit Message, and the scalar and element from the received Commit Message to the confirmation function CN.

confirm = CN(KCK*, send-confirm, commit-scalar,* ***COMMIT-ELEMENT***, *peer-commit-scalar,*
PEER-COMMIT-ELEMENT)

The message shall be transmitted to the peer as described in 8.2a.7.

8.2a.5.6 Processing of a peer's Confirm Message

Upon receipt of a peer's Confirm Message a *verifier* is computed, which is the expected value of the peer's confirmation, *peer-confirm*, extracted from the received Confirm Message. The *verifier* is computed by passing the KCK, the peer's send-confirm counter from the received Confirm Message (see 7.3.1.35), the scalar and element from the received Commit Message, and scalar and element from the sent Commit Message to the confirmation function CN.

verifier = CN(KCK*, peer-send-confirm, peer-commit-scalar,* ***PEER-COMMIT-ELEMENT***,
commit-scalar, ***COMMIT-ELEMENT***)

If the *verifier* equals *peer-confirm,* the STA shall accept the peer's authentication and set the lifetime of the PMK to the value dot11RSNAConfigPMKLifetime. If the *verifier* differs from the *peer-confirm,* the STA shall reject the peer's authentication and destroy the PMK.

8.2a.6 Anti-clogging tokens

A STA is required to do a considerable amount of work upon receipt of a Commit Message. This opens up the possibility of a distributed denial-of-service attack by flooding a STA with bogus Commit Messages from forged MAC addresses. To prevent this from happening, a STA shall maintain an *Open* counter in its SAE state machine indicating the number of open and unfinished protocol instances (see 8.2a.5.1). When that counter hits or exceeds dot11RSNASAEAntiCloggingThreshold, the STA shall respond to each Commit Message with a rejection that includes an Anti-Clogging Token statelessly bound to the sender of the Commit Message. The sender of the Commit Message shall then include this Anti-Clogging Token in a subsequent Commit Message.

The Anti-Clogging Token is a variable-length value that statelessly binds the MAC address of the sender of a Commit Message. The length of the Anti-Clogging Token needs not be specified because the generation and processing of the Anti-Clogging Token is solely up to one peer. To the other peer in the SAE protocol, the Anti-Clogging Token is merely an opaque blob whose length is insignificant. It is suggested that an Anti-Clogging Token not exceed 256 octets.

NOTE—A suggested method for producing Anti-Clogging Tokens is to generate a random secret value each time the state machine variable hits dot11RSNASAEAntiCloggingThreshold and pass that secret and the MAC address of the sender of the Commit Message to the random function H to generate the token.

As long as the state machine variable *Open* is greater than or equal to dot11RSNASAEAntiCloggingThreshold all Commit Messages that do not include a valid Anti-Clogging Token shall be rejected with a request to repeat the Commit Message and include the token (see 8.2a.5.1).

Since the Anti-Clogging Token is of fixed size and the size of the *peer-commit-scalar* and ***PEER-COMMIT-ELEMENT*** are inferred from the finite cyclic group being used, it is straightforward to determine whether a received Commit Message includes an Anti-Clogging Token or not.

Encoding of the Anti-Clogging Token and its placement with respect to the *peer-commit-scalar* and ***PEER-COMMIT-ELEMENT*** is described in 8.2a.7.4.

8.2a.7 Framing of SAE

8.2a.7.1 General

Commit Messages and Confirm Messages are sent and received by a SAE protocol using IEEE 802.11 Authentication frames.

8.2a.7.2 Data type conversion

8.2a.7.2.1 General

This protocol requires elements in finite cyclic groups to be converted to octet strings prior to transmission and back again upon receipt. To convert an element into an octet string, the first step is to represent the element in integer format and then employ an integer-to-octet string conversion prior to transmission. To convert an octet string into an element requires an octet string to integer conversion and then representing the integer(s) as an element.

8.2a.7.2.2 Integer to octet string conversion

An integer, x, shall be converted into an octet string of length m such that $2^{8\mathrm{m}} > x$ by first representing x in its binary form and then converting the result to an octet-string.

Given x, m, represent x as a sequence of x_{m-i} base 2^8:

$$x = x_{m\text{-}1} \times 2^{8(m\text{-}1)} + x_{m\text{-}2} \times 2^{8(m\text{-}2)} + \ldots + x_1 \times 2^8 + x_0$$

then let the octet M_i have the value x_i for $0 \le i \le m{-}1$ and the octet string shall be $\mathrm{M}_{m\text{-}1} \,||\, \mathrm{M}_{m\text{-}2} \,||\, \ldots \,||\, \mathrm{M}_1 \,||\, \mathrm{M}_0$ where || symbolizes concatenation.

8.2a.7.2.3 Octet string to integer conversion

An octet string shall be converted into an integer by viewing the octet string as the base 2^8 representation of the integer.

$$x = \sum_{i=1}^{m} 2^{8(m\text{-}i)} \times \mathrm{M}_{m\text{-}i}$$

8.2a.7.2.4 Element to octet string conversion

For ECC groups, each element, except the “point at infinity,” is a point on the elliptic curve satisfying the curve equation and consists of two components: an x-coordinate and a y-coordinate. To convert this point to an octet string, each component shall be treated as an integer and converted into an octet string whose length is the smallest integer m such that $2^{8m} > p$, where p is the prime number specified y the elliptic curve domain parameters, according to 8.2a.7.2.2. The point shall be represented as the concatenation of the x-coordinate and the y-coordinate, each represented as an octet string of length m octets, and is $2m$ octets long.

For FFC groups each element is a non-negative integer less than the prime number p specified by the FFC domain parameters. To convert this element into an octet string, it shall be treated directly as an integer and converted into an octet string whose length is the smallest integer m such that $2^{8m} > p$, where p is the prime number specified by the domain parameters, according to 8.2a.7.2.2.

8.2a.7.2.5 Octet string to element conversion

To convert an octet string into a point on an elliptic curve it is necessary to divide it into two octet strings of equal length *m*. If the length of the octet string does not evenly divide by two, conversion shall fail. Each octet string of length *m* shall be converted to an integer according to 8.2a.7.2.3. The first octet string conversion produces an integer that becomes the x-coordinate of the point and the second octet string conversion produces an integer that becomes the y-coordinate of the point. If either integer equals zero (0) or is greater than or equal to *p*, the prime from the elliptic curve domain parameters, conversion shall fail. If the resulting (*x*, *y*) point does not satisfy the equation of the curve, or produces the "point at infinity," conversion shall fail.

To convert an octet string into an element in a prime modulus group the octet string shall be converted into an integer according to 8.2a.7.2.3 and the integer shall be used directly as the group element.

8.2a.7.3 Authentication transaction sequence number for SAE

A Commit Message shall use Authentication Transaction Sequence Number one (1). A Confirm Message shall use Authentication Transaction Sequence Number two (2).

8.2a.7.4 Encoding and decoding of Commit Messages

A Commit Message shall be encoded as an IEEE 802.11 Authentication frame with an Authentication Algorithm of three (3), a Transaction Sequence Number of one (1) and a Status Code of zero (0). Non-zero status codes indicate a rejection of a peer's Commit Message and are described in 8.2a.7.6.

A Commit Message shall consist of a Finite Cyclic Group field (7.3.1.40) indicating the desired group, a Scalar field (7.3.1.37) containing the scalar, and an Element field containing the element (7.3.1.38). If the Commit Message is in response to an Anti-Clogging Token request (see 8.2a.7.6), the Anti-Clogging Token is present (see 7.3.1.36).

When transmitting a Commit Message, the scalar and element shall be converted to octet strings and placed in the Scalar field and Element field, respectively. The scalar shall be treated as an integer and converted into an octet string of length *m* such that $2^{8m} > r$, where *r* is the order of the group, according to 8.2a.7.2.2, and the element shall be converted into (an) octet string(s) according to 8.2a.7.2.4. When receiving a Commit Message the component octet strings in the Scalar field and Element field shall be converted into a scalar and element, respectively, according to 8.2a.7.2.3 and 8.2a.7.2.5, respectively.

8.2a.7.5 Encoding and decoding of Confirm Messages

A Confirm Message shall be encoded as an IEEE 802.11 Authentication frame with an Authentication Algorithm of three (3), a Transaction Sequence Number of two (2) and a Status Code of zero (0). Non-zero status codes indicate rejection of a peer's Confirm Message and are described in 8.2a.7.6.

A Confirm Message shall consist of a Send-Confirm field (7.3.1.35) and a Confirm field (7.3.1.39) containing the output of the random function as described in 8.2a.5.5. When transmitting a Confirm Message the output of the random function shall be treated as an integer and converted into an octet string of length *m*, where *m* is the block size of the random function, according to 8.2a.7.2.2 and placed in the Confirm field. When receiving a Confirm Message, the octet string in the Confirm field shall be converted into an integer representing the peer's Confirm according to 8.2a.7.2.3.

8.2a.7.6 Status codes

A Commit Message with a non-zero status code shall indicate that a peer rejects a previously sent Commit Message. An unsupported finite cyclic group is indicated with a status code of 77, "Authentication is rejected because the offered finite cyclic group is not supported." An Anti-Clogging Token is requested by transmitting a Commit Message with a status code of 76, "Anti-Clogging Token Requested," with the Anti-Clogging Token occupying the Token field of the Authentication frame.

A Confirm Message, with a non-zero status code, shall indicate that a peer rejects a previously sent Confirm Message. A Confirm Message that was not successfully verified is indicated with a status code of fifteen (15), "Authentication rejected; the response to the challenge failed."

8.2a.8 SAE finite state machine

8.2a.8.1 General

The protocol is instantiated by the finite state machine in Figure 8-3a. Each instance of the protocol is identified by a tuple consisting of the local MAC address and the peer MAC address. The model in which SAE is defined consists of a parent process, managed by the SME, which receives messages, and dispatches them to the appropriate protocol instance, also managed by the SME. The parent process manages a database of protocol instances indexed by the peer identity. Protocol instances maintain state, receive events from the parent process, send events to itself, and output data.

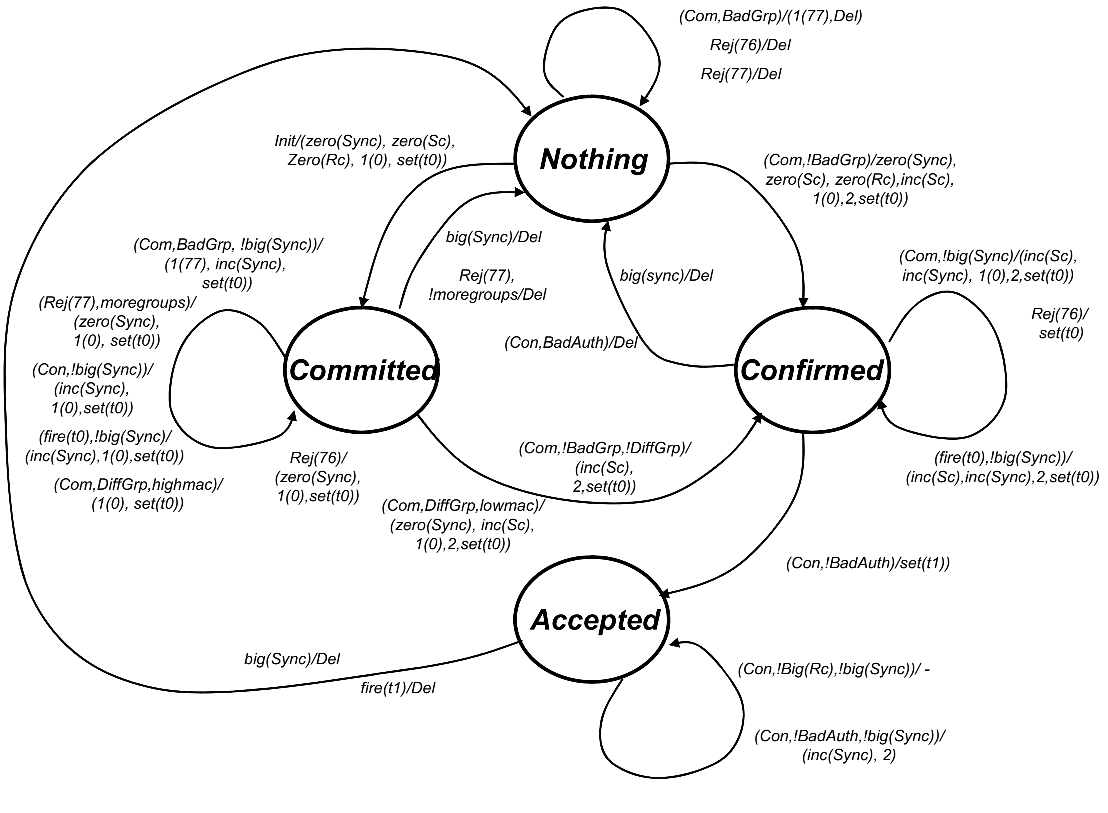

(Com,BadGrp)/(1(77),Del)
Rej(76)/Del
Rej(77)/Del
Nothing
Init/(zero(Sync), zero(Sc), Zero(Rc), 1(0), set(t0))
(Com,!BadGrp)/zero(Sync), zero(Sc), zero(Rc),inc(Sc), 1(0),2,set(t0))
big(Sync)/Del
Rej(77), !moregroups/Del
big(sync)/Del
(Com,BadGrp, !big(Sync))/ (1(77), inc(Sync), set(t0))
(Com,!big(Sync)/(inc(Sc), inc(Sync), 1(0),2,set(t0))
(Rej(77),moregroups)/ (zero(Sync), 1(0), set(t0))
Rej(76)/ set(t0)
(Con,BadAuth)/Del
Committed
Confirmed
(Con,!big(Sync))/ (inc(Sync), 1(0),set(t0))
(fire(t0),!big(Sync)/ (inc(Sync),1(0),set(t0))
(Com,DiffGrp,highmac)/ (1(0), set(t0))
Rej(76)/ (zero(Sync), 1(0),set(t0))
(Com,!BadGrp,!DiffGrp)/ (inc(Sc), 2,set(t0))
(fire(t0),!big(Sync))/ (inc(Sc),inc(Sync),2,set(t0))
(Com,DiffGrp,lowmac)/ (zero(Sync), inc(Sc), 1(0),2,set(t0))
(Con,!BadAuth)/set(t1))
Accepted
big(Sync)/Del
fire(t1)/Del
(Con,!Big(Rc),!big(Sync))/ -
(Con,!BadAuth,!big(Sync))/ (inc(Sync), 2)

Figure 8-3a—SAE finite state machine

NOTE—Figure 8-3a does not show all state machine transitions. A full description of the SAE finite state machine is in 8.2a.8.6.2.

The parent process instantiates protocol instances upon receipt of SAE messages and initiation by SME. The parent process also maintains a counter of the number of protocol instances created.

8.2a.8.2 States

8.2a.8.2.1 Parent process states

The parent process is in a continuous quiescent state.

8.2a.8.2.2 Protocol instance states

Each protocol instance is in one of the following four (4) states:

a) *Nothing*—The *Nothing* state represents the initial state of a freshly allocated protocol instance or the terminal state of a soon-to-be deallocated protocol instance. Freshly created protocol instances will immediately transition out of *Nothing* state depending on the reason for their creation. Protocol instances that transition into *Nothing* state will immediately be destroyed with their state zeroed and returned to the memory pool.

b) *Committed*—In the *Committed* state, the finite state machine has sent a Commit Message and is awaiting a Commit Message and a Confirm Message from the peer.

c) *Confirmed*—In the *Confirmed* state, the finite state machine has sent both a Commit Message and a Confirm Message and received a Commit Message. It awaits a Confirm Message.

d) *Accepted*—In the *Accepted* state, the protocol instance has both sent and received a Commit Message and a Confirm Message and the protocol instance has finished.

8.2a.8.3 Events and output

8.2a.8.3.1 Parent process events and output

The parent process receives events from three (3) sources: the SME, protocol instances, and received frames.

The SME signals the following events to the parent SAE process:

a) *Initiate*—An *Initiate* event is used to instantiate a protocol instance to begin SAE with a designated peer.

b) *Kill*—A *Kill* event is used to remove a protocol instance with a designated peer.

Protocol instances send the following events to the SAE parent process:

c) *Fail*—The peer failed to be authenticated.

d) *Auth*—The peer was successfully authenticated.

e) *Del*—The protocol instance has had a fatal event.

Receipt of frames containing SAE messages signals the following events to the SAE parent process:

f) *IEEE 802.11 Authentication frame with Transaction Sequence number 1*—This event indicates that a Commit Message has been received from a peer STA.

g) *IEEE 802.11 Authentication frames with Transaction Sequence number 2*—This event indicates that a Confirm Message has been received from a peer STA.

The parent process generates IEEE 802.11 Authentication frames with Authentication transaction sequence 1 and a Status of 76 indicating rejection of an Authentication attempt because an Anti-Clogging Token is required.

8.2a.8.3.2 Protocol instance events and output

The protocol instance receives events from the parent SAE process.

a) *Com*—Indicates receipt of a Commit Message (Authentication transaction sequence number 1) with a status of zero (0).

b) *Con*—Indicates receipt of a Confirm Message (Authentication transaction sequence number 2) with a status of zero (0).

c) *Init*—Indicates that the protocol instance should begin negotiation with a specified peer.

d) *Rej(N)*—Indicates receipt of a rejected Commit Message with status *N*.

In addition, protocol instances receive *fire(X)* events indicating expiry of timer *X*. Upon expiry of a timer and generation of a *fire()* event, the expired timer is not reset.

The protocol instance generates output from the following events:

e) *1(N)*—Indicates generation of a Commit Message (Authentication transaction sequence number 1) with status *N*.

f) *2*—Indicates generation of a Confirm Message (Authentication transaction sequence number 2).

8.2a.8.4 Timers

The parent SAE process does not use timers. Each protocol instance can set timers that result in *fire()* events to be sent to itself. The following timers can be set:

a) t0—A retransmission timer.

b) t1—A PMK expiry timer.

Timers are set by the protocol instance issuing a *set()* for the particular timer.

8.2a.8.5 Variables

8.2a.8.5.1 Parent process variables

The parent SAE process maintains a counter, *Open*, which indicates the number of protocol instances in either *Committed* or *Confirmed* state. When the parent SAE process starts up, *Open* is set to zero (0).

The parent process maintains a database of protocol instances.

NOTE—Depending on how Anti-Clogging Tokens (see 8.2a.6) are constructed, the parent SAE process might also maintain a random secret used for token creation.

8.2a.8.5.2 Protocol instance variables

Each protocol instance maintains the following three variables:

a) *Sync*—The number of state resynchronizations that have occurred.

b) *Sc*—The number of Confirm messages that have been sent. This is the send-confirm counter used in the construction of Confirm messages (see 8.2a.5.5).

c) *Rc*—The received value of the *send-confirm* counter in the last received Confirm Message. In other words, this is the value of the peer's send-confirm counter.

Function *zero(X)* assigns the value zero (0) to the variable *X*, *inc(X)* increments the variable *X*, and *big(X)* indicates that the variable *X* has exceeded a maximum value.

In addition, protocol instances maintain the following six indicators that are not maintained as state variables but, instead, indicate the cause of certain behavior.

d) *BadGrp*—The group specified in a Commit Message is not supported.

e) *DiffGrp*—The group specified in a Commit Message is supported but differs from the one offered.

f) *BadConf*—The contents of a confirm frame were incorrect.

g) *highmac*—The peer identity is numerically less than the local identity.

h) *lowmac*—The peer identity is numerically greater than the local identity.

i) *moregroups*—There are finite cyclic groups in the configuration that have not been offered to the peer.

A negative indication is shown with an exclamation point (!)—e.g., "the group specified in a Commit Message is supported" would be !BadGrp, which is read as "not BadGrp."

8.2a.8.6 Behavior of state machine

8.2a.8.6.1 Parent process behavior

For any given peer identity, there shall be only one protocol instance in *Committed* or *Confirmed* state. Similarly, for any given peer identity, there shall be only one protocol instance in *Accepted* state.

The parent process creates protocol instances based upon different actions. Creating a protocol instance entails allocation of state necessary to maintain the protocol instance state machine, putting the protocol instance in *Nothing* state, incrementing the *Open* counter, and inserting the protocol instance into its database indexed by the MAC address of the peer with whom the protocol instance will communicate.

The parent process also destroys protocol instances by zeroing out the state of the protocol instance and returning it to the memory pool.

Upon receipt of an *Initiate* event, the parent process shall check whether there exists a protocol instance for the peer MAC address (from the *Init* event) in either *Committed* or *Confirmed* state. If there is, the *Initiate* event shall be ignored. Otherwise, a protocol instance shall be created, and an *Init* event shall be sent to the protocol instance.

Upon receipt of a *Kill* event, the parent process shall destroy all protocol instances indexed by the peer MAC address (from the *Kill* event) in its database. For each protocol instance in *Committed* or *Confirmed* state, the *Open* counter shall be decremented.

Upon receipt of a *Sync, Del,* or *Fail* event from a protocol instance, the parent process shall decrement the *Open* counter and destroys the protocol instance.

Upon receipt of an *Auth* event from a protocol instance, the parent process shall decrement the *Open* counter. If another protocol instance exists in the database indexed by the same peer identity as the protocol instance that sent the *Auth* event, the other protocol instance shall be destroyed.

Upon receipt of a Commit Message, the parent process checks whether a protocol instance for the peer MAC address exists in the database. If one does, and it is in either *Committed* state or *Confirmed* state the frame shall be passed to the protocol instance. If one does and it is in Authenticated state, the scalar in the received frame is checked against the *peer-scalar* used in authentication of the existing protocol instance (in Authenticated state). If it is identical, the frame shall be dropped. If not, the parent process checks the value of *Open*. If *Open* is greater than dot11RSNASAEAntiCloggingThreshold, the parent process shall check for the presence of an Anti-Clogging Token. If an Anti-Clogging Token exists and is correct, the parent process shall create a protocol instance. If the Anti-Clogging Token is incorrect, the frame shall be silently discarded. If Open is greater than dot11RSNASAEAntiCloggingThreshold and there is no Anti-Clogging Token in the received frame, the parent process shall construct a response as an IEEE 802.11 Authentication frame with Authentication sequence number one (1), Status code 76, and the body of the frame consisting of an Anti-Clogging Token (see 8.2a.6). If *Open* is not greater than dot11RSNASAEAntiCloggingThreshold, the parent process shall create a protocol instance and the frame shall be sent to the protocol instance as a *Com* event.

Upon receipt of a Confirm Message, the parent process checks whether a protocol instance for the peer MAC address (as indicated by the SA in the received frame) exists in the database. If there is a single protocol instance, the frame shall be passed to it as a *Con* event. If there are two (2) protocol instances indexed by that peer MAC address, the frame shall be passed, as a *Con* event, to the protocol instance that is not in *Accepted* state. If there are no protocol instances indexed by that peer MAC address, the frame shall be dropped.

8.2a.8.6.2 Protocol instance behavior

8.2a.8.6.2a General

State machine behavior is illustrated in Figure 8-3a. The protocol instance receives events from the parent process and from itself. It generates SAE messages that are transmitted to a peer and sends events to itself and the parent process.

The semantics of the state diagram are "occurrence/behavior" where "occurrence" is a comma-separated list of events and/or indicators, or the special symbol "—" indicating no occurrence; and, "behavior" is a comma-separated list of outputs and/or functions, or the special symbol "—" indicating no behavior.

When the state machine calls for the t0 (retransmission) timer to be set, it shall be set to the value of dot11RSNASAERetransPeriod. When the state machine calls for the t1 (key expiry) timer to be set, it shall be set to the value of dot11RSNAConfigPMKLifetime.

8.2a.8.6.2b Nothing state

In *Nothing* state a protocol instance has just been allocated.

Upon receipt of an *Init* event, the protocol instance shall zero its *Sync* variable, *Rc*, and *Sc* variables, select a group from local configuration and generate the ***PWE*** and the secret values according to 8.2a.5.2, generate a Commit Message (see 8.2a.5.3), and set its t0 (retransmission) timer. The protocol instance transitions into *Committed* state.

Upon receipt of a *Com* event, the protocol instance shall check the Status of the Authentication frame. If the Status code is non-zero, the frame shall be silently discarded and a *Del* event shall be sent to the parent process.Otherwise, the frame shall be processed by first checking the finite cyclic group field to see if the requested group is supported. If not, *BadGrp* shall be set and the protocol instance shall construct and transmit a an Authentication frame with Status code 77 indicating rejection with the finite cyclic group field set to the rejected group, and shall send the parent process a *Del* event. If the group is supported, the protocol instance shall zero the *Sc* and *Rc* counters and it shall generate the ***PWE*** and the secret values according to 8.2a.5.2. It shall then process the received Commit Message (see 8.2a.5.4). If validation of the received Commit Message fails, the protocol instance shall send a Del event to the parent process; otherwise, it shall construct and transmit a Commit Message (see 8.2a.5.3) followed by a Confirm Message (see 8.2a.5.5). The *Sync* counter shall be set to zero and the t0 (retransmission) timer shall be set. The protocol instance transitions to *Confirmed* state.

NOTE—A protocol instance in *Nothing* state will never receive a Confirm Message due to state machine behavior of the parent process.

8.2a.8.6.2c Committed state

In *Committed* state, a protocol instance has sent its peer a Commit Message but has yet to receive (and accept) anything.

Upon receipt of a *Com* event, the t0 (retransmission) timer shall be cancelled. Then the following is performed:

- The protocol instance shall check the Status code of the Authentication frame. If the Status code is 76, a new Commit Message shall be constructed with the Anti-Clogging Token from the received Authentication frame, and the *commit-scalar* and ***COMMIT-ELEMENT*** previously sent. The new Commit Message shall be transmitted to the peer, *Sync* shall be zeroed, and the t0 (retransmission) timer shall be set.
- If the Status code is 77, the protocol instance shall check the finite cyclic group field being rejected. If the rejected group does not match the last offered group the protocol instance shall silently discard the message and set the t0 (retransmission) timer. If the rejected group matches the last offered group, the protocol instance shall choose a different group and generate the ***PWE*** and the secret values according to 8.2a.5.2; it then generates and transmits a new Commit Message to the peer, zeros *Sync*, sets the t0 (retransmission) timer, and remains in *Committed* state. If there are no other groups to choose, the protocol instance shall send a *Del* event to the parent process and transitions back to *Nothing* state.
- If the Status is some other non-zero value, the frame shall be silently discarded and the t0 (retransmission) timer shall be set.
- If the Status is zero, the finite cyclic group field is checked. If the group is not supported, *BadGrp* shall be set and the value of *Sync* shall be checked.
  - If *Sync* is greater than dot11RSNASAESync, the protocol instance shall send a *Del* event to the parent process and transitions back to *Nothing* state.
  - If *Sync* is not greater than dot11RSNASAESync, *Sync* shall be incremented, a Commit Message with Status code equal to 77 indicating rejection, and the Algorithm identifier set to the rejected algorithm, shall be sent to the peer, the t0 (retransmission) timer shall be set and the protocol instance shall remain in *Committed* state.
- If the group is supported but does not match that used when the protocol instance constructed its Commit Message, *DiffGrp* shall be set and the local identity and peer identity shall be checked.
  - The mesh STA, with the numerically greater of the two MAC addresses, drops the received Commit Message, retransmits its last Commit Message, and shall set the t0 (retransmission) timer and remain in *Committed* state.
  - The mesh STA, with the numerically lesser of the two MAC addresses, zeros *Sync*, shall increment *Sc*, choose the group from the received Commit Message, generate new ***PWE*** and new secret values according to 8.2a.5.2, process the received Commit Message according to 8.2a.5.4, generate a new Commit Message and Confirm Message, and shall transmit the new Commit and Confirm to the peer. It shall then transition to *Confirmed* state.
- If the group is supported and matches that used when the protocol instance constructed its Commit Message, the protocol instance checks the *peer-commit-scalar* and ***PEER-COMMIT-ELEMENT*** from the message. If they match those sent as part of the protocol instance's own Commit Message, the frame shall be silently discarded (because it is evidence of a reflection attack) and the t0 (retransmission) timer shall be set. If the received element and scalar differ from the element and scalar offered, the received Commit Message shall be processed according to 8.2a.5.4, the *Sc* counter shall be incremented (thereby setting its value to one), the protocol instance shall then construct a Confirm Message, transmit it to the peer, and set the t0 (retransmission) timer. It shall then transition to *Confirmed* state.

If the t0 (retransmission) timer fires, the value of the *Sync* counter is checked. If *Sync* is greater than dot11RSNASAESync, the protocol instance shall send a *Del* event to the parent process and transition back to *Nothing* state. If *Sync* is not greater than dot11RSNASAESync, the *Sync* counter shall be incremented, the last message sent shall be sent again, and the t0 (retransmission) timer shall be set.

Upon receipt of a *Con* event, the t0 (retransmission) timer shall be cancelled. Then the protocol instance checks the value of *Sync*. If it is greater than dot11RSNASAESync, the protocol instance shall send a *Del* event to the parent process and transition back to *Nothing* state. If *Sync* is not greater than dot11RSNASAESync, the protocol instance shall increment *Sync*, transmit the last Commit Message sent to the peer, and set the t0 (retransmission) timer.

8.2a.8.6.2d Confirmed state

In *Confirmed* state, a protocol instance has sent its peer a Commit Message and Confirm Message. It has received a Commit Message from its peer.

Rejection frames received in Confirmed state shall be silently discarded.

Upon receipt of a *Com* event, the t0 (retransmission) timer shall be cancelled. If the Status is non-zero, the frame shall be silently discarded, the t0 (retransmission) timer set, and the protocol instance shall remain in the *Confirmed* state. If *Sync* is greater than dot11RSNASAESync, the protocol instance shall send the parent process a *Del* event and transitions back to *Nothing* state. If *Sync* is not greater than dot11RSNASAESync, the protocol instance shall verify that the finite cyclic group is the same as the previously received Commit frame. If not, the frame shall be silently discarded. If so, the protocol instance shall increment *Sync*, increment *Sc*, and transmit its Commit and Confirm (with the new *Sc* value) messages. It then shall set the t0 (retransmission) timer.

Upon receipt of a *Con* event, the t0 (retransmission) timer shall be cancelled and the Confirm Message shall be processed according to 8.2a.5.6. If processing is successful and the Confirm Message has been verified, the *Rc* variable shall be set to the send-confirm portion of the frame, *Sc* shall be set to the value $2^{16} – 1$, the t1 (key expiry) timer shall be set, and the protocol instance shall transition to *Accepted* state.

If the t0 (retransmission) timer fires, the value of the *Sync* counter shall be checked. If *Sync* is greater than dot11RSNASAESync, the protocol instance shall send a *Del* event to the parent process and transition back to *Nothing* state. If *Sync* is not greater than dot11RSNASAESync, the *Sync* counter shall be incremented, *Sc* shall be incremented, and the protocol instance shall create a new Confirm (with the new *Sc* value) Message, transmit it to the peer, and set the t0 (retransmission) timer.

8.2a.8.6.2e Accepted state

In *Accepted* state, a protocol instance has sent a Commit Message and a Confirm Message to its peer and received a Commit Message and Confirm Message from the peer. Unfortunately, there is no guarantee that the final Confirm Message sent by the STA was received by the peer.

Upon receipt of a *Con* event, the *Sync* counter shall be checked. If the value is greater than dot11RSNASAESync, the protocol instance shall send a *Del* event to the parent process and shall transition to *Nothing* state. If the value of *Sync* is not greater than dot11RSNASAESync, the value of send-confirm shall be checked. If the value is not greater than *Rc* or is equal to $2^{16} – 1$, the received frame shall be silently discarded. Otherwise, the Confirm portion of the frame shall be checked according to 8.2a.5.6. If the verification fails, the received frame shall be silently discarded. If the verification succeeds, the *Rc* variable shall be set to the send-confirm portion of the frame, the *Sync* shall be incremented and a new Confirm Message shall be constructed (with *Sc* set to $2^{16} – 1$) and sent to the peer. The protocol instance shall remain in *Accepted* state.

If the t1 (key expiry) timer fires, the protocol instance shall send the parent process a *Del* event and transition to *Nothing* state.

8.4 RSNA security association management

8.4.1 Security associations

8.4.1.1 Security association definitions

Change the bulleted list in 8.4.1.1 as follows:

- PMKSA: A result of a successful IEEE 802.1X exchange, SAE authentication, preshared PMK information, or PMK cached via some other mechanism.
- PMK-R0 security association: A result of a successful FT initial mobility domain association.
- PMK-R1 security association: A result of a successful FT initial mobility domain association or FT authentication sequence.
- Mesh PMKSA: A result of successful completion of the active authentication protocol.
- PTKSA: A result of a successful 4-Way Handshake, FT 4-Way Handshake, or a FT authentication sequence.
- Mesh TKSA: A result of a successful authenticated mesh peering exchange (AMPE).
- GTKSA: A result of a successful Group Key Handshake, 4-Way Handshake, FT 4-Way Handshake, or FT authentication sequence.
- IGTKSA: A result of a successful Group Key Handshake, successful 4-way Handshake, FT 4-Way Handshake, or the Reassociation Response message of the Fast BSS Transition protocol.
- Mesh GTKSA: A result of a successful AMPE or mesh group key handshake.
- SMKSA: A result of a successful initial SMK Handshake.
- STKSA: A result of a successful 4-way STK Handshake following the initial SMK Handshake or subsequent rekeying.

8.4.1.1.1 PMKSA

Change 8.4.1.1.1 as follows:

When the PMKSA is the result of a successful IEEE 802.1X authentication, it is derived from the EAP authentication and authorization parameters provided by the AS. When the PMKSA is the result of a successful SAE authentication, it is generated as a result of the successful completion of the SAE exchange. This security association is bidirectional. In other words, both parties use the information in the security association for both sending and receiving. The PMKSA is created by the Supplicant's SME when the EAP authentication completes successfully or the PSK is configured. The PMKSA is created by the Authenticator's SME when the PMK is created from the keying information transferred from the AS, when IEEE 802.1X authentication is utilized, or when the SAE exchange successfully completes or the PSK is configured. The PMKSA is used to create the PTKSA. PMKSAs are cached for up to their lifetimes. The PMKSA consists of the following elements:

- PMKID, as defined in 8.5.1.2. The PMKID identifies the security association.
- Authenticator's or peer's MAC address.
- PMK.
- Lifetime, as defined in 8.5.1.2.
- AKMP.
- All authorization parameters specified by the AS or local configuration. This can include parameters such as the STA's authorized SSID.

8.4.1.1.1a PMK-R0 security association

Change the first paragraph in 8.4.1.1.1a as follows:

The PMK-R0 security association is the result of a successful completion of the IEEE 802.1X authentication, SAE authentication, or use of PSK during the FT initial mobility domain association. This security association is bidirectional. It consists of the following elements:

8.4.1.1.1b PMK-R1 security association

Change the first dashed list item after the first paragraph in 8.4.1.1.1b as follows:

- A successful completion of the IEEE 802.1X authentication, SAE authentication, or use of PSK during the FT initial mobility domain association or

Insert the following new subclause immediately after 8.4.1.1.1b:

8.4.1.1.1c Mesh PMKSA

The mesh PMKSA is the result of successful completion of the active authentication protocol. This security association is bidirectional. The two authenticated parties use the information in the security association for both sending and receiving. The mesh PMKSA is created by the Mesh STA's SME when the active authentication protocol completes successfully with the peer mesh STA. The mesh PMKSA is used to create the mesh TKSA. Mesh PMKSAs are cached for up to their lifetimes. Mesh PMKSAs contain the following elements, and are identified by their PMKID.

- PMKID, as defined in 8.2a.5.4
- Mesh STA's MAC address
- Peer mesh STA's MAC address
- PMK
- AEK, as defined in 8.8.1
- Lifetime, as defined in 8.5.1.2
- Selected AKM suite (see 7.3.2.25.2)

Insert the following new subclause immediately after 8.4.1.1.2 (PTKSA):

8.4.1.1.2a Mesh TKSA

The mesh TKSA is a result of the AMPE. This security association is also bidirectional. The mesh TKSA shall be deleted when the lifetime expires. The mesh TKSA contains the following elements:

- MTK, as defined in 8.8.1
- PMKID
- local mesh STA MAC address
- peer mesh STA MAC address
- local Link ID
- peer Link ID
- local nonce
- peer nonce
- Lifetime
- Pairwise cipher suite selector

Insert the following new subclause immediately after 8.4.1.1.3a (IGTKSA):

8.4.1.1.3b Mesh GTKSA

The mesh GTKSA results from a successful AMPE or mesh group key handshake, and is unidirectional. In an MBSS, each mesh STA defines its own "transmit mesh GTKSA," which is used to encrypt its group addressed transmissions. Also, each mesh STA stores a separate "receive mesh GTKSA" for each peer mesh STA so that encrypted group addressed traffic received from the peer mesh STAs may be decrypted.

A transmit mesh GTKSA is created by a mesh STA after the SME has changed the mesh GTK (MGTK) and the new MGTK has been sent to all peer mesh STAs. A receive mesh GTKSA is created by a mesh STA after successfully completing the AMPE in which a wrapped MGTK has been received, or after receiving a valid Message 1 of the mesh group key handshake. The receive mesh GTKSA shall be deleted when the lifetime expires or a new receive mesh GTKSA is created with the same Key ID for the same MGTK source mesh STA. See 11C.6.1.

The MGTK and the GTK shall be independently selected from a uniform distribution. The MGTK source mesh STA MAC address in the mesh GTKSA shall not be the same as the Authenticator MAC address in the GTKSA.

NOTE—The use of a distinct Transmit MGTK and ESS GTK with identical transmit MAC addresses is precluded by limitations on key rollover and reception by STAs in an ESS (see 11C.10.5 for collocated mesh STA rules). If the distinct MGTKs were to use different Key IDs, then rollover would be impossible. Since the Key ID 0 is reserved for individually addressed frame transmission, there are only three available Key IDs, and the different MGTKs would contend for the single remaining Key ID upon rollover. If the distinct MGTKs were to use the same Key IDs, then STAs would incorrectly attempt to decrypt mesh broadcast traffic using the ESS GTK, causing error counters (such as dot11RSNAStatsCCMPDecryptErrors) to continuously increment. (See 8.7.2.3 for a description of the procedure for receiving encrypted frames.)

The mesh GTKSA contains the following:

- MGTK
- MGTK source mesh STA MAC address (mesh STA that uses this GTK to encrypt transmissions)
- Group Cipher Suite Selector
- Lifetime
- Direction vector (whether this is a receive mesh GTKSA or transmit mesh GTKSA)
- Key Index

8.4.1.2 Security association life cycle

8.4.1.2.1 Security association in an ESS

Change the list item b) in 8.4.1.2.1 as follows:

b) The STA then ~~uses~~ performs IEEE 802.11 ~~Open System~~ authentication followed by association to the chosen AP. ~~Negotiation~~ Confirmation of security parameters takes place during association. A STA performing IEEE 802.1X authentication uses Open System authentication. A STA performing secure password-based, or PSK, authentication uses SAE authentication.

Change the list item c) in 8.4.1.2.1 as follows:

c) SAE authentication provides mutual authentication and derivation of a PMK. If Open System authentication is chosen instead, ~~T~~the Authenticator or the Supplicant initiates IEEE 802.1X authentication. The EAP method used by IEEE Std 802.1X-2004 will support mutual authentication, as the STA needs assurance that the AP is a legitimate AP.

Change the list item d) in 8.4.1.2.1 as follows:

d) The last step is key management. The authentication process, whether SAE authentication utilizing IEEE 802.11 authentication frames or IEEE 802.1X authentication utilizing data frames post association, creates cryptographic keys shared between the cryptographic endpoints—the AP and STA, or the IEEE 802.1X AS and the STA, when using SAE or IEEE 802.1X, respectively. When using IEEE 802.1X ~~T~~the AS transfers these keys to the AP, and the AP and STA use one of the key confirmation handshakes, e.g., the 4-Way Handshake or FT 4-Way Handshake, to complete security association establishment. When using SAE authentication there is no AS and therefore no key transfer; the 4-way Handshake is performed directly between the AP and STA. The key confirmation handshake indicates when the link has been secured by the keys and is ready to allow normal data traffic and protected Robust Management frames.

Change the third paragraph in 8.4.1.2.1 as follows:

When FT is not enabled, a STA roaming within an ESS establishes a new PMKSA by one of the ~~three~~ four schemes:

Insert the following new dashed list item after the first dashed list item after the third paragraph in 8.4.1.2.1:

— In the case of SAE authentication followed by (re)association, the STA repeats the same actions as for initial contact association, but the non-AP STA also deletes the PTKSA when it roams from the old AP. Note that a STA can take advantage of the fact that it can perform SAE authentication to multiple APs while maintaining a single association with one AP, and then use any of the PMKSAs created during authentication to effect a fast BSS transition.

Change the third dashed list item after the third paragraph in 8.4.1.2.1 as follows:

— A STA (AP) can retain PMKs for APs (STAs) in the ESS to which it has previously performed a full IEEE 802.1X authentication or SAE authentication. If a STA wishes to roam to an AP for which it has cached one or more PMKSAs, it can include one or more PMKIDs in the RSN element of its (Re)Association Request frame. An AP ~~whose Authenticator~~ that has retained a PMK for one or more of the PMKIDs can ~~skip the IEEE 802.1X authentication and~~ proceed directly with the 4-Way Handshake. If none of the PMKIDs of the cached PMKSAs matches any of the supplied PMKIDs, or if the AKM of the cached PMKSA differs from that offered in the (Re)Association Request, or the PMK in the cached PMKSA is no longer valid, then, in the case of Open System authentication, the Authenticator shall perform another IEEE 802.1X authentication, and in the case of SAE authentication shall transmit a Deauthentication frame to the STA. Similarly, if the STA fails to send a PMKID, the STA and AP ~~must~~ shall perform a full IEEE 802.1X authentication.

8.4.1.2.2 Security association in an IBSS

Change the first three paragraphs in 8.4.1.2.2 as follows:

In an IBSS utilizing IEEE 802.11 Open System authentication and IEEE 802.1X, when a STA's SME establishes a security association with a peer STA, it creates both an IEEE 802.1X Supplicant and Authenticator for the peer. A STA in such an IBSS can also receive IEEE 802.1X messages from a previously unknown MAC address.

~~A STA can receive IEEE 802.1X messages from a previously unknown MAC address.~~

In an IBSS utilizing IEEE 802.11 SAE authentication, a STA creates a security association for a peer upon successful SAE authentication.

Any STA within an IBSS may decline to form a security association with a STA joining the IBSS. An attempt to form a security association may also fail because, for example, the peer uses a different PSK or password from what the STA expects.

8.4.6 RSNA authentication in an ESS

Change the first paragraph in 8.4.6 as follows:

When establishing an RSNA in a non-FT environment or during an FT initial mobility domain association, a STA shall use IEEE 802.11 SAE authentication or Open System authentication prior to (re)association.

Insert the following new paragraph after the first paragraph in 8.4.6:

SAE authentication is initiated when a STA's MLME-SCAN.confirm primitive finds another AP within the current ESS that advertises support for SAE in its RSN information element.

8.4.6.2 Cached PMKSAs and RSNA key management

Change the second paragraph in 8.4.6.2 as follows:

If a STA in an ESS has determined it has a valid PMKSA with an AP to which it is about to (re)associate, it includes the PMKID for the PMKSA in the RSN element in the (Re)Association Request. Upon receipt of a (Re)Association Request with one or more PMKIDs, an AP checks whether its Authenticator has retained a PMK for the PMKIDs, whether the AKM in the cached PMKSA matches the AKM in the (Re)Association Request, and whether the PMK is still valid~~. If~~ and if so, it shall assert possession of that PMK by beginning the 4-Way Handshake after association has completed~~; otherwise it shall begin a full IEEE 802.1X authentication after association has completed~~. If the Authenticator does not have a PMK for the PMKIDs in the (Re)Association Request, its behavior depends on how the STA performed IEEE 802.11 authentication. If the STA performed SAE authentication, then the AP STA shall send a Deauthentication frame. If the STA performed Open System authentication, it begins a full IEEE 802.1X authentication after association has completed.

8.4.7 RSNA authentication in an IBSS

Change the eighth and ninth paragraphs in 8.4.7 as shown:

Password or PSK authentication may also be used in an IBSS. When a single password or PSK is shared among the IBSS STAs, ~~the~~ an SAE capable STA wishing to establish communication with a STA that advertises support for SAE in Beacon and Probe Response frames invokes SAE authentication, and upon successful conclusion of SAE, sends a 4-Way Handshake Message 1 to the target STA~~(s)~~. If the STA does not support SAE authentication or the target STA does not advertise support for SAE in Beacon and Probe Response frames, the STA may use the PSK as a PMK and initiate the 4-Way Handshake by sending a 4-Way Handshake Message 1 to the target STA. In either case, t~~T~~he targeted STA responds to Message 1 with Message 2 of the 4-Way Handshake and begins its 4-Way Handshake by sending Message 1 to the initiating STA. The two 4-Way Handshakes establish PTKSAs and GTKSAs to be used between the initiating STA and the targeted STA. PSK PMKIDs have security vulnerabilities when used with low-entropy keys and should be used only after taking this into account. ~~PSK PMKIDs may also be used, enabling support for pairwise PSKs.~~

The model for security in an IBSS is not general. In particular, it assumes the following:

a) The sets of use cases for which the authentication procedures described in this subclause are valid are as follows:

1) Password or PSK-based authentication using SAE to perform mutual authentication and generation of a shared PMK.
2) An alternate form of PSK-based authentication, typically managed by the pass-phrase hash method as described in H.4 (Suggested pass-phrase-to-PSK mapping). This method has security vulnerabilities and should only be used when SAE authentication is not possible.
3) EAP-based authentication, using credentials that have been issued and preinstalled on the STAs within a common administrative domain, such as a single organization

b) All of the STAs are in direct radio communication. In particular, there is no routing, bridging, or forwarding of traffic by a third STA to effect communication. This assumption is made, because the model makes no provision to protect IBSS topology information from tampering by one of the members.

8.4.8 RSNA key management in an ESS

Insert the following new paragraph after the first paragraph in 8.4.8:

When SAE authentication completes, both STAs share a PMK. With this PMK in place, the AP initiates the key confirmation handshake with the STA.

8.5 Keys and key distribution

8.5.1 Key hierarchy

8.5.1.5 FT key hierarchy

8.5.1.5.1 Overview

Change the first two paragraphs in 8.5.1.5.1 as follows:

This subclause describes the FT key hierarchy and its supporting architecture. The FT key hierarchy is designed to allow a STA to make fast BSS transitions between APs without the need to perform an SAE or IEEE 802.1X authentication at every AP within the mobility domain.

The FT key hierarchy can be used with ~~either~~ SAE, IEEE 802.1X authentication, or PSK authentication.

Replace Figure 8-22a with the following figure:

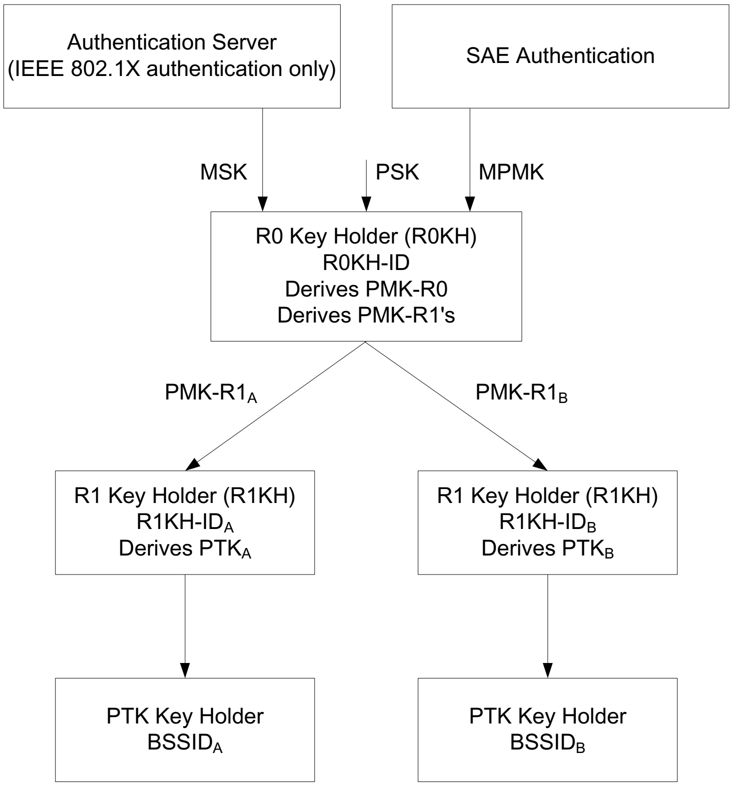


Figure 8-22a—FT key hierarchy at an Authenticator

Change the first sentence of the fifth paragraphs in 8.5.1.5.1 as follows:

As shown in Figure 8-22a, the R0KH computes the PMK-R0 ~~either~~ from the key obtained from SAE authentication (for the purposes of FT this key is identified as the Master PMK, or MPMK), the PSK or from the MSK resulting (per IETF RFC 3748-2004 [B26]) from a successful IEEE 802.1X authentication between the AS and the Supplicant.

Change the first sentence of the seventh paragraphs in 8.5.1.5.1 as follows:

The lifetime of the PMK-R0, PMK-R1, and PTK are bound to the lifetime of the MPMK, PSK, or MSK from which it was derived.

8.5.1.5.3 PMK-R0

Change the third dashed list item in 8.5.1.5.3 as follows:

— If the AKM negotiated is 00-0F-AC:3 then XXKey shall be the second 256 bits of the MSK (which is derived from the IEEE 802.1X authentication), i.e., XXKey = L(MSK, 256, 256). If the AKM negotiated is 00-0F-AC:4, then XXKey shall be the PSK. If the AKM negotiated is 00-0F-AC:9, then XXKey shall be the MPMK generated as the result of SAE authentication.

8.5.3 4-Way Handshake

8.5.3.1 4-Way Handshake Message 1

Change the first sentence of the third paragraph in 8.5.3.1 as follows:

The Authenticator sends Message 1 to the Supplicant at the end of a successful IEEE 802.1X authentication, after (re)association completes for a STA that has authenticated with SAE or PSK authentication is negotiated, when a cached PMKSA is used, or after a STA requests a new key.

8.5.6 RSNA Authenticator key management state machine

8.5.6.1 Authenticator state machine states

8.5.6.1.1 Authenticator state machine: 4-Way Handshake (per STA)

Change the pseudo-code in Figure 8-38 as follows:

```
Keycount = 0
If GUpdateStationKeys == TRUE
        GKeyDoneStation—
GUpdateStationKeys = FALSE
If Unicast cipher supported by Authenticator AND (ESS OR ((IBSS or ~~WDS~~ (FromDS==1 AND ToDS == 1)) and Local AA > Remote AA)))
        Pair = TRUE
IEEE 802.1X::portEnable = FALSE
MLME-DeleteKeys.Request(PTK)
IEEE 802.1X::portValid = FALSE
TimeoutCtr = 0
```

Insert the following new subclause after 8.7:

8.8 Keys and key derivation algorithm for the authenticated mesh peering exchange (AMPE)

8.8.1 Keys and key derivation algorithm

To execute the authenticated mesh peering exchange (AMPE), and mesh group key handshake with a candidate peer mesh STA, the mesh STA shall derive an authenticated encryption key (AEK) and a mesh temporal key (MTK) using the PMK it shares with the candidate peer mesh STA.

The AEK is derived statically from the shared PMK. The MTK is derived from the shared PMK and dynamic information provided by the mesh STA and candidate peer mesh STA.

The AEK is mutually derived by the local mesh STA and the peer mesh STA once a new PMK has been selected. The AEK shall be derived from the PMK by

AEK ← KDF-256(PMK, “AEK Derivation”, Selected AKM Suite ||
min(localMAC, peerMAC) || max(localMAC, peerMAC)).

The temporal key (MTK) shall be derived from the PMK by

MTK ← KDF-X(PMK, "Temporal Key Derivation", min(localNonce, peerNonce) ||
max(localNonce, peerNonce) || min(localLinkID, peerLinkID) ||
max(localLinkID, peerLinkID) || Selected AKM Suite ||
min(localMAC, peerMAC) || max(localMAC, peerMAC)).

CCMP uses X = 128. The "min" and "max" operations for IEEE 802 addresses are with the address converted to a positive integer, treating the first transmitted octet as the most significant octet of the integer as specified in 8.5.1.2. The min and max operations for nonces are with the nonces treated as positive integers converted as specified in 7.1.1.

The MTK is used to protect communications between two peer mesh STAs. The local mesh STA and peer mesh STA derive an MTK per peering instance and may rekey the MTK using AMPE.

9. MAC sublayer functional description

Change the first paragraph of Clause 9 as follows:

The MAC functional description is presented in this clause. The architecture of the MAC sublayer, including the distributed coordination function (DCF), the point coordination function (PCF), the hybrid coordination function (HCF), the mesh coordination function (MCF), and their coexistence in an IEEE 802.11 LAN are introduced in 9.1 (MAC architecture). These functions are expanded on in 9.2 (DCF), 9.3 (PCF), ~~and~~ 9.9 (HCF), and 9.9a (MCF). Fragmentation and defragmentation are defined in 9.4 (Fragmentation) and 9.5 (Defragmentation). Multirate support is addressed in 9.6 (Multirate support). A number of additional restrictions to limit the cases in which MSDUs are reordered or discarded are described in 9.7 (MSDU transmission restrictions). Operation across regulatory domains is defined in 9.8 (Operation across regulatory domains). The Block Ack mechanism is described in 9.10 (Block Acknowledgment (Block Ack)). The No Ack mechanism is described in 9.11 (No Acknowledgment (No Ack)). The protection mechanism is described in 9.13 (Protection mechanisms). Rules for processing MAC frames are described in 9.14 (MAC frame processing).

9.1 MAC architecture

Insert the following paragraph to the end of 9.1:

Due to the distributed nature of the MBSS, only the MCF is present in a mesh STA.

Replace Figure 9-1 with the following figure:

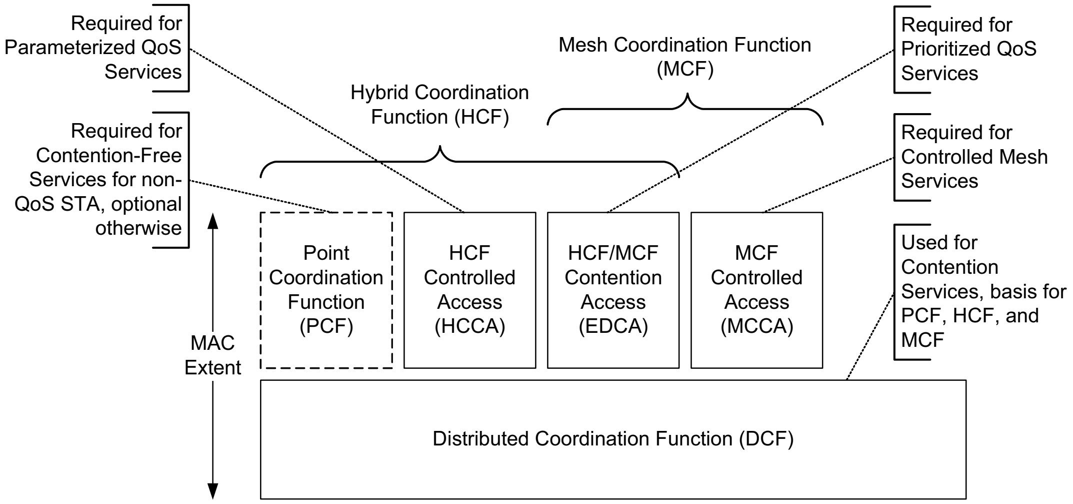

Required for Parameterized QoS Services
Required for Contention-Free Services for non-QoS STA, optional otherwise
Mesh Coordination Function (MCF)
Hybrid Coordination Function (HCF)
Required for Prioritized QoS Services
Required for Controlled Mesh Services
Used for Contention Services, basis for PCF, HCF, and MCF
MAC Extent
Point Coordination Function (PCF)
HCF Controlled Access (HCCA)
HCF/MCF Contention Access (EDCA)
MCF Controlled Access (MCCA)
Distributed Coordination Function (DCF)

Figure 9-1—MAC architecture

9.1.3 Hybrid coordination function (HCF)

Change the first paragraph in 9.1.3 as follows:

The QoS facility includes an additional coordination function called *HCF* that is only usable in QoS network configurations. The HCF shall be implemented in all QoS STAs except mesh STAs. Instead, mesh STAs implement the MCF. The HCF combines functions from the DCF and PCF with some enhanced, QoS-specific mechanisms and frame subtypes to allow a uniform set of frame exchange sequences to be used for QoS data transfers during both the CP and CFP. The HCF uses both a contention-based channel access method, called the *enhanced distributed channel access* (EDCA) mechanism for contention-based transfer and a controlled channel access, referred to as the *HCF controlled channel access* (HCCA) mechanism, for contention-free transfer.

9.1.3.1 HCF contention-based channel access (EDCA)

Change the second paragraph in 9.1.3.1 as follows:

For each AC, an enhanced variant of the DCF, called an *enhanced distributed channel access function* (EDCAF), contends for TXOPs using a set of EDCA parameters. When communicating data frames outside the context of a BSS (dot11OCBEnabled is true), the EDCA parameters are the corresponding default values or are as set by the SME in the MIB attribute table dot11EDCATable (except for TXOP limit values, which shall be set to zero for each AC). When communicating within a BSS, the EDCA parameters used are from the EDCA Parameter Set element, or from the default values for the parameters when no EDCA Parameter Set element is received from the AP of the BSS with which the STA is associated or when the STA is a mesh STA. The parameters used by the EDCAF to control its operation are defined by MIB attribute table dot11QAPEDCATable at the AP and by MIB attribute table dot11EDCATable at the non-AP STA.

Change the second to last paragraph in 9.1.3.1 as follows:

Management frames shall be sent using the access category AC_VO without being restricted by admission control procedures. A QoS STA shall also send management frames using the access category AC_VO before associating with any BSS and before establishing mesh peerings in an MBSS, even if there is no QoS facility available in that BSS. BlockAckReq and BlockAck control frames shall be sent using the same QoS parameters as the corresponding QoS data frames. PS-Poll control frames shall be sent using the access category AC_BE to reduce the likelihood of collision following a Beacon frame. When the first frame in a frame exchange sequence is an RTS or CTS, the RTS or CTS frame shall inherit the UP of the data frame(s) or the AC of the management frame(s) that are included in the frame exchange sequence.

Insert the following new subclause immediately after 9.1.3:

9.1.3a Mesh coordination function (MCF)

The mesh facility includes an additional coordination function called MCF that is usable only in an MBSS. Mesh STAs shall implement the MCF only. MCF has both a contention-based channel access and contention free channel access mechanism. The contention based mechanism is EDCA and the contention free mechanism is called the MCF controlled channel access (MCCA). MCF uses the default values for the PTKSA, GTKSA and STKSA Replay Counters. The operation rules of the EDCA are defined in 9.9.1. The operation rules of the MCCA are defined in 9.9a.3.

9.6 Multirate support

Insert the following new subclause after the 9.6.0c:

9.6.0c1 Basic Rate Set and Basic MCS Set for mesh STA

A mesh STA shall not establish a mesh peering with a mesh STA using a different BSSBasicRateSet (see 11C.2.7 and 11C.2.8).

Mesh STAs should adopt the mandatory PHY rates as the default BSSBasicRateSet to reduce the risk that a candidate peer mesh STA utilizes a different BSSBasicRateSet. If the mesh STA is also an HT STA, it should adopt the MCSs of mandatory MCSs as the default BSSBasicMCSSet.

Once the mesh STA establishes a mesh peering with a mesh STA, it shall change neither the BSSBasicRateSet nor the BSSBasicMCSSet parameters.

9.7d A-MPDU operation

Change the paragraphs of 9.7.d.4 as follows:

9.7d.4 A-MPDU aggregation of group addressed data frames

An HT STA that is neither an AP nor a mesh STA ~~A non-AP HT STA~~ shall not transmit an A-MPDU containing an MPDU with a group-addressed RA.

NOTE—An HT AP and an HT mesh STA can transmit an A-MPDU containing MPDUs with a group-addressed RA.

An HT AP and an HT mesh STA shall not transmit an A-MPDU containing group-addressed MPDUs if the HT Protection field is set to non-HT mixed mode.

When an HT AP or an HT mesh STA transmits an A-MPDU containing MPDUs with a group-addressed RA, both of the following shall apply:

— The value of maximum A-MPDU length exponent that applies is the minimum value in the Maximum A-MPDU Length Exponent subfield of the A-MPDU Parameters field of the HT Capabilities element across all HT STAs associated with the AP or all peer HT mesh STAs.

— The value of minimum MPDU start spacing that applies is the maximum value in the Minimum MPDU Start Spacing subfield of the A-MPDU Parameters field of the HT Capabilities element across all HT STAs associated with the AP or all peer HT mesh STAs.

9.9 HCF

9.9.1 HCF contention-based channel access (EDCA)

9.9.1.2 EDCA TXOPs

Change the sixth paragraph in 9.9.1.2 as follows:

It should be noted, that when transmitting multiple frames in a TXOP using acknowledgment mechanisms other than Normal Ack, a protective mechanism should be used (such as RTS/CTS or the protection mechanism described in 9.13). A QoS AP or a mesh STA may send group addressed frames without using any protection mechanism. In a QoS IBSS, group addressed frames shall be sent one at a time, and backoff shall be performed after the transmission of each of the group addressed frames. In an MBSS, a mesh STA may send multiple group addressed frames in a TXOP, bounded by the TXOP limit, without performing backoff after the TXOP is obtained.

Insert the following new subclauses after 9.9:

9.9a Mesh coordination function (MCF)

9.9a.1 General

Under MCF, the basic unit of allocation of the right to transmit onto the WM is the TXOP. Each TXOP is defined by a starting time and a defined maximum length.

There are two types of TXOP in MCF: EDCA TXOPs and MCCA TXOPs. The EDCA TXOP is obtained by a mesh STA winning an instance of EDCA contention (see 9.9.1). The MCCA TXOP is obtained by a mesh STA gaining control of the WM during an MCCAOP. The MCCAOP is an interval of time for frame transmissions that has been reserved by means of the exchange of MCCA frames (see 9.9a.3). Neither EDCA TXOPs nor MCCA TXOPs shall exceed dot11MaxDwellTime (if using an FH PHY).

EDCA TXOPs of a mesh STA that has dot11MCCAActivated true shall not overlap with the time periods of any of its tracked MCCAOP reservations.

The process of tracking MCCAOP reservations involves the recording of the MCCAOP reservations and the data that structure the MCCAOP advertisements of these reservations, namely the advertisement set sequence number, advertisement elements bitmap, and the advertisement element indexes, in a local database, and the updating of this database on the basis of received advertisements as described in 9.9a.3.7.5.

9.9a.2 MCF contention-based channel access

MCF implements the same EDCA, see 9.9.1, as does HCF.

9.9a.3 MCF controlled channel access (MCCA)

9.9a.3.1 General

MCF controlled channel access (MCCA) is an optional access method that allows mesh STAs to access the WM at selected times with lower contention than would otherwise be possible. This standard does not require all mesh STAs to use MCCA. MCCA might be used by a subset of mesh STAs in an MBSS. However, MCCAOP reservations shall only be set up among mesh STAs that have dot11MCCAActivated true and that operate on the same channel. The performance of MCCA might be impacted by STAs that do not respect MCCAOP reservations.

MCCA enabled mesh STAs use management frames to make reservations for transmissions. The mesh STA transmitting an MCCA Setup Request frame to initiate a reservation becomes the MCCAOP owner of the MCCAOP reservation. The receivers of the MCCA Setup Request frame are the MCCAOP responders. The MCCAOP owner and the MCCAOP responders advertise this MCCAOP reservation to their neighbors via an MCCAOP advertisement. The MCCA enabled neighbor mesh STAs that could cause interference to transmissions during these reserved time periods, or that would experience interference from them, shall not initiate a transmission during these reserved time periods. During its MCCAOP, the MCCAOP owner obtains a TXOP by winning an instance of EDCA contention. Because of its reservation, the MCCAOP owner experiences no competition from other MCCA enabled neighbor mesh STAs. At the start of an MCCAOP, the EDCAF of the MCCAOP owner replaces the AIFSN, CWmin, and CWmax value of its dot11EDCATable with MCCA access parameters.

In order to use MCCA, a mesh STA maintains synchronization with its neighboring mesh STAs. Mesh STAs that use MCCA shall use a DTIM interval with a duration of $2^n \times 100$ TU with n being a non-negative integer less than or equal to 17. Additionally, a mesh STA shall track the reservations of its neighboring mesh STAs.

NOTE 1—The DTIM interval of this form was chosen so that the starting times of the reservations do not change relative to each other between consecutive DTIM intervals. The restriction that *n* be less than or equal to 17 was chosen for compatibility with the maximum DTIM interval as well as the compatibility of the reservation's MCCAOP offset range (see 7.3.2.106.2) with the maximal DTIM interval length.

NOTE 2—It is allowed that a different value for the DTIM interval is used for mesh STAs that use MCCA in an MBSS that is centrally controlled and the central authority provides a coordination of the DTIM interval of the mesh STAs that use MCCA in the MBSS.

9.9a.3.2 MCCA activation

When it receives an MLME-ACTIVATEMCCA.request primitive from its SME, a mesh STA shall set the MCCA Enabled subfield of the Mesh Capability field in the Mesh Configuration element to 1 in Beacon and Probe Response frames it transmits. It shall not initiate or accept MCCA Setup Request frames for dot11MCCAScanDuration TUs after the receipt of the MLME-ACTIVATEMCCA.request primitive.

During the dot11MCCAScanDuration waiting period, the mesh STA learns its neighborhood MCCAOP periods by receiving Beacon, Probe Response, or MCCA Advertisement frames from neighboring mesh STAs.

After this period, the mesh STA may initiate and accept MCCA Setup Request frames as per 9.9a.3.6.

9.9a.3.3 MCCAOP reservations

An MCCAOP reservation specifies a schedule for frame transmissions. The time periods scheduled for frame transmissions in the reservation are called MCCAOPs. The schedule is set up between an MCCAOP owner and one (for individually addressed frames) or more (for group addressed frames) MCCAOP responders. MCCAOPs are set up by means of the procedure defined in 9.9a.3.6. Once an MCCAOP reservation is set:

— Access to the channel by MCCA enabled mesh STAs is governed by the procedures in 9.9a.3.9.

— The MCCAOP reservation is advertised according to the procedures in 9.9a.3.7.

The schedule is defined by means of the MCCAOP Reservation field defined in 7.3.2.106.2. An MCCAOP reservation schedules a series of MCCAOPs with a common duration given in the MCCAOP Duration subfield of the MCCAOP Reservation field. This series is started after the first DTIM Beacon following the successful completion of the MCCAOP setup procedure and terminated when the MCCAOP reservation is torn down.

The reservation defines a regular schedule of MCCAOPs in the DTIM interval of the MCCAOP owner. The number of MCCAOPs in the DTIM interval is given by the value of the MCCAOP Periodicity subfield of the MCCAOP Reservation field. The MCCAOP Offset subfield specifies the offset of the first scheduled MCCAOP of the transmission schedule relative to the beginning of the DTIM interval of the MCCAOP owner. The following MCCAOPs are separated by a time interval with a duration equal to the length of the DTIM period divided by the value in the MCCAOP Periodicity subfield.

An example of an MCCAOP reservation schedule is shown in Figure 9-20a. In this example, the MCCAOP Periodicity equals two, so that there are two MCCAOPs in each DTIM interval. As further illustrated in the figure, the MCCAOP Offset value indicates the beginning of the first MCCAOP in each DTIM interval.

If a mesh STA adjusts its TBTT, e.g., in response to a TBTT adjustment request, it shall adjust the MCCAOP reservations by modifying the MCCAOP Offset of each MCCAOP reservation.

An MCCAOP reservation is identified by an MCCAOP reservation ID. The MCCAOP owner shall select an MCCAOP reservation ID that is unique among all of its MCCAOP reservations. The MCCAOP reservation ID and MAC address of the MCCAOP owner uniquely identify the MCCAOP reservation in the mesh BSS.

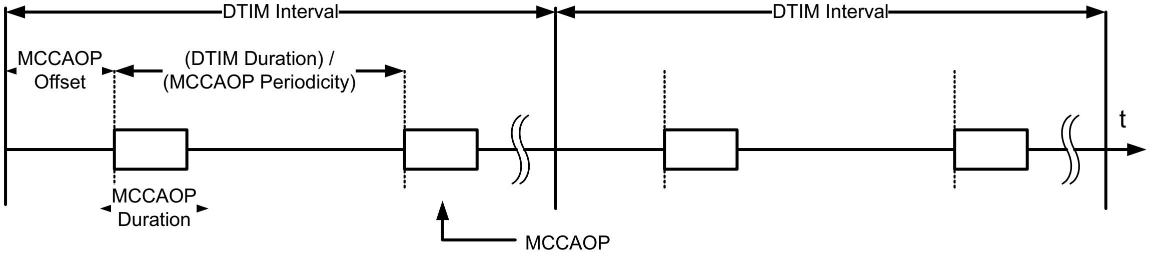


Figure 9-20a—Example MCCAOP reservation with MCCAOP Periodicity equal to 2

The MCCAOP reservation ID is an 8-bit unsigned integer and included in the MCCAOP Reservation ID field of an MCCAOP Setup Request element. If this MCCAOP setup request is for an individually addressed transmission, the MCCAOP Reservation ID is between 0 and 127. If this MCCAOP setup request is for a group addressed transmission, the MCCAOP Reservation ID is between 128 and 254. The value 255 is not used to identify a specific MCCAOP reservation but is reserved for usage in the MCCAOP teardown procedure as described in 9.9a.3.8.

A mesh STA with dot11MCCAActivated equal to true shall be able to track at least dot11MCCAMinTrackStates MCCAOP reservations, including its own reservations. If the number of tracked MCCAOP reservations is less than dot11MCCAMaxTrackStates, the mesh STA shall be able to track, set up, and accept additional reservations. In this case, the mesh STA shall set the Accept Reservations subfield in the Flags field to 1 in the MCCAOP Advertisement Overview elements it transmits.

If the number of tracked MCCAOP reservations is equal to or greater than dot11MCCAMaxTrackStates, the mesh STA shall not track, set up, or accept additional reservations. In this case, the mesh STA shall set the Accept Reservations subfield in the Flags field to 0 in the MCCAOP Advertisement Overview elements it transmits. Moreover, it shall reply to MCCA Setup Request frames with an MCCA Setup Reply frame with the MCCA Reply Code field in the MCCAOP Setup Reply element equal to 3: Reject: MCCAOP track limit exceeded.

The tracked MCCAOP reservations are advertised as described in 9.9a.3.7. How to access the medium during the tracked MCCAOP reservations is specified in 9.9a.3.9.

9.9a.3.4 Neighborhood MCCAOP periods at a mesh STA

The set of MCCAOP reservations in which a mesh STA is involved as an MCCAOP owner or an MCCAOP responder and that are used for individually addressed transmissions are referred to as the TX-RX periods of this mesh STA.

The set of MCCAOP reservations in which a mesh STA is involved as an MCCAOP owner or an MCCAOP responder and that are used for group addressed transmissions are referred to as the broadcast periods of this mesh STA. Optionally, the broadcast periods of a mesh STA includes known Target Beacon Transmission Time of Beacon frames for which this mesh STA is either the transmitter or the receiver, and transmission or reception periods of a STA that is collocated with the reporting mesh STA, for example, beacon or HCCA times of a collocated AP.

The interference periods of a mesh STA comprise the TX-RX periods and the broadcast periods of its neighbor mesh STAs in which the mesh STA is not involved as the owner or as a responder. The TX-RX periods, the broadcast periods, and the interference periods of a mesh STA shall not be used for a new

MCCAOP reservation with the mesh STA as transmissions in these periods may experience interference from the transmissions in the new MCCAOPs or may cause interference to them.

The interference periods are directly derived from the TX-RX Periods Report field and Broadcast Periods Report field of the MCCAOP Advertisement elements transmitted by the neighbor mesh STAs. The Interference Periods Report reflects the latest TX-RX Periods Reports and Broadcast Periods Reports received from the neighbor mesh STAs.

The MCCAOP reservations of a mesh STA and its neighbors define a set of MCCAOPs that are already reserved for frame transmissions in the mesh neighborhood of a mesh STA. This set of MCCAOPs is referred to as the neighborhood MCCAOP periods for the mesh STA. Thus, neighborhood MCCAOP periods at a mesh STA include all MCCAOPs for which the mesh STA or one of its neighbors, including neighbors from other MBSSs, is either transmitter or receiver.

9.9a.3.5 MCCA access fraction (MAF)

The MCCA access fraction at a mesh STA is the ratio of the time reserved for MCCAOPs in the DTIM interval of this mesh STA to the duration of the DTIM interval. This parameter is reported in the MCCA Access Fraction field of the MCCAOP Advertisement Overview elements. The maximum value for the MAF that is allowed at a mesh STA is specified by dot11MAFlimit. The dot11MAFlimit is copied into the MAF Limit field of the MCCAOP Advertisement Overview element as described in 7.3.2.108.

The MAF and the MAF Limit may be used to limit the use of MCCA in the mesh neighborhood of a mesh STA, as specified in 9.9a.3.6. Before attempting to set up an MCCAOP reservation with a neighbor peer mesh STA, a mesh STA shall verify that the new MCCAOP reservation does not cause its MAF to exceed its MAF Limit and that the new MCCAOP reservation does not cause the MAF of any of its neighbor peer mesh STAs to exceed their MAF Limit. An MCCAOP setup request shall be refused by the intended MCCAOP responder if the MAF limit of one of its neighbors is exceeded due to the new setup.

9.9a.3.6 MCCAOP setup procedure

The setup of an MCCAOP reservation is initiated by the MCCAOP owner, and is accepted or rejected by the MCCAOP responder. The setup procedure for an MCCAOP reservation is as follows:

a) The MCCAOP owner shall build a map of the neighborhood MCCAOP periods in the DTIM interval after hearing advertisements from all of its neighbor mesh STAs with the MCCA Enabled subfield of the Mesh Capability field in the Mesh Configuration element equal to 1. It shall request an MCCAOP advertisement, as described in 9.9a.3.7.8, from each neighbor mesh STA from which no advertisement was heard in the last dot11MCCAAdvertPeriodMax DTIM intervals.

b) The MCCAOP owner shall determine the MCCAOP reservation. The MCCAOP parameters shall be chosen in such a way that they satisfy the following conditions:

   1) The reservation shall not overlap with the neighborhood MCCAOP periods of the MCCAOP owner.

   2) The reservation shall not overlap with the interference periods of the intended MCCAOP responder or responders.

   3) The reservation shall not cause the MAF limit to be exceeded for either itself or its neighbor mesh STAs.

   4) The Accept Reservations subfield of the Flags field equals 1 in the most recent MCCAOP Advertisement Overview element received from all intended MCCAOP responders.

c) If the conditions in item b) are satisfied, the MCCAOP owner shall transmit an MCCAOP Setup Request element to the intended MCCAOP responder with the chosen MCCAOP parameters.

d) The MCCAOP responder shall verify the following conditions:

   1) The reservation does not overlap with its neighborhood MCCAOP periods.

   2) The reservation does not cause the MAF limit to be exceeded for itself or its neighbor mesh STAs.

   3) The number of reservations in its neighborhood MCCAOP periods does not exceed dot11MCCAMaxTrackStates.

e) If the conditions in item d) are satisfied, the responder shall send an MCCA Setup Reply frame to the MCCAOP owner with the MCCA Reply Code field in the MCCAOP Setup Reply element equal to 0: Accept, as defined in Table 7-43bj7.

f) If the conditions in item d) are satisfied and the MCCAOP request has been intended for group addressed transmissions, the responder shall include the reservation in its MCCAOP advertisement only after the MCCAOP advertisement from the MCCAOP owner is received.

g) If not all of the conditions in item d) are satisfied and the MCCAOP request is intended for individually addressed transmissions, the responder shall transmit to the MCCAOP owner an MCCA Setup Reply frame that is constructed as follows:

   1) If the condition in item d)1) is not satisfied and both conditions in item d)2) and item d)3) are satisfied, the responder may calculate an alternative MCCAOP reservation and include it in the MCCAOP Reservation field of the MCCAOP Setup Reply element. It shall set the MCCA Reply Code field of the MCCAOP Setup Reply element to 1: Reject: MCCAOP reservation conflict, as defined in Table 7-43bj7.

   2) If the condition in item d)2) is not satisfied, it shall set the MCCA Reply Code field of the MCCAOP Setup Reply element to 2: Reject: MAF limit exceeded, as defined in Table 7-43bj7.

   3) If the condition in item d)2) is satisfied and the condition in item d)3) is not satisfied, it shall set the MCCA Reply Code field of the MCCAOP Setup Reply element to 3: Reject: MCCAOP track limit exceeded, as defined in Table 7-43bj7.

h) If not all of the conditions in item d) are satisfied and the MCCAOP request is intended for group addressed transmissions, the responder shall send an MCCA Setup Reply frame to the MCCAOP owner with the MCCA Reply Code field in the MCCAOP Setup Reply element equal to 1: Reject: MCCAOP reservation conflict.

i) If the MCCAOP owner receives an MCCA Setup Reply frame with MCCA Reply Code equal to Accept, the MCCAOP reservation is established. Otherwise, the mesh STA may repeat the MCCAOP setup procedure using a modified MCCAOP Setup Request. If an alternative MCCAOP reservation is included in the MCCAOP Setup Reply element, the mesh STA may consider this alternative in its modified MCCAOP Setup Request.

9.9a.3.7 MCCAOP advertisement

9.9a.3.7.1 General

A mesh STA with dot11MCCAActivated equal to true tracks MCCAOP reservations. The tracked MCCAOP reservations contain the neighborhood MCCAOP periods and optionally other periodic transmission of itself or of neighboring STAs.

The MCCAOP advertisement set contains all MCCAOP reservations tracked by the mesh STA. The MCCAOP advertisement set is represented by an MCCAOP Advertisement Overview element and zero (if the MCCAOP advertisement set is empty) or more (if the MCCAOP advertisement set is non-empty) MCCAOP Advertisement elements. An MCCAOP Advertisement element contains one or more tracked MCCAOP reservations.

The mesh STA advertises its MCCAOP advertisement set to its neighbor mesh STAs.

This subclause describes how the mesh STA constructs the MCCAOP Advertisement Overview element and the MCCAOP Advertisement elements. Further, this subclause describes the procedure to advertise an MCCAOP advertisement set, the procedure to request an MCCAOP advertisement from a neighboring mesh STA, and the procedure to process a received MCCAOP advertisement.

9.9a.3.7.2 Construction of an MCCAOP advertisement set

Each MCCAOP reservation tracked by a mesh STA is one of the following types:

a) MCCAOP TX-RX period:
   1) An MCCAOP reservation for individually addressed frames for which the mesh STA is the MCCAOP owner or the MCCAOP responder.

b) MCCAOP broadcast period:
   1) An MCCAOP reservation for group addressed frames for which the mesh STA is the MCCAOP owner or the MCCAOP responder.
   2) Optionally, a known Target Beacon Transmission Time of Beacon frames for which the mesh STA is either the transmitter or the receiver.
   3) Optionally, a transmission or reception period of a STA that is collocated with the mesh STA, for example, beacon or HCCA times of a collocated AP.

c) MCCAOP interference period:
   1) A TX-RX or a broadcast period reported by a neighbor peer mesh STAs of the mesh STA excluding those periods for which this mesh STA is either the MCCAOP owner or the MCCAOP responder.
   2) Optionally, a TX-RX or a broadcast period reported by neighbor non-peer mesh STAs of the mesh STA.

The MCCAOP reservations are grouped into the following sets:

- MCCAOP TX-RX advertisement set
- MCCAOP broadcast advertisement set
- MCCAOP interference advertisement set

These three sets constitute the MCCAOP advertisement set. The mesh STA uses the MCCAOP Overview element and MCCAOP Advertisement elements to advertise its MCCAOP advertisement set to its neighbor mesh STAs.

The mesh STA acts as follows to construct the MCCAOP Overview elements and the MCCAOP Advertisement elements:

d) If the MCCAOP advertisement set is non-empty, the mesh STA constructs one or more MCCAOP reports according to the format described in 7.3.2.109.3 as follows:
   1) If the MCCAOP TX-RX advertisement set is non-empty, the mesh STA constructs one or more TX-RX reports according to the format described in 7.3.2.109.3 such that each reservation in the MCCAOP TX-RX advertisement set occurs exactly in one TX-RX report.
   2) If the MCCAOP broadcast advertisement set is non-empty, the mesh STA constructs one or more broadcast reports according to the format described in 7.3.2.109.3 such that each reservation in the MCCAOP broadcast advertisement set occurs exactly in one broadcast report.
   3) If the MCCAOP interference advertisement set is non-empty, the mesh STA constructs one or more interfering reports according to the format described in 7.3.2.109.3 such that each reservation in the MCCAOP interference advertisement set occurs exactly in one interfering report.

e) If the MCCAOP advertisement set is non-empty, the mesh STA constructs one or more MCCAOP Advertisement elements as follows:

   1) The MCCAOP Advertisement Set Sequence Number field is set to the MCCAOP advertisement set sequence number as explained in 9.9a.3.7.3.

   2) The MCCAOP Advertisement Element Index subfield is set to an identifier that uniquely identifies the MCCAOP Advertisement element in the MCCAOP advertisement set.

   3) Each MCCAOP Advertisement element includes at least one of the TX-RX reports, broadcast reports, or interfering reports. Moreover, it includes at most one of the TX-RX reports, at most one of the broadcast reports, and at most one of the interfering reports. In case the MCCAOP Advertisement element contains a TX-RX report, the TX-RX Report Present subfield of the MCCAOP Advertisement Element Information field is set to 1, otherwise this subfield is set to 0. In case the MCCAOP Advertisement element contains a broadcast report, the Broadcast Report Present subfield of the MCCAOP Advertisement Element Information field is set to 1, otherwise this subfield is set to 0. In case the MCCAOP Advertisement element contains an interfering report, the Interference Report Present subfield of the MCCAOP Advertisement Element Information field is set to 1; otherwise, this subfield is set to 0.

   4) Each report as constructed in step d) is present in exactly one MCCAOP Advertisement element.

f) The mesh STA constructs one MCCAOP Advertisement Overview element such that

   1) The MCCAOP Advertisement Set Sequence Number field is set to the advertisement set sequence number as explained in 9.9a.3.7.3.

   2) The Medium Access Fraction field is set to the medium access fraction.

   3) The MAF limit field is set to the value of dot11MAFlimit.

   4) The Accept Reservations field is set to 1 if the number of tracked reservations of this mesh STA is less than dot11MCCAMaxTrackStates, and set to 0 otherwise.

   5) Bit i of the Advertisement Elements Bitmap field is set to 1 if an MCCAOP Advertisement element with the MCCAOP Advertisement Element Index subfield equal to i is part of the representation of this MCCAOP advertisement set, and set to 0 otherwise.

9.9a.3.7.3 Setting the MCCAOP advertisement set sequence number

The MCCAOP advertisement set sequence number identifies an MCCAOP advertisement set. Mesh STAs with dot11MCCAActivated equal to true assign MCCAOP advertisement set sequence numbers from a single modulo-256 counter. The MCCAOP advertisement set sequence number is initialized to 0. The MCCAOP advertisement set sequence number shall be incremented by 1 if one of the following conditions holds:

a) The mesh STA sets the bit for an MCCAOP Advertisement element in the Advertisement Elements Bitmap from 0 to 1 and this bit has been set to 1 under the same MCCAOP Advertisement Sequence Number before.

b) The bit of the Advertisement Elements Bitmap corresponding to an MCCAOP Advertisement element is equal to 1 and the content of this MCCAOP Advertisement element changes.

However, the MCCAOP advertisement set sequence number may remain unchanged if

c) The mesh STA changes a bit in the Advertisement Element Bitmap from 0 to 1 and this bit has not been set to 1 under the same MCCAOP Advertisement Sequence Number before, or

d) The mesh STA changes a bit in the Advertisement Elements Bitmap from 1 to 0.

NOTE—The Advertisement Set Sequence Number identifies the current distribution of the MCCAOP advertisement set over the MCCAOP Advertisement elements. Using a new MCCAOP advertisement set sequence number signals a new, (possibly) completely different distribution of the MCCAOP advertisement set over the MCCAOP Advertisement elements, and requires an advertisement of all reservations of the MCCAOP advertisement set. Leaving the MCCAOP advertisement set sequence number unchanged as in the previous MCCAOP Advertisement Overview element indicates

MCCAOP Advertisement elements that have previously been advertised are not changed and remain current. This enables a limited advertisement procedure in which only new MCCAOP Advertisement elements are advertised. Additionally, this enables mesh STAs that operate in light or deep sleep mode in requesting a limited update of the MCCAOP advertisement set of a neighboring mesh STA in which only new MCCAOP Advertisement elements are included.

9.9a.3.7.4 Advertisement procedure

To advertise its MCCAOP advertisement set, the mesh STA constructs a representation of the MCCAOP advertisement set as described in 9.9a.3.7.2. The MCCAOP advertisement set is advertised by transmitting an MCCAOP Advertisement Overview element and zero or more MCCAOP Advertisement elements (see 9.9a.3.7.2) to neighbor peer mesh STAs. The MCCAOP Advertisement Overview element and the MCCAOP Advertisement elements are transmitted in Beacon frames, Probe Response frames, or MCCA Advertisement frames.

The mesh STA shall advertise its MCCAOP advertisement set according to the following rules:

a) The mesh STA shall advertise at least one MCCAOP Advertisement Overview element in every dot11MCCAAdvertPeriodMax DTIM intervals.

b) The mesh STA shall advertise its MCCAOP Advertisement Overview element and any new MCCAOP Advertisement elements at the latest with the transmission of its next Beacon frame after its MCCAOP advertisement set has changed.

c) The mesh STA shall advertise the requested MCCAOP Advertisement elements as described in 9.9a.3.7.8 if the mesh STA receives an MCCA Advertisement Request frame.

9.9a.3.7.5 Receipt of an MCCAOP advertisement

Upon receipt of an MCCAOP advertisement a mesh STA with dot11MCCAActivated shall compare the Advertisement Set Sequence Number contained in the MCCAOP Advertisement Overview element of the received MCCAOP advertisement with the last advertisement set sequence number that this mesh STA tracked for the sender of the received MCCAOP advertisement.

If the tracked advertisement set sequence number does not equal the Advertisement Set Sequence Number of the received MCCAOP advertisement, the mesh STA shall perform the procedure described in 9.9a.3.7.6.

If the tracked advertisement set sequence number equals the Advertisement Set Sequence Number of the received MCCAOP advertisement, the mesh STA shall compare the Advertisement Elements Bitmap contained in the received MCCAOP Advertisement Overview element with the last Advertisement Elements Bitmap that this mesh STA tracked for the sender of the received MCCAOP advertisement. If the tracked Advertisement Elements Bitmap does not equal the Advertisement Elements Bitmap of the received MCCAOP advertisement, the mesh STA shall perform the procedure described in 9.9a.3.7.7.

NOTE—If both the tracked advertisement set sequence number equals the Advertisement Set Sequence Number of the received MCCAOP advertisement and the tracked Advertisement Elements Bitmap equals the Advertisement Elements Bitmap of the received MCCAOP advertisement, the MCCAOP advertisement set of the sender of the MCCAOP advertisement tracked by the mesh STA is current, and no update of the MCCAOP advertisement set is needed.

9.9a.3.7.6 Complete update of the tracked MCCAOP reservations of a neighbor mesh STA

The mesh STA performed the steps in 9.9a.3.7.5 and detected that the MCCAOP advertisement set sequence number has been updated. Consequently, the mesh STA shall operate as follows.

The mesh STA shall discard all MCCAOP reservations that it tracked for the sender of the received MCCAOP advertisement. The mesh STA shall record the Advertisement Set Sequence Number and the source address (SA) of the received MCCAOP advertisement. The mesh STA shall record all reservations in the MCCAOP Advertisement elements of the received MCCAOP advertisement.

If the mesh STA does not receive all MCCAOP Advertisement elements of the sender of the MCCAOP advertisement before a frame exchange sequence on the wireless medium causes the mesh STA to set its NAV, the mesh STA shall perform the MCCAOP advertisement request procedure as described in 9.9a.3.7.8.

9.9a.3.7.7 Partial update of the tracked MCCAOP reservations of a neighbor mesh STA

The mesh STA performed the steps in 9.9a.3.7.5 and detected that part of the MCCAOP advertisement set of the sender of the MCCAOP advertisement has been updated. Consequently, the mesh STA shall operate as follows for each bit in the Advertisement Elements Bitmap contained in the MCCAOP Advertisement Overview element of the received MCCAOP advertisement.

If the bit in position n of the Advertisement Elements Bitmap in the received MCCAOP Advertisement is equal to 0 and if the bit in position n of the Advertisement Elements Bitmap tracked for the sender of the received MCCAOP advertisement is equal to 1, the mesh STA shall delete the reservations with the same Advertisement Sequence Number and the same MCCAOP Advertisement Element Index received from the same sender from its tracked reservations.

If the bit in position n of the Advertisement Elements Bitmap in the received MCCAOP Advertisement is equal to 1 and if the bit in position n of the Advertisement Elements Bitmap tracked for the sender of the received MCCAOP advertisement is equal to 0, the mesh STA shall add the reservations of the received MCCAOP Advertisement element with the MCCAOP Advertisement Element Index set to n to its tracked reservations. If the mesh STA does not receive this MCCAOP Advertisement element of the sender of the MCCAOP Advertisement before a frame exchange sequence on the wireless medium causes the mesh STA to set its NAV, the mesh STA shall perform the MCCAOP Advertisement request procedure as described in 9.9a.3.7.8.

NOTE—If the bit in position n of the received Advertisement Elements Bitmap contained in the received MCCAOP Advertisement is equal to the bit in position n of the Advertisement Elements Bitmap tracked for the sender of the received MCCAOP advertisement, then the Advertisement element with the MCCAOP Advertisement Element Index equal to n is current, and no update of this Advertisement element is needed.

9.9a.3.7.8 MCCAOP advertisement request procedure

To request all MCCAOP Advertisement elements from a neighbor peer mesh STA, the mesh STA transmits an MCCA Advertisement Request frame without an MCCAOP Advertisement Overview element.

To request a subset of the MCCAOP Advertisement elements of a neighbor peer mesh STA, the mesh STA transmits an MCCA Advertisement Request frame including an MCCAOP Advertisement Overview element. The mesh STA shall set the contents of the MCCAOP Advertisement Overview element as follows. The mesh STA sets

a) The Advertisement Set Sequence Number field to the Advertisement Sequence Number that it tracks for the recipient of this frame

b) In the Advertisement Element Bitmap, the bit to 1 for each MCCAOP Advertisement Element that the mesh STA requests from the recipient of this frame

c) The Flags field, the MCCA Access Fraction field, and the MAF Limit field to zero

The mesh STA shall discard the MCCA Advertisement Request frame from its frame queue if it receives all of the MCCAOP Advertisement elements that it requests in the MCCAOP Advertisement Request.

9.9a.3.8 MCCAOP teardown

9.9a.3.8.1 Conditions that trigger an MCCAOP teardown

The MCCAOP owner and the MCCAOP responder may initiate a teardown of an MCCAOP reservation, e.g., when the reservation is no longer needed. A mesh STA shall act as follows to resolve conflicts between MCCAOP reservations in its neighborhood MCCAOP periods. If the conflict is caused by overlapping reservations from its TX-RX periods and broadcast periods, it shall select one of these reservations and initiate a teardown for it. If the conflict is caused by an overlap between a reservation from its TX-RX periods or broadcast periods, and another reservation from its interference periods, it shall act as follows. It creates a first unsigned integer by inverting the bit order of its MAC address and a second unsigned integer by inverting the bit order of the lowest of the known MAC addresses of the owner and responder(s) of the reservation in the interference periods. If the first unsigned integer is smaller than the second unsigned integer, it shall initiate a teardown of the reservation in its TX-RX or broadcast periods. Otherwise, it may initiate a teardown of the reservation in its TX-RX or broadcast periods.

There are also other conditions that trigger the MCCAOP owner and responder to delete a reservation, without an explicit tear down. An MCCAOP owner shall delete a reservation for an individually addressed transmission when it has not received an acknowledgement for any frame transmission in the MCCAOPs corresponding to the reservation for greater than dot11MCCAOPtimeout time. An MCCAOP responder shall delete a reservation for individually addressed transmission or group addressed transmissions when it has not received a frame transmission in any of the MCCAOPs corresponding to the reservation for greater than dot11MCCAOPtimeout time.

9.9a.3.8.2 MCCAOP teardown procedure

The teardown is initiated by transmitting an MCCA Teardown frame. The MCCAOP Reservation ID field in the MCCAOP Teardown element is set to the MCCAOP Reservation ID of the reservation that is to be torn down. In case the tear down is initiated by an MCCAOP responder, the MCCAOP Owner field of the MCCAOP Teardown element is set to the MAC address of the MCCAOP owner.

The transmitter of the MCCA Teardown frame deletes the reservation after the MCCA Teardown frame has been successfully transmitted. The receiver of the MCCA Teardown frame acts as follows. In case the MCCAOP Reservation ID field corresponds to a reservation for individually addressed transmissions, it deletes the reservation. If the reservation is for group addressed transmissions for which it is the MCCAOP owner, it deletes the reservation if there are no other MCCAOP responders for this reservation.

The MCCAOP owner acts as follows when deleting a reservation:

- It stops executing the access procedure described in 9.9a.3.9.1 at the start of the MCCAOPs corresponding to the reservation that was deleted.
- In case the reservation was for individually addressed frames, it stops advertising the MCCAOP reservation in its TX-RX Periods Report.
- In case the reservation was for group addressed frames, it stops advertising the MCCAOP reservation in its Broadcast Periods Report.

The MCCAOP responder acts as follows when deleting a reservation:

- It stops executing the procedure described in 9.9a.3.9.2 during the MCCAOPs corresponding to the reservation that was deleted.
- In case the reservation was for individually addressed frames, it stops advertising the MCCAOP reservation in its TX-RX Periods Report.
- In case the reservation was for group addressed frames, it stops advertising the MCCAOP reservation in its Broadcast Periods Report.

9.9a.3.9 Access during MCCAOPs

9.9a.3.9.1 Access by MCCAOP owners

At the start of the MCCAOP, the EDCAF of the MCCAOP owner shall set AIFSN[AC] equal to dot11MCCAAIFSN, CWmax[AC] equal to dot11MCCACWmax, CW[AC] equal to dot11MCCACWmin, QSRC[AC] to 0, and QLRC[AC] to 0 for all ACs. The TXOP limit shall specify a duration value no larger than the MCCAOP Duration.

During the MCCAOP, the EDCAFs of the ACs operates as specified in 9.9.1, with the following modifications.

— During the MCCAOP, the EDCAF of each AC shall consider only those frame whose RA matches the MAC address of the MCCAOP responder.

— In cases where the access to the medium is delayed, the TXOPlimit value shall specify a duration to end no later than the MCCAOP start time plus the MCCAOP Duration.

— As specified in 9.9a.3.9.2, neighboring STAs shall not access the WM during an MCCAOP, until they receive a frame from either the MCCAOP owner or the MCCAOP responder. With the exception of truncation of an MCCA TXOP by means of a CF-End, standard EDCA TXOP rules apply for the remainder of the MCCAOP. For HT mesh STAs, these include the reverse direction protocol as specified in 9.15.

— At the end of the MCCAOP, the parameters used by the EDCAF of the MCCAOP owner shall be set to the MIB attribute table dot11EDCATable, and QSRC[AC] and QLRC[AC] shall be set to 0 for all ACs.

The MCCAOP owner may adjust the duration of an MCCAOP by setting the Duration/ID field in the frames it transmits. In particular, if an MCCAOP owner has no data to transmit in an MCCAOP corresponding to an MCCAOP reservation that is intended for individually addressed frames, it may transmit an individually addressed QoS Null frame during the MCCAOP to end the MCCAOP.

NOTE—It is recommended to send a QoS Null frame to end the MCCAOP although there might be situations in which the transmission of a QoS Null is not needed or undesirable.

If an MCCAOP owner has no data to transmit in an MCCAOP reservation that is intended for group addressed frames, it may transmit a group addressed QoS Null frame during the MCCAOP to end the MCCAOP.

9.9a.3.9.2 Access during an MCCAOP by mesh STAs that are not the MCCAOP owner

The MAC of a mesh STA with dot11MCCAActivated is true shall provide a Reservation Allocation Vector (RAV) mechanism to indicate a busy medium from the start of an MCCAOP corresponding to a reservation in its interference periods until the receipt of a frame transmitted by either the MCCAOP owner or the MCCAOP responder. The RAV mechanism is provided in addition to the PHY and virtual CS mechanisms described in 9.2.1. It is different from the virtual CS mechanism in two aspects. Firstly, a mesh STA might be neighbor to multiple ongoing MCCAOPs corresponding to different reservations and the regular NAV setting and updating rules do not suffice to prevent interference during these reservations. Secondly, the virtual CS mechanism is set immediately upon receipt of a frame, whereas the RAV mechanism is based on reservation frames received at some earlier time instant. When either the CS function provided by the PHY, the virtual CS function provided by the MAC via the NAV, or the RAV mechanism indicate a busy medium during an MCCAOP for which the mesh STA is neither the MCCAOP owner nor the MCCAOP responder, the medium shall be considered busy; otherwise, it shall be considered idle.

The RAV mechanism maintains an index of future MCCAOPs based on the reservation information that is available in the interference periods of a mesh STA. At the start of each MCCAOP corresponding to a reservation in the interference periods, a RAV is set to indicate a busy medium for the duration of the

MCCAOP given in the MCCAOP Duration subfield of the MCCAOP reservation. At the start of each MCCAOP corresponding to a reservation in the TX-RX or broadcast periods for which the mesh STA is an MCCAOP responder, a RAV is set to indicate a busy medium for the duration of the MCCAOP given in the MCCAOP Duration subfield of the MCCAOP reservation. The RAV may be thought of as a counter, corresponding to an MCCAOP corresponding to a reservation in the interference periods. The RAV counts down to zero at a uniform rate. When the counter is zero, the RAV indication is that the medium is idle; when nonzero, the indication is busy.

The mesh STA clears the RAV timer, i.e., sets it to 0, upon receipt of a frame from either the MCCAOP owner or responder. If a mesh STA receives an RTS frame during an MCCAOP for which it is a MCCAOP responder, with the RA address matching its MAC address and with the MAC address in the TA field in the RTS frame matching the MAC address of the MCCAOP owner, then the STA shall send the CTS frame after SIFS, without regard for the NAV and the RAV, and without resetting its NAV. The RAV for an MCCAOP is not cleared upon receipt of a frame originating from stations that are not the MCCAOP owner or responder. Since the NAV is set upon receipt of frames with a Duration/ID field, the MCCAOP owner and responder adjust the reservation period of an MCCAOP to their actual traffic needs by the Duration/ID field in the transmitted frame and obtain protection of the frame transmission via the NAV setting.

The RAV mechanism might be represented by a number of counters, where each counter corresponds to one MCCAOP. The number of counters needed at any instant is equal to the number of MCCAOPs at this instant corresponding to reservations in the interference periods of the mesh STA.

9.9a.3.10 Interaction with time synchronization

If a mesh STA adjusts its TBTT, e.g., in response to a TBTT Adjustment Request, it shall adjust the reservations by modifying the MCCAOP Offset of each of the tracked MCCAOP reservations. If a mesh STA adjusts its timing offset value with respect to a neighbor mesh STA, as specified in 11C.12.2.2, it shall adjust the reservations by modifying the MCCAOP Offset of each of the tracked MCCAOP reservations for which this neighbor mesh STA is the owner. In either case, an MCCAOP advertisement of a mesh STA shall always contain the most recent MCCAOP Offsets.

9.13 Protection mechanisms

9.13.2 Protection mechanisms for non-ERP receivers

Insert the following paragraphs after the seventh paragraph in 9.13.2:

A NonERP mesh STA shall set the NonERP_Present and Use_Protection bits to 1, when establishing a mesh peering with a mesh STA.

When a mesh STA establishes a mesh peering with a NonERP mesh STA, the mesh STA shall set the NonERP_Present bit to 1 and the mesh STA should set the Use_Protection bit to 1. In addition, a mesh STA should set the NonERP_Present bit and the Use_Protection bit to 1 when

— A mesh STA detects the overlapped presence of either a NonERP BSS, a NonERP IBSS, or a NonERP MBSS, or

— A Beacon frame is received from a neighbor STA where the supported rate set contains only Clause 15 or Clause 18 rates, or

— A management frame (excluding Probe Request) is received where the supported rate set includes only Clause 15 or Clause 18 rates.

A mesh STA may set the NonERP_Present and the Use_Protection bits to 1 based on its internal policies, which is beyond the scope of the standard.

ERP mesh STAs shall invoke the use of a protection mechanism after the transmission of the Use_Protection bit with a value of 1 in an MMPDU. In addition, ERP mesh STAs may invoke protection mechanism at other times. ERP mesh STAs may disable protection mechanism use after transmission of the Use_Protection bit with a value of 0 in an MMPDU.

When dot11ShortPreambleOptionImplemented is true and all peer mesh STAs support the short preamble, the mesh STA may set the Baker_Preamble_Mode bit to 0. When dot11ShortPreambleOptionImplemented is false or any of its peer mesh STAs do not support the short preamble, the mesh STA shall set the Barker_Preamble_Mode field to 1.

ERP mesh STAs shall use long preambles when transmitting Clause 15, Clause 18, and Clause 19 frames after transmission or reception of an ERP Information element with a Barker_Preamble_Mode value of 1 in an MMPDU to or from the MBSS to which the ERP mesh STA belongs. Mesh STAs may additionally use long preambles when transmitting Clause 15, Clause 18, and Clause 19 frames at other times.

9.13.3 Protection mechanisms for transmissions of HT PPDUs

9.13.3.1 General

Change the seventh paragraph in 9.13.3.1 as follows:

In an MBSS, the HT Protection field and the Nongreenfield HT STAs Present field are determined as described in 9.13.3.4a.

In an IBSS and an MBSS, the RIFS Mode field of the HT Operation element is also reserved, but an HT STA shall operate as though this field were set to 1.

Insert the following new subclause after 9.13.3.4:

9.13.3.4a Protection rules for an HT mesh STA in an MBSS

A mesh STA determines the HT Protection and Nongreenfield HT STAs Present fields in the HT Operation element in the transmitting frame as follows:

The HT Protection field in a mesh STA may be set to no protection mode only if

— All STAs detected in the primary or the secondary channel are HT STAs, and

— All mesh STA members of this MBSS that are one-hop neighbors of the transmitting mesh STA are either:

  — 20/40 MHz HT mesh STAs in a 20/40 MHz MBSS, or

  — 20 MHz HT mesh STAs in a 20 MHz MBSS.

The HT Protection field in a mesh STA may be set to non-member protection mode only if

— A non-HT STA is detected in either the primary or the secondary channel or in both the primary and secondary channels, that is not known by the transmitting mesh STA to be a member of this MBSS, and

— All mesh STA members of this MBSS that are one-hop neighbors of the transmitting mesh STA are HT mesh STAs.

The HT Protection field in a mesh STA may be set to 20 MHz protection mode only if

— All STAs detected in the primary and all STAs detected in the secondary channel are HT STAs and All mesh STA members of this MBSS that are one-hop neighbors of the transmitting mesh STA are HT mesh STAs, and

- The MBSS is a 20/40 MHz MBSS, and
- There is at least one 20 MHz HT mesh STA that is one-hop neighbor of the transmitting mesh STA.

The HT Protection field in a mesh STA is set to non-HT mixed mode otherwise.

If two peer HT mesh STAs report the same protection mode in HT Protection field, the protection mechanisms of the related mode shall be used to protect the transmission between the peer HT mesh STAs.

If an HT mesh STA and its peer HT mesh STA report different protection modes in HT Protection field, the following rules shall be used:

a) If an HT mesh STA or its peer HT mesh STA reports non-HT mixed mode, the protection mechanisms of non-HT mixed mode shall be used to protect the transmission between the peer HT mesh STAs.
b) If an HT mesh STA or its peer HT mesh STA reports non-member protection mode and non-HT mixed mode is not reported by any of these HT mesh STAs, the protection mechanisms of non-member protection mode shall be used to protect the transmission between the peer HT mesh STAs.
c) If an HT mesh STA or its peer HT mesh STA reports 20 MHz protection mode and neither non-HT mixed mode nor non-member protection mode is reported by any of these HT mesh STAs, the protection mechanisms of 20 MHz protection mode shall be used to protect the transmission between the peer HT mesh STA.

If at least one HT peer mesh STA in its mesh neighborhood indicates the Nongreenfield HT STAs Present equal to 1, the protection rules related to Nongreenfield HT STAs Present should also be applied to the communication between HT peer mesh STAs.

9.13.5 L-SIG TXOP protection

9.13.5.1 General rules

Change the fourth paragraph in 9.13.5.1 as follows:

In an IBSS and an MBSS, the L-SIG TXOP Protection Full Support field of the HT Operation element is reserved, but HT STAs shall operate as though the field were set to 0.

Change the first dashed list item after the sixth paragraph in 9.13.5.1 as follows:

- The frame initiates a TXOP in an IBSS or in an MBSS, or

Insert the following new subclause after 9.21:

9.22 Mesh forwarding framework

9.22.1 General

The term “mesh forwarding” refers to forwarding of MSDUs and MMPDUs on paths determined by the mesh path selection between mesh STAs at the link layer. The mesh paths are contained in the forwarding information. The forwarding information, for instance, the lifetime of a mesh path, may be updated as a consequence of mesh forwarding.

The forwarding of MSDUs and MMPDUs within an MBSS is described in 9.22.4, 9.22.5, 9.22.6, and 9.22.9. The forwarding of MSDUs and MMPDUs between the MBSS and the DS at proxy mesh gates is described in 11C.10.3.

9.22.2 Forwarding information

Forwarding information is created by the active mesh path selection protocol and is utilized for MSDU/MMPDU forwarding as described in 9.22.4 and 9.22.6.2.

The basic forwarding information to a destination mesh STA consists of the destination mesh STA address, the next-hop address, the precursor list, and the lifetime of this forwarding information.

An entry in the precursor list contains the precursor mesh STA address and the lifetime of this entry. If an existing entry in a precursor list is updated, the lifetime is the maximum of the current and the updated value. If the lifetime of a precursor expires, it will be deleted from the precursor list. Precursors are used to identify legitimate transmitters of individually addressed frames (see 9.22.4.2) and for the notification of link failures (in case of HWMP, see 11C.9.11).

The forwarding information shall be considered as invalid if its lifetime has expired. Also, forwarding information is marked as invalid when certain conditions are met in the processing of mesh path selection elements, e.g., path error processing in HWMP (11C.9.11.4).

The active path selection protocol may define additional parameters in the forwarding information. Details on the additional parameters of the forwarding information constructed by the hybrid wireless mesh protocol (HWMP) are described in 11C.9.8.4.

9.22.3 Frame addressing in an MBSS

Mesh Data frames and Multihop Action frames enable multihop MSDU and MMPDU forwarding in an MBSS using the Mesh Control field described in 7.1.3.6.3. In this subclause, addressing of the Mesh Data and Multihop Action frames and MSDU/MMPDU forwarding behavior are described.

Table 9-13 shows the valid combinations of address fields in Mesh Data frames and Multihop Action frames along with the corresponding value of the Address Extension Mode subfield in the Mesh Control field.

NOTE 1—ToDS and FromDS fields are located in the Frame Control field (see 7.1.3.1.3). The Address Extension Mode subfield is located in the Mesh Flags subfield in the Mesh Control field (see 7.1.3.6.3). Address 1, Address 2, and Address 3 fields are located in the MAC header (see 7.1.2). The Address 4 field is located in the MAC header if both ToDS and FromDS fields are 1; otherwise, the Address 4 field is located in the Mesh Address Extension subfield of the Mesh Control field (see 7.1.2 and 7.1.3.6.3). Address 5 and Address 6 fields are located in the Mesh Control field if they are present (see 7.1.3.6.3).

In individually addressed Mesh Data and Multihop Action frames, Address 1 and Address 2 correspond to the mesh STA receiver address (RA) and the mesh STA transmitter address (TA) for a particular mesh link. Address 3 and Address 4 correspond to the destination end station and the source end station of a mesh path. The Address Extension Mode subfield in the Mesh Control field indicates the presence of an optional Mesh Address Extension subfield in the Mesh Control field. When the Extension Mode subfield equals 10 (binary), the Mesh Control field includes Address 5 and Address 6 that correspond to the end-to-end destination address (DA) and source address (SA) of STAs that communicate over the mesh path, for instance, external STAs that communicate over the mesh BSS via proxy mesh gates (see Figure 9-38).

NOTE 2—The forwarding of individually addressed Mesh Data frames uses only mesh STA addresses in fields Address 1, Address 2, Address 3, and Address 4. This allows intermediate mesh STAs to forward Mesh Data frames without necessarily having any knowledge of the addresses of the source and destination end stations, which might be external addresses. Thus, proxy information only needs to be maintained by proxy mesh gates and by source mesh STAs.

The term *source mesh STA* refers to the first mesh STA on a mesh path. A source mesh STA may be a mesh STA that is the initial source of an MSDU/MMPDU or a mesh STA that receives an MSDU/MMPDU from a mesh path or from a STA outside the mesh BSS and translates and forwards the MSDU/MMPDU on the mesh path. The address of the source mesh STA is referred to as the Mesh SA.

Table 9-13—Valid address field usage for Mesh Data and Multihop Action frames

Supported frames	ToDS FromDS field	Address Extension Mode value (binary)	Address 1	Address 2	Address 3	Address 4	Address 5	Address 6
Mesh Data (individually addressed)	11	00	RA	TA	DA = Mesh DA	SA = Mesh SA	*Not Present*	*Not Present*
Mesh Data (group addressed)	01	00	DA	TA	SA = Mesh SA	*Not Present*	*Not Present*	*Not Present*
Mesh Data (proxied, individually addressed)	11	10	RA	TA	Mesh DA	Mesh SA	DA	SA
Mesh Data (proxied, group addressed)	01	01	DA	TA	Mesh SA	SA	*Not Present*	*Not Present*
Multihop Action (individually addressed)	00	01	RA	TA	DA = Mesh DA	SA = Mesh SA	*Not Present*	*Not Present*
Multihop Action (group addressed)	00	00	DA	TA	SA = Mesh SA	*Not Present*	*Not Present*	*Not Present*

The term destination mesh STA refers to the final mesh STA on a mesh path. A destination mesh STA may be a mesh STA that is the final destination of an MSDU/MMPDU or a mesh STA that receives an MSDU/MMPDU from a mesh path and translates and forwards the MSDU/MMPDU on another mesh path or to a STA outside of the mesh BSS. The address of the destination mesh STA is referred to as the Mesh DA.

In group addressed Mesh Data frames, Address 1 and Address 2 correspond to the group address and the mesh STA transmitter address (TA). Address 3 corresponds to the mesh source address (mesh SA) of the group addressed Mesh Data frame. The Address Extension Mode indicates the presence of an optional address extension field Address 4 in the Mesh Control field that corresponds to the source address (SA) of external STAs that communicate over the mesh BSS via proxy mesh gates.

NOTE 3—The reason for not using the four-address MAC header format for group addressed traffic is to avoid interactions with existing implementations. Earlier revisions of this standard defined the four-address MAC header format without defining procedures for its use. As a result there is a large number of deployed devices that use the four-address frame format in ways that would affect and be affected by mesh traffic if four-address group addressed frames were to be used.

Figure 9-38 illustrates the addressing of a Mesh Data frame that contains an MSDU transmitted and forwarded on a mesh path from a mesh STA collocated with a portal (STA 1) to a mesh STA collocated with an AP (STA 2) where the source is a STA outside of the mesh BSS (STA 33) that is reachable via the portal and the destination is an IEEE 802.11 STA associated with the AP (STA 22).

Details on how these address mappings work in forwarding processing are described in 9.22.4 and 9.22.5.

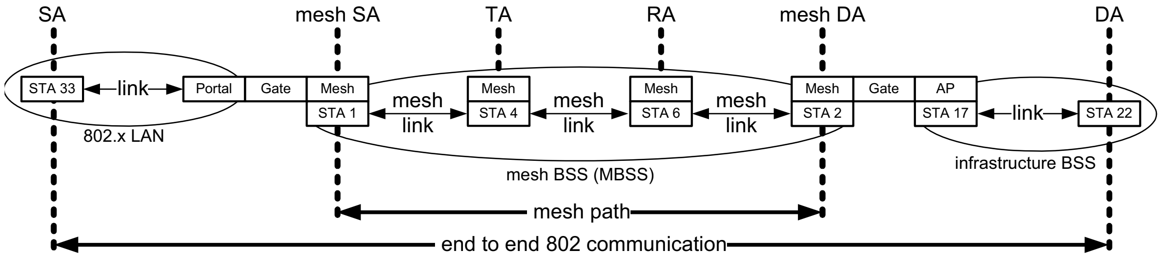


Figure 9-38—Example addressing for a Mesh Data frame

9.22.4 Addressing and forwarding of individually addressed Mesh Data frames

9.22.4.1 At source mesh STAs (individually addressed)

MSDUs sent by a mesh STA (as a consequence of an MA-UNITDATA.request with an individual destination address) and destined to another mesh STA in the MBSS shall be transmitted using a frame with the four-address MAC header format [with the Address Extension Mode subfield in the Mesh Control field set to 00 (binary)], where the four address fields are set as follows [see row “Mesh Data (individually addressed)” in Table 9-13]:

- — Address 1: The address of the next-hop mesh STA (toward the destination mesh STA according to the forwarding information—see 9.22.2)
- — Address 2: The address of the transmitter mesh STA
- — Address 3: The address of the destination mesh STA
- — Address 4: The address of the source mesh STA

MSDUs that are sent by a mesh STA as a consequence of a MA-UNITDATA.request with an individual destination address and are destined to an address that is different from the mesh STA at the end of a mesh path shall be transmitted using a frame with the four-address MAC header format [with the Address Extension Mode subfield in the Mesh Control field set to 10 (binary)], where the Mesh Address Extension subfield in the Mesh Control field carries the addresses of the end stations, as specified in row “Mesh Data (proxied, individually addressed)” of Table 9-13. The additional addresses 5 and 6 are defined as follows:

- — Address 5: The address of the destination end mesh STA (may be the same as Address 3 if the destination is the mesh STA at the end of the mesh path)
- — Address 6: The address of the source end mesh STA (may be the same as Address 4 if the source is the mesh STA at the beginning of the mesh path)

NOTE—The destination address is distinct from the mesh STA at the end of the mesh path in two cases: 1) when the destination is an external address and 2) when the destination is a mesh STA distinct from the destination mesh STA at the end of the mesh path. The former case is described in 11C.10.3. The latter case might occur if a source mesh STA sends the MSDU to another intermediate mesh STA that sends the MSDU on a different mesh path to the destination mesh STA in the MBSS.

The Mesh TTL subfield in the Mesh Control field shall be set to the value of dot11MeshTTL.The MSDUs are forwarded multiple hops, limited by the Mesh TTL value.

The source mesh STA shall set the Mesh Sequence Number subfield in the Mesh Control field to a value from a modulo-2^{32} counter that is incremented by 1 for each new MSDU transmitted with a Mesh Control field and for each new MMPDU transmitted using a Multihop Action frame.

9.22.4.2 At intermediate and destination mesh STAs (individually addressed)

On receipt of an individually addressed Mesh Data frame, a mesh STA shall perform the following:

a) The mesh STA shall decipher the frame and check it for authenticity. If it is not from a peer mesh STA, the frame shall be silently discarded.

b) The mesh STA shall check to see whether the MAC address in the Address 3 field is a known destination address; if it is an unknown destination address, the mesh STA may perform any of the following three actions:

   1) Silently discard the frame.

   2) Trigger a path discovery procedure depending on the path selection protocol that is currently active in the mesh BSS. For HWMP, see 11C.9.9.3 Case A.

   3) Inform the mesh STA in Address 2 that the destination is unreachable depending on the path selection protocol that is currently active in the mesh BSS. For HWMP, see 11C.9.11.3 Case B.

c) If Address 2 is not one of the precursors for this destination mesh STA (see 9.22.2), the frame shall be discarded.

If the frame is not discarded and one or more MSDUs are collected from the frame, the mesh STA may detect duplicate MSDUs according to 9.22.7 and discard them.

If Address 3 does not match the mesh STA's own address, but is a known individual destination MAC address in the forwarding information then the following actions are taken:

- The lifetime of the forwarding information to the destination (Address 3) is set to its initial value.
- The lifetime of the forwarding information to the source (Address 4) is set to its initial value.
- The lifetime of the precursor list entry for the precursor to the destination (Address 2) is set to the maximum of the initial value and the current value.
- The lifetime of the precursor list entry for the precursor to the source (next hop to the destination) is set to the maximum of the initial value and the current value.
- The Mesh TTL in the corresponding Mesh Control field of the collected MSDU is decremented by 1. If zero has been reached, the MSDU shall be discarded.
- If the MSDU has not been discarded, the mesh STA shall forward the MSDU via a frame with the Address 1 field set to the MAC address of the next-hop mesh STA as determined from the forwarding information (see 9.22.2) and the Address 2 field set to its own MAC address and queue the frame for transmission.

If Address 3 matches the mesh STA's own MAC address, the following actions are taken:

- The lifetime of the forwarding information to the source (Address 4) is set to its initial value.
- The lifetime of the precursor list entry for the precursor to the destination (Address 2) is set to the maximum of the initial value and the current value.
- If the Address Extension Mode subfield in the Mesh Control field is 00 (binary), the MA-UNITDATA.indication primitive is passed from the MAC sublayer entity to the LLC sublayer entity or entities.
- If the Address Extension Mode subfield in the Mesh Control field is 10 (binary) and Address 5 is equal to Address 3, the mesh STA is the final destination of the MSDU, and the MA-UNITDATA.indication primitive is passed from the MAC sublayer entity to the LLC sublayer entity or entities.
- If the Address Extension Mode subfield in the Mesh Control field is 10 (binary) and Address 5 is a known destination MAC address in the forwarding information (mesh STA), the mesh STA shall forward the MSDU via a frame as described in 9.22.4.1 with the Address 3 field set to the MAC Address of the Address 5 field.

- If the Address Extension Mode subfield in the Mesh Control field is 10 (binary), the MSDU is forwarded according to 11C.10.3.2 in all other cases.

If Address 3 matches the group address, the mesh STA shall perform the procedures as given in 9.22.5.2.

Note that during the forwarding process at intermediate mesh STAs, the content of the MSDU is not changed.

9.22.5 Addressing and forwarding of group addressed Mesh Data frames

9.22.5.1 At source mesh STAs (group addressed)

MSDUs sent by a mesh STA (as a consequence of a MA-UNITDATA.request with a group destination address) shall be transmitted using a group addressed Mesh Data frame [with the Address Extension Mode subfield in the Mesh Control field set to 00 (binary)] (see row "Mesh Data (group addressed)" in Table 9-13). An implementation may circumvent the unreliability of group addressed transmissions by using multiple individually addressed Mesh Data frames, which are individually acknowledged. In such case, the frame may be converted to individually addressed frames and transmitted as individually addressed Mesh Data frames to each peer mesh STA as described in 9.22.4.1 with the Address 3 field set to the group address. The circumstances for choosing this method are outside the scope of the standard.

In group addressed Mesh Data frames, the address fields are set as follows:

- Address 1: The group address
- Address 2: The address of the transmitter mesh STA
- Address 3: The address of the source mesh STA

The source mesh STA shall set the Mesh TTL subfield in the Mesh Control field to dot11MeshTTL in order to control the hop count. The MSDUs are forwarded multiple hops, limited by the Mesh TTL value. For example, if the Mesh TTL subfield is 1, MSDUs are delivered only to immediate neighbors.

The source mesh STA shall set the Mesh Sequence Number subfield in the Mesh Control field to a value from a modulo-2^{32} counter that is incremented by 1 for each new MSDU transmitted with a Mesh Control field and for each new MMPDU transmitted using a Multihop Action frame.

Procedures that enhance the reliability or efficiency of group addressed transmissions are outside the scope of this standard.

9.22.5.2 At recipient mesh STAs (group addressed)

On receipt of a group addressed Mesh Data frame with Address 1 (DA) equal to the group address, or on receipt of an individually addressed Mesh Data frame with Address 3 (Mesh DA) equal to the group address, a mesh STA shall perform the following:

a) The mesh STA shall decipher the frame and check it for authenticity. If it is not from a peer mesh STA, the frame shall be silently discarded.

b) If the frame is not discarded and one or more MSDUs are collected from the frame, the mesh STA may detect duplicate MSDUs according to 9.22.7 and discard them.

c) The mesh STA decrements the Mesh TTL in the Mesh Control field. If the Mesh TTL value has reached zero, the corresponding MSDU shall not be forwarded to other mesh STAs.

d) If the Mesh TTL value has not reached zero and if dot11MeshForwarding is true, the mesh STA shall forward the MSDU via a group address Mesh Data frame with the Address 2 field set to its own MAC address.

e) If the Address Extension Mode is 01 (binary) and the recipient mesh STA is a proxy mesh gate and if the Mesh TTL value has not reached zero and if dot11MeshForwarding is true, the MSDU is forwarded according to 11C.10.3.2.

When the SA and the Mesh SA are not identical (the source address is therefore an external address), the MSDU shall be forwarded by using a frame with the three-address MAC header format [with the Address Extension Mode subfield in the Mesh Control field set to 01 (binary)] as specified in row "Mesh Data (proxied, group addressed)" of Table 9-13. Otherwise, the MSDU shall be forwarded by using a frame with the three-address MAC header format [with the Address Extension Mode subfield in the Mesh Control field set to 00 (binary)] as specified in row "Mesh Data (group addressed)" of Table 9-13.

An implementation may circumvent the unreliability of group addressed transmissions by using multiple individually addressed Mesh Data frames, which are individually acknowledged. In such case, the frame may be converted to individually addressed frames and transmitted as an individually addressed Mesh Data frame to each peer mesh STAs as described in 9.22.4.2 with the Address 3 field set to the group address. If the Address Extension Mode subfield in the Mesh Control field in the group addressed Mesh Data frame is equal to 01 (binary), the Address Extension Mode subfield in the Mesh Control field in the individually addressed Mesh Data frames is set to 10 (binary), the Address 5 field is set to the group address, and the Address 6 field set to the Source Address contained in the Address 4 field of the group address Mesh Data frame. The circumstances for choosing this method and the ability to determine all the addresses of the neighbor peer mesh STAs are beyond the scope of the standard.

If one or more MSDUs collected from the frame have not been discarded, the MA-UNITDATA.indication primitive is passed from the MAC sublayer entity to the LLC sublayer entity or entities.

9.22.6 Addressing of Management frames and MMPDU forwarding

9.22.6.1 General

All MMPDUs except MMPDUs transmitted using Multihop Action frames are transmitted over only one hop to peer mesh STAs.

NOTE—In several cases, the reception and processing of an Action frame leads to the transmission of a new Action frame of the same type that might include an identical or a modified version of the contents from the elements of the received Action frame. This is called propagation in contrast to forwarding.

A mesh STA may convert a group addressed management frame to individually addressed management frames and transmit them as individually addressed frames to each peer mesh STA, if the frame is intended to be delivered only to its peer mesh STAs. The circumstances for choosing this method are outside the scope of the standard.

9.22.6.2 MMPDU forwarding using individually addressed Multihop Action frames

MMPDUs sent by a mesh STA and destined to another mesh STA in the MBSS using individually addressed Multihop Action frames (see 7.4.16) shall be transmitted using a management frame with the three-address MAC header format [with the Address Extension Mode subfield in the Mesh Control field set to 01 (binary)], where the four address fields are set as follows [see row "Multihop Action (individually addressed)]" in Table 9-13:

— Address 1: The address of the next-hop mesh STA (toward the destination mesh STA according to the forwarding information—see 9.22.2.
— Address 2: The address of the transmitter mesh STA.
— Address 3: The address of the destination mesh STA.
— Address 4: The address of the source mesh STA.

The source mesh STA shall set the Mesh TTL subfield in the Mesh Control field to the value of dot11MeshTTL, and set the Mesh Sequence Number subfield in the Mesh Control field to a value from a modulo-2^{32} counter that is incremented by 1 for each new MSDU transmitted with a Mesh Control field and for each new MMPDU transmitted using a Multihop Action frame.

At intermediate and destination mesh STAs, on receipt of an individually addressed Multihop Action frame, the address matching, the reception procedures, the forwarding information update, and the Mesh TTL decrement are performed as described in 9.22.4.2, and the MMPDU is forwarded according to the forwarding information and the procedures in 9.22.4.2.

At intermediate mesh STAs, frame fields following the Mesh Control field are not required to be examined.

If the Address 3 in the received Multihop Action frame matches the mesh STA’s own MAC address or the group address, the mesh STA (destination mesh STA) shall process the content of the MMPDU.

9.22.6.3 MMPDU forwarding using group addressed Multihop Action frames

MMPDUs sent by a mesh STA and destined to all other mesh STAs in the MBSS using group addressed Multihop Action frames (see 7.4.16) shall be transmitted using a management frame with the three-address MAC header format [with the Address Extension Mode subfield in the Mesh Control field set to 00 (binary)], where the three address fields are set as follows [see row “Multihop Action (group addressed)” in Table 9-13]:

- Address 1: The group address
- Address 2: The address of the transmitter mesh STA
- Address 3: The address of the source mesh STA

An implementation may circumvent the unreliability of group addressed transmissions by using multiple individually addressed Multihop Action frames, which are individually acknowledged. In such case, the frame may be converted to individually addressed frames and transmitted as individually addressed Multihop Action frames to each peer mesh STA as described in 9.22.6.2 with the Address 3 field set to the group address. The circumstances for choosing this method are outside the scope of the standard.

The source mesh STA shall set the Mesh TTL subfield in the Mesh Control field to dot11MeshTTL, and set the Mesh Sequence Number subfield in the Mesh Control field to a value from a modulo-2^{32} counter that is incremented by 1 for each new MSDU transmitted with a Mesh Control field and for each new MMPDU transmitted using a Multihop Action frame.

At recipient mesh STAs, on receipt of Multihop Action frame, the address matching, the reception procedures, the forwarding information update, and the Mesh TTL decrement are performed as described in 9.22.5.2, and the MMPDU is forwarded according to the forwarding information and the procedures in 9.22.5.2.

If the Address 1 in the received Multihop Action frame matches the group address, the mesh STA shall process the content of the MMPDU.

Procedures that enhance the reliability or efficiency of group addressed transmissions are outside the scope of this standard.

9.22.7 Detection of duplicate MSDUs/MMPDUs

A mesh STA may receive multiple copies of the same MSDU or MMPDU from different neighbor peer mesh STAs.

The filtering of such duplicates is facilitated through the inclusion of a Mesh Sequence Number subfield in the Mesh Control field in Mesh Data frames and Multihop Action frames as specified in 7.1.3.6.3.

The receiving mesh STA shall keep a cache of recently received <Mesh SA, Mesh Sequence Number> tuples. The Mesh Source Address (Mesh SA) is contained in Address 4 for individually addressed Mesh Data frames and Multihop Action frames. The Mesh Source Address (Mesh SA) is contained in Address 3 for group addressed Mesh Data frames.

A mesh STA shall reject an MSDU/MMPDU with a Mesh Control field as a duplicate if it matches a <Mesh SA, Mesh Sequence Number> tuple of an entry in the cache.

The rules in 9.2.9 (Duplicate detection and recovery) also apply to the filtering of duplicates sent by the same neighbor peer mesh STA.

9.22.8 Mesh STAs that do not forward

A mesh STA that has dot11MeshForwarding equal to false does not forward either MSDUs, or MMPDUs of type Multihop Action. The circumstances in which a mesh STA may be allowed to become a non-forwarding entity and the authority to set dot11MeshForwarding to false are beyond the scope of this standard.

A mesh STA that does not forward is a special case of a mesh STA. Such mechanism depends on whether the path selection protocol provides a mechanism to allow mesh STAs not to participate in forwarding. The HWMP path selection protocol provides such a mechanism, see 11C.9.

9.22.9 Frame forwarding and unknown destination

A source mesh STA in the MBSS might not able to forward an MSDU that it has received as a consequence of an MA-UNITDATA.request with an individual destination address. This is the case if the destination of the MSDU is unknown to the mesh STA. The destination is unknown to a mesh STA if the mesh STA has no forwarding information for this destination or if the destination is not in its proxy information as an external STA (see 11C.10.4.2). Note that the procedure to determine that an address is unknown depends on the active path selection protocol. It may require an attempt to establish a path to the destination (see 11C.7).

If the source mesh STA is not able to forward the frame because its destination is unknown, the mesh STA shall assume that the destination is outside the MBSS and shall forward the frame to known mesh gates in the MBSS as an individually addressed frame according to the procedures for frame addressing and data forwarding of individually addressed frames at source mesh STAs in an MBSS (9.22.4.1). The MSDU shall be transmitted using a frame with the four-address MAC header format (with the Address Extension Mode subfield in the Mesh Control field set to 10 (binary)), where the Mesh Address Extension subfield in the Mesh Control field carries the address of the destination end station, as specified in row "Mesh Data (proxied, individually addressed)" of Table 9-13. The address fields are set as follows:

- — Address 1: The address of the next-hop mesh STA (toward the known mesh gate in the MBSS according to the forwarding information—see 9.22.2)
- — Address 2: The address of the source mesh STA
- — Address 3: The address of the known mesh STA in the MBSS
- — Address 4: The address of the source mesh STA

- Address 5: The address of the destination end mesh STA, which is the unknown destination address of the MSDU
- Address 6: The address of the source mesh STA, which is the same as Address 4

If there is no mesh gate available, the mesh STA shall silently discard the frame.

Discovery of mesh gates by mesh STAs is performed using propagated elements, such as a GANN (11C.10.2). Other methods specific to the HWMP path selection protocol are also available, such as the proactive PREQ (11C.9.4.2) or the proactive RANN (11C.9.4.3), when the Gate Announcement subfield in the Flags field in these HWMP elements is set to 1.

10. Layer management

10.3 MLME SAP interface

10.3.2 Scan

10.3.2.1 MLME-SCAN.request

10.3.2.1.2 Semantics of the service primitive

Change the primitive parameters in 10.3.2.1.2 as follows:

The primitive parameters are as follows:

```
MLME-SCAN.request(
                              BSSType,
                              BSSID,
                              SSID,
                              ScanType,
                              ProbeDelay,
                              ChannelList,
                              MinChannelTime,
                              MaxChannelTime,
                              RequestInformation,
                              SSIDList,
                              ChannelUsage,
                              AccessNetworkType,
                              HESSID,
                              MeshID,
                              VendorSpecificInfo
                              )
```

Change the BSSType row of the untitled table defining the primitive parameters in 10.3.2.1.2 as follows:

Name	Type	Valid range	Description
BSSType	Enumeration	NFRASTRUCTURE, INDEPENDENT, MESH, ANY_BSS	Determines whether infrastructure BSS, IBSS, MBSS, or ~~both~~ all are included in the scan.

Insert the following new row before the VendorSpecificInfo row of the untitled table defining the primitive parameters in 10.3.2.1.2:

Name	Type	Valid range	Description
Mesh ID	Octet string	0–32 octets	Only present if BSSType = MESH or BSSType = ANY_BSS. Specifies the desired Mesh ID or wildcard Mesh ID.

10.3.2.2 MLME-SCAN.confirm

10.3.2.2.2 Semantics of the service primitive

Change the BSSID, BSSType, Beacon Period, and DTIM Period rows in the BSSDescription parameter table in 10.3.2.2.2 as follows:

Name	Type	Valid range	Description	IBSS adoption
BSSID	MACAddress	N/A	The BSSID of the found BSS or the MAC address of the found mesh STA.	Adopt
BSSType	Enumeration	INFRASTRUCTURE, INDEPENDENT, MESH	The type of found BSS.	Adopt
Beacon Period	Integer	N/A	The Beacon period (in TU) of the found BSS ~~(in TU)~~ if the BSSType is not MESH, or of the found mesh STA if the BSSType = MESH.	Adopt
DTIM Period	Integer	As defined in 7.3.2.6	The DTIM period (in beacon periods) of the BSS ~~(in beacon periods)~~ if the BSSType is not MESH, or of the mesh STA if the BSSType = MESH.	Adopt

Insert the following new rows to the BSSDescription parameter table in 10.3.2.2.2:

Name	Type	Valid range	Description	IBSS adoption
MeshID	Mesh ID element	As defined in frame format	The value of MeshID element if such element was present in the probe response or Beacon frame, else null.	Do not adopt
MeshConfiguration	Mesh Configuration element	As defined in frame format	The values from the Mesh Configuration element if such an element was present in the probe response or Beacon frame, else null.	Do not adopt
Mesh Awake Window	Mesh Awake Window element	As defined in frame format	The values from the Mesh Awake Window element if such an element was present in the Probe response or Beacon frame, else null.	Do not adopt
BeaconTiming	Beacon Timing element	As defined in frame format	The values from the Beacon Timing element if such an element was present in the Probe response or Beacon frame, else null.	Do not adopt
MCCAOP Advertisement Overview	MCCAOP Advertisement Overview element	As defined in frame format	The values from the Beacon Timing element if such an element was present in the Probe response or Beacon frame, else null.	Do not adopt

Name	Type	Valid range	Description	IBSS adoption
MCCAOP Advertisement	MCCAOP Advertisement	As defined in frame format	The values from the Beacon Timing element if such an element was present in the Probe response or Beacon frame, else null.	Do not adopt

10.3.3 Synchronization

10.3.3.1 MLME-JOIN.request

10.3.3.1.1 Function

Change 10.3.3.1.1 as follows:

This primitive requests synchronization with a BSS, of which type is infrastructure or independent.

10.3.4 Authenticate

10.3.4.1 MLME-AUTHENTICATE.request

10.3.4.1.2 Semantics of the service primitive

Change the primitive parameter list in 10.3.4.1.2 as follows:

The primitive parameters are as follows:

MLME-AUTHENTICATE.request(
PeerSTAAddress,
AuthenticationType,
AuthenticateFailureTimeout,
Content of FT Authentication Elements,
Content of SAE Authentication Frame,
VendorSpecificInfo
)

Change the Authentication Type row of the untitled table defining the primitive parameters in 10.3.4.1.2 as follows:

Name	Type	Valid range	Description
Authentication Type	Enumeration	OPEN_SYSTEM, SHARED_KEY, FAST_BSS_TRANSITION, SAE	Specifies the type of authentication algorithm to use during the authentication process.

Insert the following new row to the untitled table defining the primitive parameters in 10.3.4.1.2:

Name	Type	Valid range	Description
Content of SAE Authentication Frame	Sequence of octets	As defined in 7.3.1.35, 7.3.1.36, 7.3.1.37, 7.3.1.38, 7.3.1.39, and 7.3.1.40	The contents of the SAE Commit Message or SAE Confirm Message. Present only if Authentication-Type indicates SAE authentication.

10.3.4.1.3 When generated

Change 10.3.4.1.3 as follows:

This primitive is generated by the SME for a STA to establish authentication with a specified peer MAC entity in order to permit Class 2 frames, or Mesh Peering Management frames for AMPE utilizing SAE authentication [when dot11AuthenticationAlgorithm is simultaneousAuthEquals (4)], to be exchanged between the two STAs. During the authentication procedure, the SME can generate additional MLME-AUTHENTICATE.request primitives.

10.3.4.2 MLME-AUTHENTICATE.confirm

10.3.4.2.2 Semantics of the service primitive

Change the primitive parameter list in 10.3.4.2.2 as follows:

The primitive parameters are as follows:

MLME-AUTHENTICATE.confirm(
PeerSTAAddress,
AuthenticationType,
ResultCode,
Content of FT Authentication Elements,
Content of SAE Authentication Frame,
VendorSpecificInfo
)

Change the Authentication Type and ResultCode rows of the untitled table defining the primitive parameters in 10.3.4.2.2 as follows:

Name	Type	Valid range	Description
Authentication-Type	Enumeration	OPEN_SYSTEM, SHARED_KEY, FAST_BSS_TRANSITION, SAE	Specifies the type of authentication algorithm that was used during the authentication process. This value ~~must~~ matches the AuthenticationType parameter specified in the corresponding MLME-AUTHENTICATE.request primitive.

Name	Type	Valid range	Description
ResultCode	Enumeration	SUCCESS, INVALID_PARAMETERS, TIMEOUT, TOO_MANY_SIMULTANEOUS_REQUESTS, REFUSED, ANTI-CLOGGING TOKEN REQUIRED, FINITE CYCLIC GROUP NOT SUPPORTED, AUTHENTICATION REJECTED	Indicates the result of the MLME-AUTHENTICATE.request primitive.

Insert the following new row to the untitled table defining the primitive parameters in 10.3.4.2.2:

Name	Type	Valid range	Description
Content of SAE Authentication Frame	Sequence of octets	As defined in 7.3.1.35, 7.3.1.36, 7.3.1.37, 7.3.1.38, 7.3.1.39, and 7.3.1.40	The contents of the SAE Commit Message or SAE Confirm Message. Present only if AuthenticationType indicates SAE authentication.

10.3.4.3 MLME-AUTHENTICATE.indication

10.3.4.3.2 Semantics of the service primitive

Change the primitive parameter list in 10.3.4.3.2 as follows:

The primitive parameters are as follows:

MLME-AUTHENTICATE.indication(
PeerSTAAddress,
AuthenticationType,
ResultCode,
Content of FT Authentication Elements,
Content of SAE Authentication Frame,
VendorSpecificInfo
)

Change the Authentication Type row of the untitled table defining the primitive parameters in 10.3.4.3.2 as follows:

Name	Type	Valid range	Description
AuthenticationType	Enumeration	OPEN_SYSTEM, SHARED_KEY, FAST_BSS_TRANSITION, SAE	Specifies the type of authentication algorithm that was used during the authentication process.

Insert the following new row to the untitled table defining the primitive parameters in 10.3.4.3.2:

Name	Type	Valid range	Description
Content of SAE Authentication Frame	Sequence of octets	As defined in 7.3.1.35, 7.3.1.36, 7.3.1.37, 7.3.1.38, 7.3.1.39, and 7.3.1.40	The contents of the SAE Commit Message or SAE Confirm Message. Present only if Authentication-Type indicates SAE authentication.

10.3.4.4 MLME-AUTHENTICATE.response

10.3.4.4.2 Semantics of the service primitive

Change the primitive parameter list in 10.3.4.4.2 as follows:

The primitive parameters are as follows:

MLME-AUTHENTICATE.response(
PeerSTAAddress,
AuthenticationType,
ResultCode,
Content of FT Authentication Elements,
Content of SAE Authentication Frame,
VendorSpecificInfo
)

Change the ResultCode row of the untitled table defining the primitive parameters in 10.3.4.4.2 as follows:

Name	Type	Valid range	Description
ResultCode	Enumeration	SUCCESS, REFUSED, ANTI-CLOGGING TOKEN REQUIRED, FINITE CYCLIC GROUP NOT SUPPORTED, AUTHENTICATION REJECTED	Indicates the result of the response to the authentication request from the peer MAC entity.

Insert the following new row to the untitled table defining the primitive parameters in 10.3.4.4.2:

Name	Type	Valid range	Description
Content of SAE Authentication Frame	Sequence of octets	As defined in 7.3.1.35, 7.3.1.36, 7.3.1.37, 7.3.1.38, 7.3.1.39, and 7.3.1.40	The contents of the SAE Commit Message or SAE Confirm Message. Present only if the AuthenticationType of the MLME-AUTHENTICATE.indication primitive that generated this response indicated SAE authentication.

10.3.5 Deauthenticate

10.3.5.1 MLME-DEAUTHENTICATE.request

10.3.5.1.3 When generated

Change 10.3.5.1.3 as follows:

This primitive is generated by the SME for a STA to invalidate authentication with a specified peer MAC entity in order to prevent the exchange of Class 2 frames, or Mesh Peering Management frames for AMPE utilizing SAE authentication [when dot11AuthenticationAlgorithm is simultaneousAuthEquals (4)], between the two STAs. During the deauthentication procedure, the SME can generate additional MLME-DEAUTHENTICATE.request primitives.

10.3.10 Start

Change 10.3.10 as follows:

This mechanism supports the process of creating a new BSS or becoming a member of an MBSS.

10.3.10.1 MLME-START.request

10.3.10.1.1 Function

Change 10.3.10.1.1 as follows:

This primitive requests that the MAC entity start a new BSS or become a member of an MBSS.

10.3.10.1.2 Semantics of the service primitive

Change the primitive parameter list in 10.3.10.1.2 as follows:

The primitive parameters are as follows:

MLME-START.request(

SSID,
BSSType,
BeaconPeriod,
DTIMPeriod,
CF parameter set,
PHY parameter set,
IBSS parameter set,
ProbeDelay,
CapabilityInformation,
BSSBasicRateSet,
OperationalRateSet,
Country,
IBSS DFS Recovery Interval,
EDCAParameterSet,
DSERegisteredLocation,
HT Capabilities,
HT Operation,
BSSMembershipSelectorSet,

BSSBasicMCSSet,
HTOperationalMCSSet,
Extended Capabilities,
20/40 BSS Coexistence,
Overlapping BSS Scan Parameters,
MultipleBSSID,
InterworkingInfo,
AdvertisementProtocolInfo,
RoamingConsortiumInfo,
Mesh ID,
Mesh Configuration,
VendorSpecificInfo
)

Change the BSSType, Beacon Period, and DTIM Period rows of the untitled table defining the primitive parameters in 10.3.10.1.2 as follows:

Name	Type	Valid range	Description
BSSType	Enumeration	INFRASTRUCTURE, INDEPENDENT, MESH	The Type of the BSS.
Beacon Period	Integer	≥1	The Beacon period (in TU) of the BSS ~~(in TU)~~ if the BSSType is not MESH, or of the mesh STA if the BSSType is MESH.
DTIM Period	Integer	As defined in 7.3.2.6	The DTIM Period (in beacon periods) of the BSS ~~(in beacon periods)~~ if the BSSType is not MESH, or of the mesh STA if the BSSType is MESH.

Insert the following new rows to the untitled table defining the primitive parameters in 10.3.10.1.2:

Name	Type	Valid range	Description
Mesh ID	Octet string	0–32 octets	The value of MeshID. This element is present only if the BSSType = MESH.
Mesh Configuration	As defined in frame format	As defined in 7.3.2.98	The values from the Mesh Configuration element. This element is present only if the BSSType = MESH.

10.3.10.1.3 When generated

Change the first and second paragraph in 10.3.10.1.3 as follows:

This primitive is generated by the SME to start either an infrastructure BSS (with the MAC entity within an AP), ~~or~~ an IBSS (with the MAC entity acting as the first STA in the IBSS), or an MBSS (with the MAC entity acting as the first mesh STA in the MBSS), or to become a member of an existing MBSS. In an MBSS, this primitive starts the process of mesh beaconing.

In an infrastructure BSS or an IBSS, t~~T~~he MLME-START.request primitive ~~must~~ shall be generated after an MLME-RESET.request primitive has been used to reset the MAC entity and before an MLME-JOIN.request primitive has been used to successfully join an existing infrastructure BSS or IBSS.

In a mesh BSS, the MLME-START.request primitive shall be generated after an MLME-RESET.request primitive has been used to reset the MAC entity and before any synchronization and mesh peering have been established. When the mesh STA uses the default synchronization method and the default mesh peering protocol, the MLME-START.request primitive shall be generated before an MLME-MESHNEIGHBOROFFSETSYNCSTART.request primitive and MLME-MESHPEERINGMANAGEMENT.request primitive have been used.

10.3.10.2 MLME-START.confirm

10.3.10.2.1 Function

Change 10.3.10.2.1 as follows:

This primitive reports the results of a BSS creation procedure, or a procedure becoming a member of an MBSS.

10.3.10.2.3 When generated

Change 10.3.10.2.3 as follows:

This primitive is generated by the MLME as a result of an MLME-START.request primitive to create a new BSS or to become a member of an MBSS.

10.3.10.2.4 Effect of receipt

Change 10.3.10.2.4 as follows:

The SME is notified of the results of the BSS creation procedure, or a procedure becoming a member of an MBSS.

10.3.15 Channel switch

10.3.15.1 MLME-CHANNELSWITCH.request

10.3.15.1.2 Semantics of the service primitive

Change the primitive parameter list in 10.3.15.1.2 as follows:

The primitive parameters are as follows:

```
MLME-CHANNELSWITCH.request(
                    Mode,
                    Channel Number,
                    Secondary Channel Offset,
                    Channel Switch Count,
                    Mesh Channel Switch Parameters,
                    VendorSpecificInfo
                    )
```

Change the row containing Channel Switch Count in the untitled table defining the primitive parameters in 10.3.15.1.2 as follows:

Name	Type	Valid range	Description
Channel Switch Count	As defined in 7.3.2.20 ~~Integer~~	As defined in 7.3.2.20 ~~0-255~~	Specifies the time period ~~number of TBTTs~~ until the channel switch event, as described in 7.3.2.20. ~~for the Channel Switch Announcement element~~

Insert the following new row to the untitled table defining the primitive parameters in 10.3.15.1.2:

Name	Type	Valid range	Description
Mesh Channel Switch Parameters	As defined in 7.3.2.103	As defined in 7.3.2.103	Specifies MBSS Channel Switch Parameters use by a mesh STA. This parameter is present if the dot11MeshActivated is true; otherwise, the parameter is not present.

10.3.15.1.4 Effect of receipt

Change 10.3.15.1.4 as follows:

On receipt of this primitive, the MLME schedules the channel switch event and announces this switch to other STAs in the BSS using the Channel Switch Announcement frame or element. A mesh STA shall include the Mesh Channel Switch Parameters element in the Beacon and Probe response frames if Beacon and Probe response frames contain Channel Switch Announcement element. The MLME ensures the timing of frame transmission takes into account the activation delay. The actual channel switch can be achieved at the appropriate time through the MLME-PLME interface using the PLME-SET primitive on dot11CurrentFrequency.

10.3.15.3 MLME-CHANNELSWITCH.indication

10.3.15.3.2 Semantics of the service primitive

Change the primitive parameter list in 10.3.15.3.2 as follows:

The primitive parameters are as follows:

```
MLME-CHANNELSWITCH.indication(
                              Peer MAC Address,
                              Mode,
                              Channel Number,
                              Secondary Channel Offset,
                              Channel Switch Count,
                              Mesh Channel Switch Parameters,
                              VendorSpecificInfo
                              )
```

Change the row containing Channel Switch Count in the untitled table defining the primitive parameters in 10.3.15.3.2 as follows:

Name	Type	Valid range	Description
Channel Switch Count	As defined in 7.3.2.20 ~~Integer~~	As defined in 7.3.2.20 ~~0-255~~	Specifies the time period ~~number of TBTTs~~ until the channel switch event, as described in 7.3.2.20. ~~for the Channel Switch Announcement element~~

Insert the following new row to the untitled table defining the primitive parameters in 10.3.15.3.2:

Name	Type	Valid range	Description
Mesh Channel Switch Parameters	As defined in 7.3.2.103	As defined in 7.3.2.103	Specifies MBSS Channel Switch Parameters use by a mesh STA. This parameter is present if the dot11MeshActivated is true; otherwise, the parameter is not present.

10.3.15.4 MLME-CHANNELSWITCH.response

10.3.15.4.2 Semantics of the service primitive

Change the primitive parameter list in 10.3.15.4.2 as follows:

The primitive parameters are as follows:

MLME-CHANNELSWITCH.response(
Mode,
Channel Number,
Secondary Channel Offset,
Channel Switch Count,
Mesh Channel Switch Parameters,
VendorSpecificInfo
)

Change the row containing Channel Switch Count in the untitled table defining the primitive parameters in 10.3.15.4.2 as follows:

Name	Type	Valid range	Description
Channel Switch Count	As defined in 7.3.2.20 ~~Integer~~	As defined in 7.3.2.20 ~~0-255~~	Specifies the time period ~~number of TBTTs~~ until the channel switch event, as described in 7.3.2.20. ~~for the Channel Switch Announcement element~~

Insert the following new row to the untitled table defining the primitive parameters in 10.3.15.4.2:

Name	Type	Valid range	Description
Mesh Channel Switch Parameters	As defined in 7.3.2.103	As defined in 7.3.2.103	Specifies MBSS Channel Switch Parameters use by a mesh STA. This parameter is present if the dot11MeshActivated is true; otherwise, the parameter is not present.

10.3.35 Extended channel switch announcement

10.3.35.1 MLME-EXTCHANNELSWITCH.request

10.3.35.1.1 Function

Change 10.3.35.1.1 as follows:

This primitive requests that a (Protected) Extended Channel Switch Announcement frame be sent by an AP or a mesh STA in an MBSS.

10.3.35.1.2 Semantics of the service primitive

Change the primitive parameter list in 10.3.35.1.2 as follows:

The primitive parameters are as follows:

MLME-EXTCHANNELSWITCH.request(

Mode,

RegulatoryClass,

Channel Number,

Channel Switch Count,

Protected,

Mesh Channel Switch Parameters,

VendorSpecificInfo

)

Change the row containing Channel Switch Count in the untitled table defining the primitive parameters in 10.3.35.1.2 as follows:

Name	Type	Valid range	Description
Channel Switch Count	As defined in 7.3.2.53 ~~Integer~~	As defined in 7.3.2.53 ~~0-255~~	Specifies the time period ~~number of TBTTs~~ until the channel switch event, as described in 7.3.2.53. ~~for the Extended Channel Switch Announcement element~~

Insert the following new row to the untitled table defining the primitive parameters in 10.3.35.1.2:

Name	Type	Valid range	Description
Mesh Channel Switch Parameters	As defined in 7.3.2.103	As defined in 7.3.2.103	Specifies MBSS Channel Switch Parameters use by a mesh STA. This parameter is present if the dot11MeshActivated is true; otherwise, the parameter is not present.

10.3.35.1.3 When generated

Change 10.3.35.1.3 as follows:

This primitive is generated by the STA management entity (SME) to request that a (Protected) Extended Channel Switch Announcement frame be sent to a STA that is associated to the AP or to peer mesh STAs in the MBSS.

10.3.35.3 MLME-EXTCHANNELSWITCH.indication

10.3.35.3.1 Function

Change 10.3.35.3.1 as follows:

This primitive indicates that a (Protected) Extended Channel Switch Announcement frame was received from an AP or from a peer mesh STA.

10.3.35.3.2 Semantics of the service primitive

Change the primitive parameter list in 10.3.35.3.2 as follows:

The primitive parameters are as follows:

```
MLME-EXTCHANNELSWITCH.indication(
                        Peer MAC Address,
                        Mode,
                        RegulatoryClass,
                        Channel Number,
                        Channel Switch Count,
                        Protection,
                        Mesh Channel Switch Parameters,
                        VendorSpecificInfo
                        )
```

Change the row containing Channel Switch Count in the untitled table defining the primitive parameters in 10.3.35.3.2 as follows:

Name	Type	Valid range	Description
Channel Switch Count	As defined in 7.3.2.53 ~~Integer~~	As defined in 7.3.2.53 ~~0-255~~	Specifies the time period ~~number of TBTTs~~ until the channel switch event, as described in 7.3.2.53. ~~for the Extended Channel Switch Announcement element~~

Insert the following new row to the untitled table defining the primitive parameters in 10.3.35.3.2:

Name	Type	Valid range	Description
Mesh Channel Switch Parameters	As defined in 7.3.2.103	As defined in 7.3.2.103	Specifies MBSS Channel Switch Parameters use by a mesh STA. This parameter is present if the dot11MeshActivated is true; otherwise, the parameter is not present.

10.3.35.4 MLME-EXTCHANNELSWITCH.response

10.3.35.4.2 Semantics of the service primitive

Change the primitive parameter list in 10.3.35.4.2 as follows:

The primitive parameters are as follows:

MLME-EXTCHANNELSWITCH.response(
Mode,
RegulatoryClass,
Channel Number,
Channel Switch Count,
Mesh Channel Switch Parameters,
VendorSpecificInfo
)

Change the row containing Channel Switch Count in the untitled table defining the primitive parameters in 10.3.35.4.2 as follows:

Name	Type	Valid range	Description
Channel Switch Count	As defined in 7.3.2.53 ~~Integer~~	As defined in 7.3.2.53 ~~0-255~~	Specifies the time period ~~number of TBTTs~~ until the channel switch event, as described in 7.3.2.53. ~~for the Extended Channel Switch Announcement element~~

Insert the following new row to the untitled table defining the primitive parameters in 10.3.35.4.2:

Name	Type	Valid range	Description
Mesh Channel Switch Parameters	As defined in 7.3.2.103	As defined in 7.3.2.103	Specifies MBSS Channel Switch Parameters use by a mesh STA. This parameter is present if the dot11MeshActivated is true; otherwise, the parameter is not present.

Insert the following new subclauses after 10.3.76:

10.3.77 Mesh peering management

10.3.77.1 Introduction

The following primitives facilitate the mesh peering management protocol and authenticated mesh peering exchange protocol.

10.3.77.2 MLME-MESHPEERINGMANAGEMENT.request

10.3.77.2.1 Function

This primitive requests that the MAC entity establish, confirm, or close a mesh peering with the specified peer MAC entity by sending a Mesh Peering Management frame to the peer MAC entity. The mesh peering management procedures are specified in 11C.3.

10.3.77.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MESHPEERINGMANAGEMENT.request(
                              PeerMACAddress,
                              MeshPeeringMgmtFrameContent
                              )
```

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Valid individual MAC address	Specifies the address of the peer MAC entity to which the Mesh Peering Management frame is to be sent.
MeshPeeringMgmtFrameContent	Sequence of octets	As defined in 7.4.14.2, 7.4.14.3, or 7.4.14.4	The contents of the Action field of the Mesh Peering Open, Mesh Peering Confirm, or Mesh Peering Close frame to send to the peer MAC entity.

10.3.77.2.3 When generated

This primitive is generated by the SME to request that a Mesh Peering Management frame be sent to the specified mesh STA.

10.3.77.2.4 Effect of receipt

On receipt of this primitive, the MLME constructs a Mesh Peering Management frame containing the information specified. The frame is scheduled for transmission.

10.3.77.3 MLME-MESHPEERINGMANAGEMENT.confirm

10.3.77.3.1 Function

This primitive reports the results of a request to send a Mesh Peering Management frame.

10.3.77.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MESHPEERINGMANAGEMENT.confirm(
                                PeerMACAddress,
                                ResultCode,
                                MeshPeeringMgmtFrameContent
                                )
```

Name	Type	Valid range	Description
Peer-MACAd-dress	MAC Address	Valid individual MAC address	Specifies the address of the peer MAC entity to which the Mesh Peering Management frame was sent.
ResultCode	Enumeration	SUCCESS, TIMEOUT, INVALID_PARAMETERS, or UNSPECIFIED_FAILURE	Reports the outcome of the request to send a Mesh Peering Management frame.
MeshPeer-ingMgmtFr-ameContent	Sequence of octets	As defined in 7.4.14.2, 7.4.14.3, or 7.4.14.4	The contents of the Action field of the Mesh Peering Open, Mesh Peering Confirm, or Mesh Peering Close frame received from the peer MAC entity.

10.3.77.3.3 When generated

This primitive is generated as a result of an MLME-MESHPEERINGMANAGEMENT.request with a specified MAC peer.

10.3.77.3.4 Effect of receipt

The SME is notified of the results of the mesh peering management protocol request.

10.3.77.4 MLME-MESHPEERINGMANAGEMENT.indication

10.3.77.4.1 Function

This primitive indicates to the SME that the MLME has received a Mesh Peering Management frame from a peer MAC entity.

10.3.77.4.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MESHPEERINGMANAGEMENT.indication(
                              PeerMACAddress,
                              MeshPeeringMgmtFrameContent
                              )
```

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Valid individual MAC address	Specifies the address of the peer MAC entity from which the Mesh Peering Management frame was received.
MeshPeeringMg-mtFrameContent	Sequence of octets	As defined in 7.4.14.2, 7.4.14.3, or 7.4.14.4	The contents of the Action field of the Mesh Peering Open, Mesh Peering Confirm, or Mesh Peering Close frame received from the peer MAC entity.

10.3.77.4.3 When generated

This primitive is generated by the MLME as a result of the receipt of a Mesh Peering Management frame from a peer MAC entity.

10.3.77.4.4 Effect of receipt

The SME is notified of the reception of a Mesh Peering Management frame, and is provided the contents of the frame.

10.3.77.5 MLME-MESHPEERINGMANAGEMENT.response

10.3.77.5.1 Function

This primitive is used to send a response to a Mesh Peering Management frame to the specified peer MAC entity.

10.3.77.5.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MESHPEERINGMANAGEMENT.response(
                              PeerMACAddress,
                              ResultCode,
                              MeshPeeringMgmtFrameContent
                              )
```

10.3.77.5.3 When generated

This primitive is generated by the SME as a response to an MLME-MESHPEERINGMANAGEMENT.indication primitive.

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Valid individual MAC address	Specifies the address of the peer MAC entity to which the Mesh Peering Management frame is to be sent.
ResultCode	Enumeration	SUCCESS, INVALID_PARAMETERS, or UNSPECIFIED_FAILURE	Reports the result response to the Mesh Peering Management frame from the peer MAC entity.
MeshPeeringMgmtFrameContent	Sequence of octets	As defined in 7.4.14.2, 7.4.14.3, or 7.4.14.4	The contents of the Action field of the Mesh Peering Open, Mesh Peering Confirm, or Mesh Peering Close frame to send to the peer MAC entity.

10.3.77.5.4 Effect of receipt

This primitive indicates scheduling for transmission of a Mesh Peering management frame containing the indicated response.

10.3.78 Mesh power management

10.3.78.1 Introduction

The following primitives describe how a mesh entity changes its mesh power mode for a mesh peering.

10.3.78.2 MLME-MESHPOWERMGT.request

10.3.78.2.1 Function

This primitive requests a change in the mesh STAs mesh power mode for the mesh peering.

10.3.78.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MESHPOWERMGT.request(
                              PeerMACAddress,
                              Mesh Power Mode
                              )
```

Name	Type	Valid Range	Description
PeerMACAddress	MAC Address	Valid individual MAC address	Specifies the address of the peer MAC entity to which the mesh power mode is changed.
Mesh Power Mode	Enumeration	ACTIVE_MODE, LIGHT_SLEEP_MODE, DEEP_SLEEP_MODE	Specifies the mesh power mode that the local mesh STA is using for the mesh peering.

10.3.78.2.3 When generated

The primitive is generated when the mesh entity wishes to change its mesh power mode for a mesh peering.

10.3.78.2.4 Effect of receipt

This primitive initiates the local mesh STA's mesh power mode change for the mesh peering. The MLME subsequently issues an MLME-MESHPOWERMGT.confirm that reflects the results.

10.3.78.3 MLME-MESHPOWERMGT.confirm

10.3.78.3.1 Function

This primitive reports the result of a mesh power mode change attempt.

10.3.78.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MESHPOWERMGT.confirm(
                                PeerMACAddress,
                                ResultCode
                                )
```

Name	Type	Valid Range	Description
PeerMACAddress	MAC Address	Valid individual MAC address	Specifies the address of the peer MAC entity to which the mesh power mode is changed.
ResultCode	Enumeration	SUCCESS, INVALID_PARAMETERS, NOT_SUPPORTED	Indicates the result of the MLME-MESHPOWERMGT.request.

10.3.78.3.3 When generated

This primitive is generated as a result of an MLME-MESHPOWERMGT.request.

10.3.78.3.4 Effect of receipt

The SME is notified of the results of the mesh power mode change for a mesh peering procedure.

10.3.79 Mesh neighbor offset synchronization

10.3.79.1 Introduction

This mechanism manages the neighbor offset synchronization method with the specified neighbor STA.

10.3.79.2 MLME-MESHNEIGHBOROFFSETSYNCSTART.request

10.3.79.2.1 Function

This primitive requests to start the neighbor offset synchronization method with the specified neighbor STA.

10.3.79.2.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-MESHNEIGHBOROFFSETSYNCSTART.request(
PeerMACAddress
)

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Any valid individual MAC address	Specifies the address of the peer MAC entity with which to start the neighbor offset synchronization method.

10.3.79.2.3 When generated

This primitive is generated by the SME to start the neighbor offset synchronization method with the specified neighbor STA.

10.3.79.2.4 Effect of receipt

On receipt of this primitive, the MLME commences the neighbor offset synchronization method and the calculation of the TSF timer offset value. The MLME subsequently issues an MLME-MESHNEIGHBOROFFSETSYNCSTART.confirm that reflects the results of this request.

10.3.79.3 MLME-MESHNEIGHBOROFFSETSYNCSTART.confirm

10.3.79.3.1 Function

This primitive reports the results of a mesh neighbor offset synchronization request.

10.3.79.3.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-MESHNEIGHBOROFFSETSYNCSTART.confirm(
PeerMACAddress,
ResultCode,
TSFOffsetValue
)

10.3.79.3.3 When generated

This primitive is generated by the MLME as a result of an MLME-MESHNEIGHBOROFFSETSYNCSTART.request and report the TSF offset value.

10.3.79.3.4 Effect of receipt

The SME is notified of the results of the mesh neighbor offset synchronization request.

10.3.79.4 MLME-MESHNEIGHBOROFFSETCALCULATE.request

10.3.79.4.1 Function

This primitive requests a calculation result of the TSF timer offset value for the specified neighbor STA.

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Valid individual MAC address	Specifies the address of the peer MAC entity to which the neighbor offset synchronization is requested.
ResultCode	Enumeration	SUCCESS, INVALID_PARAMETERS, TOO_MANY_NEIGHBORS, or NOT_SUPPORTED	Indicates the result of the mesh neighbor offset synchronization request.
TSFOffsetValue	Integer	$–2^{63}$ to $(2^{63}–1)$	Indicates the TSF offset value with the specified neighbor STA, expressed in twos complement in µs. Valid only if the ResultCode is SUCCESS.

10.3.79.4.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MESHNEIGHBOROFFSETCALCULATE.request(
                                PeerMACAddress
                                )
```

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Any valid individual MAC address	Specifies the address of the peer MAC entity with which to report the TSF offset value.

10.3.79.4.3 When generated

This primitive is generated by the SME to order a calculation of the TSF timer offset value with the specified neighbor STA.

10.3.79.4.4 Effect of receipt

On receipt of this primitive, the MLME receives a Beacon or Probe Response frame and calculates the TSF timer offset value from the received frame. The MLME tries to receive a Beacon frame immediately after the issue of MLME-MESHNEIGHBOROFFSETCALCULATE.request even if the mesh STA does not listen to the Beacon frame from the specified neighbor STA regularly (i.e., in deep sleep mode toward the specified neighbor STA). The MLME subsequently issues an MLME-MESHNEIGHBOROFFSETCALCULATE.confirm that reflects the results of this request.

10.3.79.5 MLME-MESHNEIGHBOROFFSETCALCULATE.confirm

10.3.79.5.1 Function

This primitive reports the results of a mesh neighbor offset calculation request.

10.3.79.5.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MESHNEIGHBOROFFSETCALCULATE.confirm(
                              PeerMACAddress,
                              ResultCode,
                              TSFOffsetValue
                              )
```

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Valid individual MAC address	Specifies the address of the peer MAC entity to which the Neighbor Offset Measure is requested.
ResultCode	Enumeration	SUCCESS, TIMEOUT, INVALID_PARAMETERS, or NOT_SUPPORTED	Indicates the result of the mesh neighbor offset calculation request.
TSFOffsetValue	Integer	-2^{63} to $(2^{63}-1)$	Indicates the TSF offset value with the specified neighbor STA, expressed in twos complement in µs. Valid only if the ResultCode is SUCCESS.

10.3.79.5.3 When generated

This primitive is generated by the MLME as a result of an MLME-MESHNEIGHBOROFFSETCALCULATE.request to report a TSF offset value.

10.3.79.5.4 Effect of receipt

The SME is notified of the results of the mesh neighbor offset calculation request.

10.3.79.6 MLME-MESHNEIGHBOROFFSETSYNCSTOP.request

10.3.79.6.1 Function

This primitive requests to stop the neighbor offset synchronization method with the specified neighbor STA.

10.3.79.6.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MESHNEIGHBOROFFSETSYNCSTOP.request(
                              PeerMACAddress
                              )
```

10.3.79.6.3 When generated

This primitive is generated by the SME to stop the maintenance of the neighbor offset synchronization method with the specified neighbor STA.

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Any valid individual MAC address	Specifies the address of the peer MAC entity with which to stop the neighbor offset synchronization method.

10.3.79.6.4 Effect of receipt

On receipt of this primitive, the MLME stops the neighbor offset synchronization method with the specified peer. The MLME subsequently issues an MLME-MESHNEIGHBOROFFSETSYNCSTOP.confirm that reflects the results of this request.

10.3.79.7 MLME-MESHNEIGHBOROFFSETSYNCSTOP.confirm

10.3.79.7.1 Function

This primitive reports the results of a neighbor offset synchronization method stop request.

10.3.79.7.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-MESHNEIGHBOROFFSETSYNCSTOP.confirm(
PeerMACAddress,
ResultCode,
)

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Valid individual MAC address	Specifies the address of the peer MAC entity to which the Neighbor Offset Stop is requested.
ResultCode	Enumeration	SUCCESS, INVALID_PARAMETERS, or NOT_SUPPORTED	Indicates the result of the mesh neighbor offset synchronization stop request.

10.3.79.7.3 When generated

This primitive is generated by the MLME as a result of an MLME-MESHNEIGHBOROFFSETSYNCSTOP.request.

10.3.79.7.4 Effect of receipt

The SME is notified of the results of the mesh neighbor offset synchronization stop request.

10.3.80 Mesh TBTT adjustment

10.3.80.1 Introduction

The following primitives describe how a mesh STA requests a TBTT adjustment from a neighboring peer mesh STA.

10.3.80.2 MLME-MESHTBTTADJUSTMENT.request

10.3.80.2.1 Function

This primitive requests transmission of a TBTT Adjustment Request frame.

10.3.80.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MESHTBTTADJUSTMENT.request(
                    PeerMACAddress,
                    BeaconTiming,
                    VendorSpecificInfo
                    )
```

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Any valid individual MAC address	Specifies the address of the peer MAC entity to which the TBTT Adjustment Request is sent.
BeaconTiming	A set of Beacon Timing elements	As defined in 7.3.2.105	A set of Beacon Timing elements of the mesh STA.
VendorSpecificInfo	A set of elements	As defined in 7.3.2.26	Zero or more elements.

10.3.80.2.3 When generated

This primitive is generated by the SME to request that a TBTT Adjustment Request frame be sent to a peer entity to request the adjustment of the peer entity's TBTT.

10.3.80.2.4 Effect of receipt

On receipt of this primitive, the MLME constructs a TBTT Adjustment Request frame containing the Beacon Timing elements. This frame is then scheduled for transmission. The MLME subsequently issues an MLME-MESHTBTTADJUSTMENT.confirm that reflects the result of this request.

10.3.80.3 MLME-MESHTBTTADJUSTMENT.confirm

10.3.80.3.1 Function

This primitive reports the result of a mesh TBTT adjustment request.

10.3.80.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MESHTBTTADJUSTMENT.confirm(
                        PeerMACAddress,
                        ResultCode,
                        BeaconTiming,
                        VendorSpecificInfo
                        )
```

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Any valid individual MAC address	Specifies the address of the peer MAC entity from which the TBTT Adjustment Response is received.
ResultCode	Enumeration	SUCCESS, INVALID_PARAMETERS, TIMEOUT, CAN_NOT_FIND_ALTERNATIVE_TBTT, or UNSPECIFIED_FAILURE	Indicates the result of the TBTT adjustment request.
BeaconTiming	A set of Beacon Timing elements	As defined in 7.3.2.105	A set of Beacon Timing elements of the responding mesh STA. Present only when such an element was present in the TBTT Adjustment Response frame.
VendorSpecificInfo	A set of elements	As defined in 7.3.2.26	Zero or more elements.

10.3.80.3.3 When generated

This primitive is generated by the MLME as a result of an MLME-MESHTBTTADJUSTMENT.request primitive to indicate the result of that request.

10.3.80.3.4 Effect of receipt

The SME is notified of the result of the mesh TBTT adjustment request.

10.3.80.4 MLME-MESHTBTTADJUSTMENT.indication

10.3.80.4.1 Function

This primitive indicates that a specific peer MAC entity is requesting adjustment of the TBTT.

10.3.80.4.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MESHTBTTADJUSTMENT.indication(
                                   PeerMACAddress,
                                   BeaconTiming,
                                   VendorSpecificInfo
                                   )
```

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Any valid individual MAC address	Specifies the address of the peer MAC entity from which the TBTT Adjustment request was received.
BeaconTiming	A set of Beacon Timing elements	As defined in 7.3.2.105	A set of Beacon Timing elements of the requesting mesh STA.
VendorSpecificInfo	A set of elements	As defined in 7.3.2.26	Zero or more elements.

10.3.80.4.3 When generated

This primitive is generated by the MLME as a result of the receipt of a TBTT Adjustment Request frame from the specified peer MAC entity.

10.3.80.4.4 Effect of receipt

The SME is notified of the receipt of the TBTT adjustment request by the specified peer MAC entity. The mesh STA that received this primitive subsequently processes the TBTT scanning and adjustment procedure described in 11C.12.4.4.3, and responds with the MLME-MESHTBTTADJUSTMENT.response.

10.3.80.5 MLME-MESHTBTTADJUSTMENT.response

10.3.80.5.1 Function

This primitive is used to send a response to the specified peer MAC entity that requested a TBTT adjustment from the mesh STA that issued this primitive.

10.3.80.5.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MESHTBTTADJUSTMENT.response(
                                   PeerMACAddress,
                                   Status Code,
                                   BeaconTiming,
                                   VendorSpecificInfo
                                   )
```

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Any valid individual MAC address	Specifies the address of the peer MAC entity to which the TBTT Adjustment Response is sent.
Status Code	As defined in frame format	As defined in frame format	Indicates the result response to the TBTT adjustment request from the peer mesh STA.
BeaconTiming	A set of Beacon Timing elements	As defined in 7.3.2.105	A set of Beacon Timing elements of the mesh STA. Present only when Status Code is set to 78.
VendorSpecificInfo	A set of elements	As defined in 7.3.2.26	Zero or more elements.

10.3.80.5.3 When generated

This primitive is generated by the SME of a mesh STA as response to an MLME-MESHTBTTADJUSTMENT.indication primitive.

10.3.80.5.4 Effect of receipt

This primitive initiates the transmission of a TBTT Adjust Response frame to the peer MAC entity that requested the TBTT adjustment.

On receipt of this primitive, the MLME constructs a TBTT Adjustment Response frame. When the Status Code is set to 78, the frame contains the Beacon Timing element. This frame is then scheduled for transmission.

10.3.81 MCCA management interface

10.3.81.1 Introduction

The following primitives describe how a mesh entity manages its MCCA operation.

10.3.81.2 MLME-ACTIVATEMCCA.request

10.3.81.2.1 Function

This primitive requests that the MAC entity activate MCCA.

10.3.81.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-ACTIVATEMCCA.request(
                                MCCAScanDuration,
                                MAFLimit,
                                MCCAAdvertPeriodMax,
                                MCCAMaxTrackStates,
```

MCCACWmin,
MCCACWmax,
MCCAAIFSN
)

Name	Type	Valid range	Description
MCCAScan-Duration	Integer	0–65535	Specifies the duration in TUs that the mesh STA shall not initiate or accept MCCA Setup Request frames.
MAFLimit	Integer	0–255	Specifies the maximum MCCA access fraction allowed at the mesh STA. This number is always a multiple of (1/255) of the DTIM Interval.
MCCAAdvert-PeriodMax	Integer	0–255	Specifies the maximum interval that a mesh STA with dot11MCCAActivated equal to true waits for an MCCAOP advertisement. It is expressed in number of DTIM intervals.
MCCAMax-TrackStates	Integer	dot11MCCAMinTrackStates –65535	Specifies the total number of MCCAOP reservations that the MAC entity is able to track.
MCCACWmin	Integer	0–15	Specifies the value of the minimum size of the contention window that the MAC entity uses for channel access during an MCCAOP.
MCCACW-max	Integer	0–63	Specifies the value of the maximum size of the contention that the MAC entity uses for channel access during an MCCAOP.
MCCAAIFSN	Integer	0–15	Specifies the value of the AIFSN that the MAC entity uses for channel access during an MCCAOP.

10.3.81.2.3 When generated

This primitive is generated by the SME to start the use of MCCA.

10.3.81.2.4 Effect of receipt

This primitive sets dot11MCCAEnabled to true and initializes the MCCA parameters. The MLME subsequently issues an MLME-ACTIVATEMCCA.confirm primitive that reflects the results.

10.3.81.3 MLME-ACTIVATEMCCA.confirm

10.3.81.3.1 Function

This primitive reports the result of an MLME-MCCAACTIVATE.request.

10.3.81.3.2 Semantics of the service primitive

The primitive parameters are as follows:
MLME-ACTIVATEMCCA.confirm(
ResultCode

)

Name	Type	Valid Range	Description
ResultCode	Enumeration	SUCCESS, INVALID_PARAMETERS, MCCA_NOT_IMPLEMENTED, INVALID_MAF_LIMIT, INVALID_MCCA_ADVERT_PERIOD_MAX, INVALID_MAXTRACKSTATES, INVALID_MCCACWMIN, INVALID_MCCACWMAX, INVALID_MCCAAIFSN	Indicates the result of the MLME-ACTIVATEMCCA.request.

10.3.81.3.3 When generated

This primitive is generated as a result of an MLME-ACTIVATEMCCA.request.

10.3.81.3.4 Effect of receipt

The SME is notified of the results of the request to start the use of MCCA.

10.3.81.4 MLME-MCCASETUP.request

10.3.81.4.1 Function

This primitive requests that the MAC entity set up an MCCAOP reservation.

10.3.81.4.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-MCCASETUP.request(
MCCAOPDuration,
MCCAOPPeriodicity,
MCCAOPOffset,
MCCAOPResponder,
VendorSpecificInfo
)

10.3.81.4.3 When generated

This primitive is generated by the SME to start an MCCAOP setup procedure.

10.3.81.4.4 Effect of receipt

This primitive causes the transmission of an MCCA Setup Request frame to the MCCAOP responder provided that the conditions for the transmission are met. The MLME subsequently issues an MLME-MCCASETUP.confirm primitive that reflects the results.

Name	Type	Valid range	Description
MCCAOPDuration	Integer	0–65535	Specifies the MCCAOP Duration of the needed MCCAOPs as described in 7.3.2.106.2.
MCCAOPPeriodicity	Integer	0–255	Specifies the MCCAOP Periodicity of the needed MCCAOPs as described in 7.3.2.106.2.
MCCAOPOffset	Integer	0 – 16 777 215	Specifies the MCCAOP offset of the needed MCCAOPs as described in 7.3.2.106.2.
MCCAOPResponder	MAC address	Any valid individual or group MAC address	Specifies the MAC address of the intended MCCAOP responder.
VendorSpecificInfo	A set of elements	As defined in 7.3.2.26	Zero or more elements.

10.3.81.5 MLME-MCCASETUP.confirm

10.3.81.5.1 Function

This primitive is generated by the MLME to report the result of an MLME-MCCASETUP.request primitive, which was issued in order to establish an MCCAOP reservation with the peer MAC entity specified in MCCAOPResponder.

10.3.81.5.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MCCASETUP.confirm(
                                MCCAOPParameters,
                                MCCAOPID,
                                MCCAOPResponder,
                                ResultCode,
                                VendorSpecificInfo
                                )
```

10.3.81.5.3 When generated

This primitive is generated by the MLME as a result of an MLME-MCCASETUP.request to establish an MCCAOP reservation with the peer mesh STA identified in MCCAOPResponder or upon receipt of an MCCA Setup Reply frame from the peer mesh STA identified in MCCAOPResponder.

10.3.81.5.4 Effect of receipt

The SME is notified of the results of the MCCAOP setup procedure.

Name	Type	Valid range	Description
MCCAOPParameters	MCCAOP Reservation	See 7.3.2.106.2	The MCCAOP reservation parameters.
MCCAOPID	Integer	0–254	MCCAOP reservation ID of the MCCAOP reservation.
MCCAOPResponder	MAC address	Any valid individual or group MAC address	Specifies the MAC address of the intended MCCAOP responder.
ResultCode	Enumeration	SUCCESS, INVALID_PARAMETERS, MCCAOP_RESERVATION_CONFLICT, MAF_LIMIT_EXCEEDED, MCCA_TRACK_LIMIT_EXCEEDED, MCCA_SETUP_TIMEOUT	Indicates the result of the MLME-MCCASETUP.request.
VendorSpecificInfo	A set of elements	As defined in 7.3.2.26	Zero or more elements.

10.3.81.6 MLME-MCCASETUP.indication

10.3.81.6.1 Function

This primitive indicates the receipt of an MCCA Setup Request frame from the peer MAC entity specified in MCCAOPOwner.

10.3.81.6.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MCCASETUP.indication(
                              MCCAOPParameters,
                              MCCAOPID,
                              MCCAOPOwner,
                              VendorSpecificInfo
                              )
```

Name	Type	Valid range	Description
MCCAOPParameters	MCCAOP Reservation	See 7.3.2.106.2	The MCCAOP reservation parameters.
MCCAOPID	Integer	0–254	MCCAOP reservation ID of the MCCAOP reservation.
MCCAOPOwner	MAC address	Any valid individual MAC address	Specifies the MAC address of the MCCAOP owner.
VendorSpecificInfo	A set of elements	As defined in 7.3.2.26	Zero or more elements.

10.3.81.6.3 When generated

This primitive is generated by the MLME as result of the receipt of an MCCA Setup Request frame from the peer MAC entity specified in MCCAOPOwner.

10.3.81.6.4 Effect of receipt

The SME is notified of the request to establish an MCCAOP reservation with the peer MAC entity specified in MCCAOPOwner.

10.3.81.7 MLME-MCCASETUP.response

10.3.81.7.1 Function

This primitive is used to send a response to the peer MAC entity specified in MCCAOPOwner that requested the set up of the MCCAOP reservation.

10.3.81.7.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MCCASETUP.response(
                        MCCAOPParameters,
                        MCCAOPID,
                        MCCAOPOwner,
                        ResultCode,
                        VendorSpecificInfo
                        )
```

Name	Type	Valid range	Description
MCCAOPParameters	MCCAOP Reservation	See 7.3.2.106.2	The MCCAOP reservation parameters.
MCCAOPID	Integer	0–254	MCCAOP reservation ID of the MCCAOP reservation.
MCCAOPOwner	MAC address	Any valid individual MAC address	Specifies the MAC address of the MCCAOP owner.
ResultCode	Enumeration	SUCCESS, INVALID_PARAMETERS, MCCAOP_RESERVATION_CONFLICT, MAF_LIMIT_EXCEEDED, MCCA_TRACK_LIMIT_EXCEEDED	Indicates the result of the MLME-MCCASETUP.request.
VendorSpecificInfo	A set of elements	As defined in 7.3.2.26	Zero or more elements.

10.3.81.7.3 When generated

This primitive is generated by the SME of a STA as a response to an MLME-MCCASETUP.indication procedure.

10.3.81.7.4 Effect of receipt

This primitive initiates transmission of a response to the peer MAC entity specified in the MCCAOPOwner that requested the set up of an MCCAOP reservation.

10.3.81.8 MLME-MCCAADVERTISEMENT.request

10.3.81.8.1 Function

This primitive requests that the MAC entity request an MCCAOP advertisement from the specified peer MAC entity by sending an MCCA Advertisement Request frame to the peer MAC entity.

10.3.81.8.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MCCAADVERTISEMENT.request(
                PeerMACAddress,
                VendorSpecificInfo
                )
```

Name	Type	Valid range	Description
PeerMACAddress	MAC address	Any valid individual MAC address	Specifies the MAC address of the peer MAC that will send the Advertisement.
VendorSpecificInfo	A set of elements	As defined in 7.3.2.26	Zero or more elements.

10.3.81.8.3 When generated

This primitive is generated by the SME to request an MCCAOP advertisement from the specified peer MAC entity.

10.3.81.8.4 Effect of receipt

This primitive causes the transmission of an MCCA Advertisement Request frame to the specified peer MAC entity. The MLME subsequently issues an MLME-MCCAADVERTISEMENT.confirm primitive that reflects the results.

10.3.81.9 MLME-MCCAADVERTISEMENT.confirm

10.3.81.9.1 Function

This primitive reports the result of an MLME-MCCAADVERTISEMENT.request.

10.3.81.9.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MCCAADVERTISEMENT.confirm(
                                MCCAOPAdvertisement,
                                PeerMACAddress,
                                ResultCode,
                                VendorSpecificInfo
                                )
```

Name	Type	Valid range	Description
MCCAOPAdvertisement	MCCAOP Advertisement	See 7.3.2.109	One or more MCCAOP Advertisement elements.
PeerMACAddress	MAC address	Any valid individual MAC address	Specifies the MAC address of the transmitter of the MCCAOP advertisement.
ResultCode	Enumeration	SUCCESS, INVALID_ADVERTISEMENT, ADVERTISEMENT_TIMEOUT	Indicates the result of the MLME-MCCAADVERTISEMENT.request.

Name	Type	Valid range	Description
VendorSpecificInfo	A set of elements	As defined in 7.3.2.26	Zero or more elements.

10.3.81.9.3 When generated

This primitive is generated by the MLME as a result of an MLME-MCCAADVERTISEMENT.request.

10.3.81.9.4 Effect of receipt

The SME is notified of the results of the MCCA Advertisement Request frame.

10.3.81.10 MLME-MCCAADVERTISEMENT.indication

10.3.81.10.1 Function

This primitive reports that an MCCA Advertisement Request frame has been received from the specified peer MAC entity.

10.3.81.10.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MCCAADVERTISEMENT.indication(
                                  PeerMACAddress,
                                  VendorSpecificInfo
                                  )
```

Name	Type	Valid range	Description
PeerMACAddress	MAC address	Any valid individual MAC address	Specifies the MAC address of the transmitter of the MCCA Advertisement Request frame.
VendorSpecificInfo	A set of elements	As defined in 7.3.2.26	Zero or more elements.

10.3.81.10.3 When generated

This primitive is generated by the MLME upon receipt of an MCCA Advertisement Request frame from the specified peer MAC entity.

10.3.81.10.4 Effect of receipt

The SME is notified of the request to advertise its MCCAOP reservations.

10.3.81.11 MLME-MCCAADVERTISEMENT.response

10.3.81.11.1 Function

This primitive requests that the MAC entity respond to the MCCAOP advertisement request from the specified peer MAC entity by sending an MCCA Advertisement frame to the peer MAC entity.

10.3.81.11.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MCCAADVERTISEMENT.response(
                              MCCAOPAdvertisement,
                              PeerMACAddress,
                              ResultCode,
                              VendorSpecificInfo
                              )
```

Name	Type	Valid range	Description
MCCAOPAdvertisement	MCCAOP Advertisement	See 7.3.2.109	One or more MCCAOP Advertisement elements.
PeerMACAddress	MAC address	Any valid individual MAC address	Specifies the MAC address of the transmitter of the MCCA Advertisement frame.
ResultCode	Enumeration	SUCCESS, REFUSED	Indicates the result of the MLME-MCCAADVERTISEMENT.response.
VendorSpecificInfo	A set of elements	As defined in 7.3.2.26	Zero or more elements.

10.3.81.11.3 When generated

This primitive is generated by the SME of a STA as a response to an MLME-MCCAADVERTISEMENT.indication procedure.

10.3.81.11.4 Effect of receipt

This primitive initiates transmission of a response to the specified peer MAC entity that requested advertisement of the MCCAOP reservations.

10.3.81.12 MLME-MCCATEARDOWN.request

10.3.81.12.1 Function

This primitive requests that the MAC entity tear down an MCCAOP reservation.

10.3.81.12.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-MCCATEARDOWN.request(
MCCAOPID,
PeerMACAddress
)

Name	Type	Valid range	Description
MCCAOPID	Integer	0–255	Specifies the MCCAOP reservation ID of the MCCAOP reservation to be torn down.
PeerMACAddress	MAC address	Any valid individual MAC address	Specifies the MAC address of the peer MAC for the MCCAOP reservation.

10.3.81.12.3 When generated

This primitive is generated by the SME to start an MCCAOP teardown procedure.

10.3.81.12.4 Effect of receipt

This primitive causes the teardown MCCAOP reservation identified by means of the MCCAOP reservation ID in MCCAOPID, and the transmission of an MCCA Teardown frame to the peer MAC entity in PeerMACAddress. The MLME subsequently issues an MLME-MCCATEARDOWN.confirm primitive that reflects the results.

10.3.81.13 MLME-MCCATEARDOWN.confirm

10.3.81.13.1 Function

This primitive reports the result of an MLME-MCCATEARDOWN.request.

10.3.81.13.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-MCCATEARDOWN.confirm(
MCCAOPID,
PeerMACAddress,
ResultCode

)

Name	Type	Valid range	Description
MCCAOPID	Integer	0–255	MCCAOP reservation ID of the MCCAOP reservation to be torn down.
PeerMACAddress	MAC address	Any valid individual MAC address	Specifies the MAC address of the peer MAC for the MCCAOP reservation.
ResultCode	Enumeration	SUCCESS, INVALID_PARAMETERS, INVALID_PEER_MAC, INVALID_MCCAOPID	Indicates the result of the MLME-MCCATEARDOWN.request.

10.3.81.13.3 When generated

This primitive is generated by the MLME as a result of an MLME-MCCATEARDOWN.request.

10.3.81.13.4 Effect of receipt

The SME is notified of the results of the request to start an MCCAOP teardown procedure.

10.3.81.14 MLME-MCCATEARDOWN.indication

10.3.81.14.1 Function

This primitive reports that an MCCA Teardown frame has been received from the specified peer MAC entity.

10.3.81.14.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MCCATEARDOWN.indication(
                              MCCAOPID,
                              PeerMACAddress
                              )
```

Name	Type	Valid range	Description
MCCAOPID	Integer	0–255	MCCAOP reservation ID of the MCCAOP reservation to be torn down.
PeerMACAddress	MAC address	Any valid individual MAC address	Specifies the MAC address of the peer MAC for the MCCAOP reservation.

10.3.81.14.3 When generated

This primitive is generated by the MLME as result of receipt of a MCCA Teardown frame.

10.3.81.14.4 Effect of receipt

The SME is notified of the request to start an MCCAOP teardown procedure.

10.3.82 MBSS congestion control

10.3.82.1 Introduction

The following primitives describe how a mesh STA manages its congestion control operation.

10.3.82.2 MLME-MBSSCONGESTIONCONTROL.request

10.3.82.2.1 Function

This primitive requests that the MAC entity notify the peer MAC entity on the congestion level or requests to traffic generation by transmitting a Congestion Control Notification frame to the specified peer MAC entity.

10.3.82.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME- MBSSCONGESTIONCONTROL.request(
                                    PeerMACAddress,
                                    CongestionNotification,
                                    VendorSpecificInfo
                                    )
```

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Any valid individual MAC address	Specifies the address of the peer MAC entity to which the Congestion Control Notification frame is sent.
CongestionNotification	A set of Congestion Notification elements	As defined in 7.3.2.101	Congestion notification information generated by the mesh STA.
VendorSpecificInfo	A set of elements	As defined in 7.3.2.26	Zero or more elements.

10.3.82.2.3 When generated

This primitive is generated by the SME to request that the MAC entity notifies the peer MAC entity on the congestion level or requests to traffic generation rate.

10.3.82.2.4 Effect of receipt

On receipt of this primitive, the MLME constructs a Congestion Control Notification frame. This frame is then scheduled for transmission. The MLME subsequently issues an MLME-MBSSCONGESTIONCONTROL.confirm that reflects the results of this request.

10.3.82.3 MLME-MBSSCONGESTIONCONTROL.confirm

10.3.82.3.1 Function

This primitive reports the results of a congestion control notification request.

10.3.82.3.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME- MBSSCONGESTIONCONTROL.confirm(
ResultCode
)

Name	Type	Valid range	Description
Result Code	Enumeration	SUCCESS, INVALID_PARAMETERS, NOT_SUPPORTED	Indicates the result of the MLME-MBSSCONGESTIONCONTROL request.

10.3.82.3.3 When generated

This primitive is generated by the MLME as a result of an MLME-MBSSCONGESTIONCONTROL.request primitive to indicate the result of that request.

10.3.82.3.4 Effect of receipt

The SME is notified of the results of the congestion control notification request.

10.3.82.4 MLME-MBSSCONGESTIONCONTROL.indication

10.3.82.4.1 Function

This primitive indicates that a congestion notification has been received.

10.3.82.4.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME- MBSSCONGESTIONCONTROL.indication(
PeerMACAddress,
CongestionNotification,
VendorSpecificInfo

)

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Any valid individual MAC address	Specifies the address of the peer MAC entity from which the Congestion Control Notification frame was received.
CongestionNotification	A set of Congestion Notification elements	As defined in 7.3.2.101	Congestion notification information contained in the received Congestion Control Notification frame.
VendorSpecificInfo	A set of elements	As defined in 7.3.2.26	Zero or more elements.

10.3.82.4.3 When generated

This primitive is generated by the MLME as a result of the receipt of a Congestion Control Notification frame from a specific peer MAC entity.

10.3.82.4.4 Effect of receipt

The SME is notified of the results of the receipt of the congestion control notification from the specified peer MAC entity. The mesh STA that received this primitive subsequently activates the local rate control as described in 11C.11.

10.3.83 MBSS proxy update

10.3.83.1 Introduction

The following primitives describe how a mesh STA reports the proxy update information to another mesh STA in the MBSS.

10.3.83.2 MLME-MBSSPROXYUPDATE.request

10.3.83.2.1 Function

This primitive requests that the MAC entity inform a destination mesh STA about its proxy information by transmitting a Proxy Update frame to the specified peer MAC entity.

10.3.83.2.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME- MBSSPROXYUPDATE.request(
PeerMACAddress,
ProxyUpdate,
VendorSpecificInfo
)

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Any valid individual MAC address	Specifies the address of the peer MAC entity to which the Proxy Update frame is sent.
ProxyUpdate	A set of Proxy Update elements	As defined in 7.3.2.116	A set of proxy information available at the mesh STA.
VendorSpecificInfo	A set of elements	As defined in 7.3.2.26	Zero or more elements.

10.3.83.2.3 When generated

This primitive is generated by the SME to request that a Proxy Update frame is sent to the specified peer MAC entity.

10.3.83.2.4 Effect of receipt

On receipt of this primitive, the MLME constructs a Proxy Update frame containing the Proxy Update element. This frame is then scheduled for transmission. The MLME subsequently issues an MLME-MBSSPROXYUPDATE.confirm that reflects the results of this request.

10.3.83.3 MLME-MBSSPROXYUPDATE.confirm

10.3.83.3.1 Function

This primitive reports the results of a proxy update request.

10.3.83.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME- MBSSPROXYUPDATE.confirm(
                              PeerMACAddress,
                              ResultCode,
                              ProxyUpdateConfirmation,
                              VendorSpecificInfo
                              )
```

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Any valid individual MAC address	Specifies the address of the peer MAC entity from which the Proxy Update Confirmation frame is received.

Name	Type	Valid range	Description
ResultCode	Enumeration	SUCCESS, INVALID_PARAMETERS, TIME_OUT, UNSPECIFIED_FAILURE	Indicates the result of the MLME-MBSSPROXYUPDATE.request primitive.
ProxyUpdateConfirmation	A set of Proxy Update Confirmation elements	As defined in 7.3.2.117	A set of proxy update confirmation information from the peer MAC entity to which the Proxy Update frame was sent.
VendorSpecificInfo	A set of elements	As defined in 7.3.2.26	Zero or more elements.

10.3.83.3.3 When generated

This primitive is generated by the MLME as a result of an MLME-MBSSPROXYUPDATE.request primitive to indicate the result of that request.

10.3.83.3.4 Effect of receipt

The SME is notified of the results of the MBSS proxy update request.

10.3.83.4 MLME-MBSSPROXYUPDATE.indication

10.3.83.4.1 Function

This primitive indicates that an update of the proxy information has been received from a specific peer MAC entity.

10.3.83.4.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME- MBSSPROXYUPDATE.indication(
                                PeerMACAddress,
                                ProxyUpdate,
                                VendorSpecificInfo
                                )
```

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Any valid individual MAC address	Specifies the address of the peer MAC entity from which the update of the proxy information was received.
ProxyUpdate	A set of Proxy Update elements	As defined in 7.3.2.116	A set of proxy information received from the peer mesh STA.

Name	Type	Valid range	Description
VendorSpecificInfo	A set of elements	As defined in 7.3.2.26	Zero or more elements.

10.3.83.4.3 When generated

This primitive is generated by the MLME as a result of the receipt of a Proxy Update frame from a specific peer MAC entity.

10.3.83.4.4 Effect of receipt

The SME is notified of the results of the receipt of the proxy update request by the specified peer MAC entity. The mesh STA that received this primitive subsequently updates the proxy information as described in11C.10.4.3.

10.3.83.5 MLME-MBSSPROXYUPDATE.response

10.3.83.5.1 Function

This primitive is used to send a response to a specific peer MAC entity that sent an update of the proxy information to the mesh STA that issued this primitive.

10.3.83.5.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME- MBSSPROXYUPDATE.response(
                              PeerMACAddress,
                              ProxyUpdateConfirmation,
                              VendorSpecificInfo
                              )
```

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Any valid individual MAC address	Specifies the address of the peer MAC entity to which the Proxy Update Confirmation frame is sent.
ProxyUpdateConfirmation	A set of Proxy Update Confirmation elements	As defined in 7.3.2.117	A set of proxy update confirmation information to be sent to the peer MAC entity.
VendorSpecificInfo	A set of elements	As defined in 7.3.2.26	Zero or more elements.

10.3.83.5.3 When generated

This primitive is generated by the SME of a STA as a response to an MLME-MBSSPROXYUPDATE.indication primitive.

10.3.83.5.4 Effect of receipt

This primitive initiates transmission of a response to the specific peer MAC entity that sent an update of the proxy information.

On receipt of this primitive, the MLME constructs a Proxy Update Confirmation frame. The frame contains one or more Proxy Update Confirmation elements. This frame is then scheduled for transmission.

10.3.84 MBSS mesh gate announcement

10.3.84.1 Introduction

The following primitives describe how a mesh STA announces mesh gate reachability:

10.3.84.2 MLME-MBSSGATEANNOUNCEMENT.request

10.3.84.2.1 Function

This primitive requests that the MAC entity update the mesh gate information by transmitting a Gate Announcement frame to the specified MAC entity.

10.3.84.2.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME- MBSSGATEANNOUNCEMENT.request(
PeerMACAddress,
GateAnnouncement,
VendorSpecificInfo
)

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Any valid group MAC address	Specifies the address of the MAC entity to which the Gate announcement frame is sent.
GateAnnouncement	GANN element	As defined in 7.3.2.111	A set of gate announcement information to be sent through a Gate Announcement frame.
VendorSpecificInfo	A set of elements	As defined in 7.3.2.26	Zero or more elements.

10.3.84.2.3 When generated

This primitive is generated by the SME to request that a Gate Announcement frame is sent to the specified MAC entity.

10.3.84.2.4 Effect of receipt

On receipt of this primitive, the MLME constructs a Gate Announcement frame containing the GANN element. This frame is then scheduled for transmission following the interval specified by dot11MeshGateAnnouncementInterval. The MLME subsequently issues an MLME-MBSSGATEANNOUNCEMENT.confirm that reflects the results of this request.

10.3.84.3 MLME-MBSSGATEANNOUNCEMENT.confirm

10.3.84.3.1 Function

This primitive reports the results of a mesh gate announcement request.

10.3.84.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME- MBSSGATEANNOUNCEMENT.confirm(
                                   Result Code
                                   )
```

Name	Type	Valid range	Description
Result Code	Enumeration	SUCCESS, INVALID_PARAMETERS, NOT_SUPPORTED	Indicates the result of the MLME-MBSSGATEANNOUNCEMENT.request.

10.3.84.3.3 When generated

This primitive is generated by the MLME as a result of an MLME-MBSSGATEANNOUNCEMENT.request primitive to indicate the result of that request.

10.3.84.3.4 Effect of receipt

The SME is notified of the results of the mesh gate announcement request.

10.3.84.4 MLME-MBSSGATEANNOUNCEMENT.indication

10.3.84.4.1 Function

This primitive indicates that a mesh gate announcement has been received from the specific peer MAC entity.

10.3.84.4.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME- MBSSGATEANNOUNCEMENT.indication(
PeerMACAddress,
GateAnnouncement,
VendorSpecificInfo
)

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Any valid individual MAC address	Specifies the address of the peer MAC entity from which the gate announcement was received.
GateAnnouncement	GANN element	As defined in 7.3.2.111	A set of gate announcement information contained in the received Gate Announcement frame.
VendorSpecificInfo	A set of elements	As defined in 7.3.2.26	Zero or more elements.

10.3.84.4.3 When generated

This primitive is generated by the MLME as a result of the receipt of a Gate Announcement frame from a specific peer MAC entity.

10.3.84.4.4 Effect of receipt

The SME is notified of the reachability to a mesh gate in the mesh BSS. The mesh STA received this primitive subsequently triggers MBSSGATEANNOUNCEMENT.request as described in 11C.10.2.

10.3.85 Mesh link metric

10.3.85.1 Introduction

This subclause describes the management procedures associated with mesh link metric reporting.

10.3.85.2 MLME-MESHLINKMETRICREAD.request

10.3.85.2.1 Function

This primitive requests to read a link metric value between the local MAC entity and a specific peer MAC entity.

10.3.85.2.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-MESHLINKMETRICREAD.request(
PeerMACAddress

)

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Any valid individual MAC address	Specifies the address of the peer MAC entity for which the link metric value is read

10.3.85.2.3 When generated

This primitive is generated by the SME to read the link metric value for the mesh link to the specified peer MAC entity.

10.3.85.2.4 Effect of receipt

On receipt of this primitive, the MLME reports the link metric value. The MLME subsequently issues an MLME-MESHLINKMETRICREAD.confirm that reflects the results of this request.

10.3.85.3 MLME-MESHLINKMETRICREAD.confirm

10.3.85.3.1 Function

This primitive reports the results of a link metric read request.

10.3.85.3.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME- MESHLINKMETRICREAD.confirm(
ResultCode,
LinkMetricValue,
VendorSpecificInfo
)

Name	Type	Valid range	Description
ResultCode	Enumeration	SUCCESS, INVALID_PARAMETERS, or UNSPECIFIED_ERROR	Indicates the result of the link metric measurement request. SUCCESS indicates that both forward and reverse link metrics are available. Either INVALID_PARAMETERS or UNSPECIFIED_ERROR indicates that the request was not processed properly.
LinkMetricValue	Mesh Link Metric Report element	As defined in 7.3.2.100	The link metric value for the mesh link to the specified peer MAC entity.

Name	Type	Valid range	Description
VendorSpecificInfo	A set of elements	As defined in 7.3.2.26	Zero or more elements.

10.3.85.3.3 When generated

This primitive is generated by the MLME as a result of an MLME-MESHLINKMETRICREAD.request primitive to request a link metric value.

10.3.85.3.4 Effect of receipt

The SME is notified of the results of the link metric read request.

10.3.85.4 MLME-MESHLINKMETRICREPORT.request

10.3.85.4.1 Function

This primitive requests that the MAC entity either transmit a link metric to or request a link metric from the specified peer MAC entity.

10.3.85.4.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MESHLINKMETRICREPORT.request(
                        PeerMACAddress,
                        LinkMetricRequestFlag,
                        MeshLinkMetricReport,
                        VendorSpecificInfo
                        )
```

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Any valid individual MAC address	Specifies the address of the peer MAC entity to which the Mesh Link Metric Report is sent.
LinkMetricRequestFlag	Enumeration	REPORT_ONLY, or REPORT_AND_REQUEST	Indicates whether the mesh STA requests a link metric report from the peer MAC entity.
MeshLinkMetricReport	Mesh Link Metric Report element	As defined in 7.3.2.100	A metric value computed for the corresponding link.
VendorSpecificInfo	A set of elements	As defined in 7.3.2.26	Zero or more elements.

10.3.85.4.3 When generated

This primitive is generated by the SME to request that a Mesh Link Metric Report frame is sent to a peer MAC entity in order to report a link metric value and to request a mesh link metric report from the peer MAC entity if LinkMetricRequestFlag is equal to REPORT_AND_REQUEST.

10.3.85.4.4 Effect of receipt

On receipt of this primitive, the MLME constructs a Mesh Link Metric Report frame.The Request subfield in the Flags field of the Mesh Link Metric Report element is set depending on the parameter given by the LinkMetricRequestFlag. If LinkMetricRequestFlag is equal to REPORT_ONLY, the Request subfield is set to 0. If LinkMetricRequestFlag is equal to REPORT_AND_REQUEST, the Request subfield is set to 1. This frame is then scheduled for transmission. The MLME subsequently issues an MLME-MESHLINKMETRICREPORT.confirm that reflects the results of this request.

10.3.85.5 MLME-MESHLINKMETRICREPORT.confirm

10.3.85.5.1 Function

This primitive reports the results of a link metric report request.

10.3.85.5.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME- MESHLINKMETRICREPORT.confirm(
ResultCode,
PeerMACAddress,
VendorSpecificInfo
)

Name	Type	Valid range	Description
ResultCode	Enumeration	SUCCESS, INVALID_PARAMETERS, TRANSMISSION_FAILURE, or UNSPECIFIED_FAILURE	Reports the outcome of a request to submit a link metric report.
PeerMACAddress	MAC Address	Any valid individual MAC address	Specifies the address of the peer MAC entity to which the Mesh Link Metric Report is sent.
VendorSpecificInfo	A set of elements	As defined in 7.3.2.26	Zero or more elements.

10.3.85.5.3 When generated

This primitive is generated by the MLME as a result of an MLME-MESHLINKMETRICREPORT.request primitive to indicate the result of that request.

10.3.85.5.4 Effect of receipt

The SME is notified of the results of the MLME-MESHLINKMETRICREPORT.request.

10.3.85.6 MLME-MESHLINKMETRICREPORT.indication

10.3.85.6.1 Function

This primitive indicates that a Mesh Link Metric Report frame has been received from a peer MAC entity. This Mesh Link Metric Request Report can be in response to an earlier MLME-MESHLINKMETRICREPORT.request primitive with LinkMetricRequestFlag equal to REPORT_AND_REQUEST.

10.3.85.6.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME- MESHLINKMETRICREPORT.indication(
PeerMACAddress,
LinkMetricRequestFlag,
MeshLinkMetricReport,
VendorSpecificInfo
)

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Any valid individual MAC address	Specifies the address of the peer MAC entity from which the Mesh Link Metric Report frame was received.
LinkMetricRequestFlag	Enumeration	REPORT_ONLY, or REPORT_AND_REQUEST	Indicates whether the peer MAC entity requests a link metric report.
MeshLinkMetricReport	Mesh Link Metric Report element	As defined in 7.3.2.100	A metric value reported from the specified peer MAC entity.
VendorSpecificInfo	A set of elements	As defined in 7.3.2.26	Zero or more elements.

10.3.85.6.3 When generated

This primitive is generated by the MLME as a result of the receipt of a Mesh Link Metric Report frame from a specific peer MAC entity.

10.3.85.6.4 Effect of receipt

The SME is notified of the receipt of the link metric report from the specified peer MAC entity. When LinkMetricRequestFlag is equal to REPORT_AND_REQUEST, the mesh STA responds with a Mesh Link Metric Report frame.

10.3.86 HWMP mesh path selection

10.3.86.1 Introduction

The following primitives describe how a mesh STA establishes and maintains a mesh path to a specified peer MAC entity.

10.3.86.2 MLME-HWMPMESHPATHSELECTION.request

10.3.86.2.1 Function

This primitive requests that the MAC entity establish or maintain a mesh path to the specified peer MAC entity by transmitting an HWMP Mesh Path Selection frame to the specified peer MAC entity.

10.3.86.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-HWMPMESHPATHSELECTION.request(
                                    PeerMACAddress,
                                    RootAnnouncement,
                                    PathRequest,
                                    PathReply,
                                    PathError,
                                    VendorSpecificInfo
                                    )
```

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Any valid individual or group MAC address	Specifies the address of the peer MAC entity to which the HWMP Mesh Path Selection frame is sent.
RootAnnouncement	RANN element	As defined in 7.3.2.112	A set of Root Announcement elements generated by the mesh STA. Present only if the mesh STA is configured as a root mesh STA using the proactive RANN mechanism [dot11MeshHWMProotMode = rann (4)], and as described in 11C.9.12.
PathRequest	PREQ element	As defined in 7.3.2.113	A set of Path Request elements generated by the mesh STA. Present as described in 11C.9.9.
PathReply	PREP element	As defined in 7.3.2.114	A set of Path Reply elements generated by the mesh STA. Present as described in 11C.9.10.
PathError	PERR element	As defined in 7.3.2.115	A set of Path Error elements generated by the mesh STA. Present as described in 11C.9.11.

Name	Type	Valid range	Description
VendorSpecificInfo	A set of elements	As defined in 7.3.2.26	Zero or more elements.

10.3.86.2.3 When generated

This primitive is generated by the SME to request that an HWMP Mesh Path Selection frame be sent to a specified peer entity.

10.3.86.2.4 Effect of receipt

On receipt of this primitive, the MLME constructs an HWMP Mesh Path Selection frame. This frame is then scheduled for transmission. The MLME subsequently issues an MLME HWMPMESHPATHSELECTION.confirm that reflects the results of this request.

10.3.86.3 MLME-HWMPMESHPATHSELECTION.confirm

10.3.86.3.1 Function

This primitive reports the results of an HWMP Mesh Path Selection request.

10.3.86.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-HWMPMESHPATHSELECTION.confirm(
                                    ResultCode
                                    )
```

Name	Type	Valid range	Description
Result Code	Enumeration	SUCCESS, INVALID_PARAMETERS, NOT_SUPPORTED	Indicates the result of the MLME-HWMPPathSelection.request.

10.3.86.3.3 When generated

This primitive is generated by the MLME as a result of an MLME-HWMPMESHPATHSELECTION.request primitive to indicate the result of that request.

10.3.86.3.4 Effect of receipt

The SME is notified of the results of the HWMP Mesh Path Selection request.

10.3.86.4 MLME-HWMPMESHPATHSELECTION.indication

10.3.86.4.1 Function

This primitive indicates that an HWMP Mesh Path Selection frame has been received from the specified peer MAC entity.

10.3.86.4.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-HWMPMESHPATHSELECTION.indication(
                                        PeerMACAddress,
                                        RootAnnouncement,
                                        PathRequest,
                                        PathReply,
                                        PathError,
                                        VendorSpecificInfo
                                        )
```

Name	Type	Valid range	Description
PeerMACAddress	MAC Address	Any valid individual MAC address	Specifies the address of the peer MAC entity from which the HWMP Mesh Path Selection frame was received.
RootAnnouncement	RANN element	As defined in 7.3.2.112	A set of Root Announcement elements contained in the received frame. Present only when such an element was present in the received frame.
PathRequest	PREQ element	As defined in 7.3.2.113	A set of Path Request elements contained in the received frame. Present only when such an element was present in the received frame.
PathReply	PREP element	As defined in 7.3.2.114	A set of Path Reply elements contained in the received frame. Present only when such an element was present in the received frame.
PathError	PERR element	As defined in 7.3.2.115	A set of Path Error elements contained in the received frame. Present only when such an element was present in the received frame.
VendorSpecificInfo	A set of elements	As defined in 7.3.2.26	Zero or more elements.

10.3.86.4.3 When generated

This primitive is generated by the MLME as a result of the receipt of an HWMP Mesh Path Selection frame from a specific peer MAC entity.

10.3.86.4.4 Effect of receipt

The SME is notified of the results of the receipt of the HWMP Mesh Path Selection from the specified peer MAC entity. The mesh STA received this primitive subsequently activates path selection procedures described in 11C.9.

11. MLME

11.1 Synchronization

Change the first paragraph in 11.1 as follows:

~~All~~ STAs in ~~within~~ a single infrastructure BSS or IBSS are synchronized to a common clock using the mechanisms defined herein.

In mesh BSSs, STAs use a synchronization method that is part of the extensible synchronization framework. The synchronization in an MBSS is described in 11C.12.

A STA for which dot11OCBEnabled is true is not a member of a BSS, and therefore is not required to synchronize to a common clock or use these mechanisms.

11.1.1 Basic approach

Insert the following new subclause after 11.1.1.2:

11.1.1.3 TSF for an MBSS

The TSF in an MBSS is provided by the active synchronization method. A mesh STA shall initialize its TSF timer according to the active synchronization method. The mesh STA shall periodically transmit Beacon frames that contain a copy of its TSF timer to announce its local time reference. Mesh STAs receiving a Beacon frame use the timing information in the Beacon frame as specified by the active synchronization method. See 11C.12.2 for details.

11.1.2 Maintaining synchronization

Insert the following new subclause after 11.1.2.2:

11.1.2.2a Beacon generation in an MBSS

Beacon generation in an MBSS is described in 11C.12.3.1.

11.1.2.3 Beacon reception

Insert the following paragraph after the third paragraph in 11.1.2.3:

STAs in an MBSS shall use information in received Beacon frames as described in 11C.12.3.2.

11.1.3 Acquiring synchronization, scanning

Change the fifth paragraph in 11.1.3 as follows:

Upon receipt of an MLME-JOIN.request, the non-mesh STA shall use the synchronization procedure described in 11.1.3.4. The MLME-JOIN.request primitive is not used to start synchronization in an MBSS. The synchronization in an MBSS is described in 11C.12.

Insert the following paragraph after the seventh paragraph in 11.1.3:

When scanning MBSSs, the STA shall process the procedures described in 11C.2.

11.1.3.2 Active scanning

11.1.3.2.1 Sending a probe response

Change the first paragraph in 11.1.3.2.1 as follows:

STAs, ~~when dot11InterworkingServiceActivated is true~~ subject to the criteria below, receiving Probe Request frames ~~containing an Interworking field in the Extended Capabilities information element set to 1~~ shall respond with a probe response only if:

~~a) The SSID in the probe request is the wildcard SSID, the SSID in the probe request is the specific SSID of the STA, or the specific SSID of the STA is included in the SSID List element,~~

~~b) The BSSID field in the probe request is the wildcard BSSID or the BSSID of the STA, and~~

~~c) The DA field in the probe request is the broadcast address or the specific MAC address of the STA.~~

~~d) the HESSID field, if present in the Interworking element, is the wildcard HESSID or the HESSID of the STA, and~~

~~e) the Access Network Type field in the Interworking element is the wildcard Access Network Type or the Access Network Type of the STA.~~

a) The Address 1 field in the probe request is the broadcast address or the specific MAC address of the STA, and either item b) or item c) below.

b) The STA is a mesh STA and the Mesh ID in the probe request is the wildcard Mesh ID or the specific Mesh ID of the STA.

c) The STA is not a mesh STA and

    1) The SSID in the probe request is the wildcard SSID, the SSID in the probe request is the specific SSID of the STA, or the specific SSID of the STA is included in the SSID List element, and

    2) The Address 3 field in the probe request is the wildcard BSSID or the BSSID of the STA.

Additionally, STAs with dot11InterworkingServiceActivated equal to true, receiving Probe Request frames containing an Interworking field in the Extended Capabilities information element set to 1 shall examine the Interworking element in the received Probe Request frame and respond with a probe response only if

d) The HESSID field, if present in the Interworking element, is the wildcard HESSID or the HESSID of the STA, and

e) The Access Network Type field in the Interworking element is the wildcard Access Network Type or the Access Network Type of the STA.

Insert the following sentence to the end of the second paragraph in 11.1.3.2.1:

In an MBSS, mesh STAs that receives a probe request shall respond to the probe requests meeting the above criteria.

Change the third paragraph in 11.1.3.2.1 as follows:

Only APs and STAs in an IBSS or in an MBSS respond to probe requests. The procedures defined in this subclause ensure that in each infrastructure BSS and IBSS there is at least one STA that is awake at any given time to receive and respond to probe requests. In an MBSS, STAs may not be awake at any given time to respond to probe requests. In an infrastructure BSS or in an IBSS, a~~A~~ STA that sent a Beacon frame shall remain in the Awake state and shall respond to probe requests, subject to criteria in the next paragraph, until a Beacon frame with the current BSSID is received. If the STA is an AP, it shall remain in the Awake state and always respond to probe requests, subject to criteria in the next paragraph. There may be more than one STA in an IBSS that responds to any given probe request, particularly in cases where more than one STA transmitted a Beacon frame following the most recent TBTT, either due to not receiving successfully a previous Beacon frame or due to collisions between beacon transmissions.

Change the forth paragraph in 11.1.3.2.1 as follows:

In an infrastructure BSS or in an IBSS, STAs receiving Probe Request frames shall respond with a probe response when the SSID in the probe request is the wildcard SSID, matches the specific SSID of the STA, or the specific SSID of the STA is included in the SSID List element. Furthermore, a STA with dot11RadioMeasurementEnabled true receiving a probe request with a DS Parameter Set element containing a Current Channel field value that is not the same as the value of dot11CurrentChannelNumber shall not respond with a probe response. ~~Probe Response frames shall be sent as directed frames to the address of the STA that generated the probe request.~~ An AP shall respond to all probe requests meeting the above criteria. In an IBSS, a STA that transmitted a Beacon frame since the last TBTT shall respond to probe requests.

In an MBSS, mesh STAs receiving Probe Request frames shall respond with a probe response when the SSID in the probe request is the wildcard SSID and the Mesh ID in the probe request is the wildcard Mesh ID or matches the specific Mesh ID of the mesh STA, except when dot11RadioMeasurementEnabled is true and the probe request contains a DS Parameter Set element with a Current Channel field value that is not the same as the value of dot11CurrentChannelNumber.

Probe Response frames shall be sent as directed frames to the address of the STA that generated the probe request. The SSID List element shall not be included in a Probe Request frame in an IBSS.

11.2 Power management

Insert the following new subclause after 11.2.2:

11.2.2a Power management in an MBSS

Power management in an MBSS is described in 11C.13.

11.3 STA authentication and association

Insert the following paragraph after the first paragraph in 11.3:

A STA for which dot11MeshActivated is true (i.e., a mesh STA) does not use procedures described in 11.3.2. Instead, a mesh STA uses a mesh peering management protocol (MPM) or a authenticated mesh peering exchange (AMPE) to manage states and state variables for each peer STA. See 11C.3 and 11C.5 for details.

Change the first sentence in the second paragraph in 11.3 as follows:

~~These two variables create four local states for the relationship between the STA and the remote STA:~~

For non-mesh STAs, this state variable expresses the relationship between the local STA and the remote STA. It takes on the following values:

Insert the following paragraph after the dashed list after the second paragraph in 11.3:

Mesh STAs manage the state variable as described in 11C.3.2.

Change the third paragraph in 11.3 as follows:

~~The relationships between these STA state variables and the services are given in Figure 11-6.~~ Figure 11-6 shows the state transition diagram for these non-mesh STA states. Note that only events causing state changes are shown. The state of the sending STA given by Figure 11-6 is with respect to the intended receiving STA.

Change the forth paragraph in 11.3 as follows:

The current state existing between the ~~source~~ transmitter and ~~destination~~ receiver STAs determines the IEEE 802.11 frame types that may be exchanged between that pair of STAs (see Clause 7). ~~The state of the sending STA given by Figure 11-6 is with respect to the intended receiving STA.~~ The allowed frame types are grouped into classes and the classes correspond to the STA state. In State 1, only Class 1 frames are allowed. In State 2, either Class 1 or Class 2 frames are allowed. In State 3 and State 4, all frames are allowed (Classes 1, 2, and 3). The frame classes are defined as follows:

Insert the following lettered list item "vii)" under the lettered list "a) Class 1 frames" and "2) Management frames," after "vi) Public Action" in 11.3:

vii) Self-protected Action

Delete the currnt list item "c)" and all its hanging items in its entirety.

Insert a new lettered list item "c)" as follows:

c) Class 3 frames

1) Data frames

i) Data frames between STAs in an infrastructure BSS or in an MBSS

2) Management frames

i) Within an infrastructure BSS or an MBSS, all Action and Action No Ack frames except those that are declared to be Class 1 or Class 2 frames (above)

2) Control frames

i) PS-Poll

ii) Within an infrastructure BSS or an MBSS, Block Ack (BlockAck)

iii) Within an infrastructure BSS or an MBSS, Block Ack Request (BlockAckReq)

11.3.1 Authentication and deauthentication

11.3.1.1 Authentication—originating STA

Insert the following dashed list item to the end of the dashed list items under the lettered list "a)" in 11.3.1.1:

— For SAE authentication in an ESS, IBSS, or MBSS, the authentication mechanism described in 8.2a.

11.3.1.2 Authentication—destination STA

Insert the following dashed list item to the end of the dashed list items under the lettered list "a)" in 11.3.1.2:

— For SAE authentication in an ESS, IBSS, or MBSS, the authentication mechanism described in 8.2a.

Change the second paragraph in 11.3.1.2 as follows:

If the requested authentication mechanism is other than FT authentication and the destination STA is not in an MBSS, the STA's SME shall delete any PTKSA and temporal keys held for communication with the indicated STA by using the MLMEDELETEKEYS.request primitive (see 8.4.10) upon receiving a MLME-AUTHENTICATE.indication primitive if and only if Management Frame Protection had not been negotiated when the PTKSA(s) were created.

11.3.1.3 Deauthentication—originating STA

Insert the following lettered list item "d)" to the end of the lettered list items in 11.3.1.3:

d) If the STA is a mesh STA, its SME shall inform the mesh peering instance controller (see 11C.3.4) of the deauthentication.

11.3.1.4 Deauthentication—destination STA

Insert the following lettered list item "c)" to the end of the lettered list items in 11.3.1.4:

c) If the STA is a mesh STA, its SME shall inform the mesh peering instance controller (see 11C.3.4) of the deauthentication.

Insert the following new subclause to the end of 11.3:

11.3.3 Additional mechanisms for an AP collocated with a mesh STA

If the state of an associating STA (a non-AP STA in an infrastructure BSS) located at an access point that is collocated with a mesh STA has successfully reached State 2 (Figure 11-6), the mesh STA collocated with this access point shall verify that the MAC address of the STA does not belong to a mesh STA in the MBSS. If the mesh STA collocated with the access point determines that the authenticated STA has a MAC address that is a MAC address of a mesh STA in the MBSS, then the collocated access point shall deauthenticate the STA with Reason Code "unspecified reason" or "MAC-ADDRESS-ALREADY-EXISTS-IN-MBSS."

The mechanism for verifying the MAC address of the authenticated STA depends on the active path selection protocol and might be vendor specific. See 11C.9.13 when HWMP is the active path selection protocol.

11.7 DLS operation

Change the third paragraph in 11.7 as follows:

DLS does not apply in an IBSS, where frames are always sent directly from one STA to another. DLS does not apply in an MBSS, because frames in an MBSS are always sent directly from one mesh STA to another.

11.8 TPC procedures

Insert the following dashed list item after the first dashed list item after the third paragraph in 11.8:

— Peering of mesh STAs based on the mesh STAs' power capability (see 11.8.1a)

Change the first dashed list item after the fourth paragraph in 11.8:

— A STA with dot11SpectrumManagementRequired set to TRUE shall not operate in an infrastructure BSS, MBSS, or IBSS unless the Spectrum Management bit is set to 1 in the Capability Information field in Beacon frames and Probe Response frames received from other STAs in the infrastructure BSS, MBSS, or IBSS, *with the following exception.*

Insert the following dashed list item after the third dashed list item after the fourth paragraph in 11.8:

— A mesh STA shall set dot11SpectrumManagementRequired to true before becoming a member of an MBSS in which the Spectrum Management bit is set to 1 in the Capability Information field in Beacon frames and Probe Response frames received from the MBSS.

Insert the following new subclause after 11.8.1:

11.8.1a Peering based on transmit power capability

A mesh STA shall provide a candidate peer mesh STA with its minimum and maximum transmit power capability for the current channel when becoming a member of an MBSS, using a Power Capability element in Mesh Peering Open frames.

A mesh STA may use the minimum and maximum transmit power capability of a neighbor peer mesh STA as an input into the algorithm used to determine the local transmit power constraint. The specification of the algorithm is beyond the scope of this standard.

A mesh STA may reject a Mesh Peering Open request from a candidate mesh STA if it considers the candidate mesh STA’s minimum or maximum transmit power capability is unacceptable. For example, a candidate mesh STA’s power capability might be unacceptable if it violates local regulatory constraints. The criteria for establishing or rejecting a Mesh Peering Open request on the basis of transmit power capability are beyond the scope of this standard.

11.8.2 Specification of regulatory and local maximum transmit power levels

Change the first and second paragraph in 11.8.2 as follows:

A STA shall determine a regulatory maximum transmit power for the current channel. The STA shall use the minimum of the following:

— Any regulatory maximum transmit power received in a Country element from the AP in its BSS, ~~or~~ another STA in its IBSS, or a neighbor peer mesh STA in its MBSS and

— Any regulatory maximum transmit power for the channel in the current regulatory domain known by the STA from other sources.

A STA shall determine a local maximum transmit power for the current channel. The STA shall use the minimum of the following:

— Any local maximum transmit power received in the combination of a Country element and a Power Constraint element from the AP in its BSS, ~~or~~ another STA in its IBSS, or a neighbor peer mesh STA in its MBSS and

— Any local maximum transmit power for the channel regulatory domain known by the STA from other sources.

Change the fourth and fifth paragraph in 11.8.2 as follows:

The regulatory and local maximum transmit powers may change in a STA during the life of an infrastructure BSS and an MBSS. However, network stability should be considered when deciding how often or by how much these maximums are changed. The regulatory and local maximum transmit powers shall not change during the life of an IBSS.

An AP in a BSS, ~~and~~ a STA in an IBSS, and a mesh STA in an MBSS shall advertise the regulatory maximum transmit power for the current channel in Beacon frames and Probe Response frames using a Country element. An AP in a BSS, ~~and~~ a STA in an IBSS, and a mesh STA in an MBSS shall advertise the local maximum transmit power for the current channel in Beacon frames and Probe Response frames using the combination of a Country element and a Power Constraint element.

11.9 DFS procedures

Change the third dashed list item after the fourth paragraph in 11.9 as follows:

— A STA shall set dot11SpectrumManagementRequired to TRUE before associating with an infrastructure BSS, ~~or~~ IBSS, or MBSS in which the Spectrum Management bit is set to 1 in the Capability Information field in Beacon frames and Probe Response frames received from the infrastructure BSS, ~~or~~ IBSS, or MBSS.

11.9.2 Quieting channels for testing

Change the first paragraph in 11.9.2 as follows:

An AP in a BSS or a mesh STA in an MBSS may schedule quiet intervals by transmitting one or more Quiet elements in Beacon frames and Probe Response frames. The AP or mesh STA may stop scheduling quiet intervals or change the value of the Quiet Period field, the Quiet Duration field, and the Quiet Offset field in Quiet elements as required. Only the most recently received Beacon frame or Probe Response frame defines all future quiet intervals; therefore, quiet intervals based on older Beacon frames or Probe Response frames shall be discarded.

11.9.6 Requesting and reporting of measurements

Change the last row of Table 11-8 as follows:

Table 11-8—Allowed measurement requests

Service Set	Source of request	Destination of request	Type of measurement request allowed
IBSS, MBSS	STA	STA	Individual or group

11.9.7 Selecting and advertising a new channel

Insert the following new subclause after 11.9.7.2:

11.9.7.2a MBSS channel switching

11.9.7.2a.1 General

The mesh channel switch may be triggered by the need to avoid interference to a detected radar signal, or to reassign mesh STA channels to ensure the MBSS connectivity.

A mesh STA may make use of the information in Supported Channel elements, Supported Regulatory Classes elements, and the results of measurements undertaken by the mesh STAs in the MBSS to assist the selection of the new channel. The algorithm to choose a new channel is beyond the scope of this standard, but shall satisfy applicable regulatory requirements, including uniform spreading rules and channel testing rules.

A 20/40 MHz MBSS may be changed to a 20 MHz MBSS and a 20 MHz MBSS may be changed to a 20/40 MHz MBSS.

When an MBSS switches from a 20 MHz MBSS to a 20/40 MHz MBSS or switches from a 20/40 MHz MBSS to a 20 MHz MBSS, a mesh STA may need to do path maintenance to find an optimized path.

In the following subclauses, Mesh Channel Switch Announcement refers to Mesh Channel Switch Parameters element together with Channel Switch Announcement element or Extended Channel Switch Announcement element.

11.9.7.2a.2 Initiating MBSS channel switch

A mesh STA shall not initiate a new channel switch attempt if there is an ongoing channel switch attempt by this mesh STA.

A mesh shall inform each of the peer mesh STAs that the mesh STA is moving to a new channel while maintaining mesh peerings by advertising the switch using Channel Switch Announcement elements together with Mesh Channel Switch Parameters element in Beacon frames, Probe Response frames, and Channel Switch Announcement frames until the intended channel switch time. The channel switch should be scheduled so that all mesh STAs in the MBSS, including mesh STAs in power save mode, have the opportunity to receive at least one Channel Switch Announcement element before the switch.

The fields in the Channel Switch Announcement element shall be set as follows. The Channel Switch Count field shall be set to the time period until the mesh STA sending the Channel Switch Announcement element switches to the new channel so that the channel switch attempt is propagated throughout the MBSS before the mesh STA leaves the channel. The Channel Switch Mode field is reserved. The New Channel Number field shall be set to the number of the channel to which the mesh STA is moving.

The fields in the Mesh Channel Switch Parameters element shall be set as follows. The Precedence Value field shall be set to a random value selected from a uniform distribution in the range from 0 to 65535. The mesh STA may force mesh STAs in the MBSS to stop transmissions of frames except frames containing Channel Switch Announcement element until the channel switch takes place by setting the Transmit Restrict subfield of the Flags field to 1.

The Reason subfield in the Flags field shall be set to 1 to indicate that the content of the Reason Code field as defined in Table 8-34 (Reason codes) of 8.4.1.7 (Reason Code field) is valid. The Reason Code field shall be set to MESH-CHANNEL-SWITCH-REGULATORY-REQUIREMENTS when channel switch is initiated to meet regulatory requirement; otherwise, the Reason Code field shall be set to MESH-CHANNEL-SWITCH-UNSPECIFIED.

The Initiator subfield of the Flag field shall be set to 1. The Time To Live field should be set to the maximum number of hops (e.g., dot11MeshHWMPnetDiameter) for which this Channel Switch attempt is intended.

11.9.7.2a.3 Processing channel switch announcement

Upon receipt of a Channel Switch Announcement, a mesh STA shall not accept and shall not process the received Channel Switch Announcement element or Extended Channel Switch Announcement element if any of the following is true:

— The Mesh Channel Switch Parameters element is not present in the received frame containing Channel Switch Announcement element or Extended Channel Switch Announcement element.

— The Time To Live field in the received Mesh Channel Switch Parameters element is 0.

— A mesh Channel switch is already running and mesh STA has not yet moved into the new channel and/or regulatory class and the Current Precedence value is greater than or equal to the received Precedence Value.

A mesh STA that receives a Channel Switch Announcement element may choose not to perform the specified switch, but to take alternative action. For example, it may choose to move to a different MBSS.

When mesh STA accepts a channel switch, it shall adopt information received in Channel Switch Announcement element and Mesh Channel Switch Parameters element. The mesh STA shall schedule the Channel Switch as per this information. If the Time To Live field value in the received Mesh Channel Switch Parameters element is greater than one, the mesh STA shall transmit Channel Switch Announcement frame and shall include Channel Switch Announcement element together with Mesh Channel Switch Parameters element in the Beacon and Probe Response frames until the intended channel switch time. The fields in the Channel Switch Announcement shall be set to the values identical to those in the received Channel Switch Announcement frame. The fields in the Mesh Channel Switch Parameters element shall be set to the values identical to those in the received Mesh Channel Switch Parameters element, except for the Time To Live field, Initiator field and the Transmit Restrict subfield of the Flags field. The Time To Live field shall be set to the received Time To Live field minus 1. The Initiator field shall be set to 0. The Transmit Restrict field shall be set to 1 when the mesh STA requires neighboring mesh STAs not to transmit further frames not containing Channel Switch Announcement element on the current channel until the scheduled channel switch. The Transmit Restrict subfield shall be set to 0 otherwise.

It is possible that a channel switch is not successful in moving all the mesh STAs in MBSS to the new operating channel. Transitioning to a new channel does not tear down mesh peerings and existing mesh peerings may be maintained in the new operating channel.

After moving into a new operating channel, the mesh STA shall perform CCA until a frame sequence is detected by which it can correctly set its NAV, or until a period of time equal to the ProbeDelay has transpired.

11.9.7.2a.4 Channel switch across a regulatory class

When dot11RegulatoryClassesImplemented is true and the mesh STA is capable of operating in multiple regulatory classes, the mesh STA shall include the Supported Regulatory Classes element within its Mesh Peering Open frames. The Supported Regulatory Classes element announces the regulatory classes that the mesh STA supports.

When dot11RegulatoryClassesImplemented is true, mesh STAs may switch from the operating channel to a channel in a different regulatory class.

11.9a Extended channel switching (ECS)

11.9a.1 General

Change the first paragraph in 11.9a.1 as follows:

This subclause describes ECS procedures that can be used to change BSS operation in channel frequency and Channel bandwidth. Enabling STAs (see 11.11), APs, ~~and~~ DFS owners, and mesh STAs are each STAs that may construct and transmit frames containing Extended Channel Switch Announcement elements when dot11ExtendedChannelSwitchEnabled is true.

11.9a.3 Selecting and advertising a new channel and/or regulatory class

Insert the following new subclause right after the 11.9a.3.2:

11.9a.3.3 Selecting and advertising a new channel in an MBSS

A mesh STA may make use of the information in the Supported Channels element, Supported Regulatory Classes element, and the results of measurements undertaken by this mesh STA and other mesh STAs in the MBSS to assist the selection of the new channel and/or regulatory class.

The mesh STA that advertises a channel switch shall follow the rules defined in 11.9.7.2a with the following extensions:

a) If a mesh STA is switching to a different regulatory class, then the mesh STA shall use the Extended Channel Switch Announcement element and frame. Alternatively, both the Extended Channel Switch Announcement and the Channel Switch Announcement elements and frames may be used when Channel Switch Announcement elements and frames are permitted for operation in the band signified by the new regulatory class.

b) If a mesh STA is switching to a new channel within the same regulatory class, then the mesh STA shall send the Channel Switch Announcement element and frame, or both the Extended Channel Switch Announcement and the Channel Switch Announcement elements and frames.

c) If both the Extended Channel Switch Announcement and the Channel Switch Announcement elements are transmitted in Public Action frames, they shall be sent in separate frames.

d) The Extended Channel Switch Announcement element shall be included in the Beacon and Probe Response frames until the intended channel switch time.

11.19 STAs communicating data frames outside the context of a BSS

Insert the following paragraph after the second paragraph in 11.19:

When a mesh STA starts an MBSS or becomes a member of an MBSS, it shall set dot11OCBEnabled to false. The STA shall keep dot11OCBEnabled false as long as it provides the mesh facility.

11.21 Tunneled direct-link setup

11.21.1 General

Insert the following paragraph after the tenth paragraph in 11.21.1:

TDLS shall not be used in an MBSS.

11.23 WLAN interworking with external networks procedures

11.23.2 Interworking capabilities and information

Change 11.23.2 as follows:

STAs indicate their support for interworking service by setting the dot11InterworkingServiceEnabled MIB variable to true. When dot11InterworkingServiceEnabled is true, STAs include the Interworking element in Beacon and Probe Response frames, and non-AP STAs or mesh STAs include the Interworking element in Probe Request frames.

When dot11InterworkingServiceEnabled and dot11ExtendedChannelSwitchEnabled are both set to TRUE in an infrastructure BSS, the AP may provide its operating channel and regulatory class to an Interworked SSPN using the values from dot11RegulatoryClassesTable MIB entry.

In an infrastructure BSS, t~~T~~he Interworking element contains signaling for Homogeneous ESSs. The HESSID is a 6-octet MAC address that identifies the homogeneous ESS. The HESSID value shall be identical to one of the BSSIDs in the homogeneous ESS. Thus, it is a globally unique identifier that in conjunction with the SSID, may be used to provide network identification for an SSPN.

NOTE—It is required by this standard that the HESSID field in the Interworking element is administered consistently across all BSSs in a homogeneous ESS.

The Interworking element also provides an Access Network Type in Beacon and Probe Response frames to assist the non-AP STA or mesh STA with network discovery and selection.

11.23.6 Interworking procedures: emergency services support

Change the first paragraph in 11.23.6 as follows:

Emergency service support provides STAs with the ability to contact authorities in an emergency situation. The following procedures allow the STA to determine whether emergency services are supported by the AP or mesh STA, and whether unauthenticated emergency service access is allowed.

Change the third paragraph in 11.23.6 as follows:

When the AP of an infrastructure BSS is located in a regulatory domain that requires location capabilities, the ESR field shall only be set to 1 and the Network Type shall only be set to “Emergency services only network” (see Table 7-43bb), if location capability is enabled on the AP. In Beacon and Probe Response frames, location capability is advertised when the Civic Location or Geo Location field in the Extended Capabilities Element is set to 1.

In an MBSS, if location capability is supported, the mesh STA shall report its location for emergency services.

When dot11ESNetwork is true in a mesh STA, the ESR shall be set to 1. When that mesh STA receives a Mesh Peering Open frame that includes the Interworking element with the ASRA field equal to 1, it allows access to emergency services and forwards MSDUs to an emergency server.

NOTE —The ASRA bit set to 1, informs the mesh STA to prioritize resources for the emergency call, to proactively find a better path before the link conditions deteriorate below a certain threshold, and/or to change some of the mesh STA’s behavior (for example, to disable any power save features).

When dot11ESNetwork is false in a mesh STA, the ESR shall be set to 0. When that mesh STA receives a Mesh Peering Open frame that includes the Interworking element with the ASRA field equal to 1, it is unable to support the mesh peering of emergency services and does not forward MDSUs to an emergency server.

11A. Fast BSS transition

11A.2 Key holders

11A.2.2 Authenticator key holders

Change the third paragraph in 11A.2.2 as follows:

The R0KH derives the PMK-R0 for use in the mobility domain utilizing ~~either~~ the MSK (when the AKM negotiated is 00-0F-AC:3), ~~or~~ the PSK (when the AKM negotiated is 00-0F-AC:4) or the PMK (when the AKM negotiated is 00-0F-AC:9). The R0KH shall be responsible for deriving a PMK-R1 for each R1KH within the mobility domain.

11A.2.3 Supplicant key holders

Change the third paragraph in 11A.2.3 as follows:

The S0KH derives the PMK-R0 for use in the mobility domain utilizing ~~either~~ the MSK (when the AKM negotiated is 00-0F-AC:3), ~~or~~ the PSK (when the AKM negotiated is 00-0F-AC:4) or the PMK (when the AKM negotiated is 00-0F-AC:9).

11A.4 FT initial mobility domain association

11A.4.2 FT initial mobility domain association in an RSN

Change the first paragraph in 11A.4.2 as follows:

A STA indicates its support for the FT procedures by including the MDIE in the (Re)Association Request frame and indicates its support of security by including the RSNIE. The AP responds by including the FTIE, MDIE, and RSNIE in the (Re)Association Response frame. After a successful IEEE 802.1X authentication (if needed) or SAE authentication, the STA and AP perform an FT 4-Way Handshake. At the end of the sequence, the IEEE 802.1X Controlled Port is opened, and the FT key hierarchy has been established. The message flow is shown in Figure 11A-2.

Change the fourth paragraph in 11A.4.2 as follows:

Upon successful IEEE 802.11 Open System authentication (if the suite type is 00-0F-AC:3 or 00-0F-AC:4) or SAE authentication (if the suite type is 00-0F-AC:9), the STA shall send a (Re)Association Request frame to the AP that includes the MDE. The contents of the MDE shall be the values advertised by the AP in its Beacon or Probe Response frames. Additionally, the STA includes its security capabilities in the RSNE.

Change the sixth paragraph in 11A.4.2 as follows:

If the contents of the MDE received by the AP do not match the contents advertised in the Beacon and Probe Response frames, the AP shall reject the (Re)Association Request frame with status code 54 (i.e., Invalid MDE). If an MDE is present in the (Re)Association Request frame and the contents of the RSNE do not indicate a negotiated AKM of Fast BSS Transition (suite type 00-0F-AC:3, ~~or~~ 00-0F-AC:4, or 00-0F-AC:9), the AP shall reject the (Re)Association Request frame with status code 43 (i.e., Invalid AKMP).

Change the eighth paragraph in 11A.4.2 as follows:

On successful (re)association, the S0KH on the STA and the R0KH on the AP then proceed with an IEEE 802.1X authentication using EAPOL messages carried in IEEE 802.11 data frames if SAE authentication was not performed (i.e., if the suite type is not 00-0F-AC:9). The S0KH shall use the value of R0KH-ID as the endpoint identifier of the NAS Client (NAS-Identifier if RADIUS is used) in the exchange as defined in IETF RFC 3748-2004 [B26].

Change the ninth paragraph in 11A.4.2 as follows:

~~Upon successful completion of the IEEE 802.1X authentication,~~ If IEEE 802.1X authentication was performed, then upon successful completion of authentication the R0KH receives the MSK and authorization attributes. If SAE authentication was performed the R0KH receives the PMK resulting in the successful completion of SAE. If a key hierarchy already exists for this non-AP STA belonging to the same mobility domain (i.e., having the same MDID), the R0KH shall delete the existing PMK-R0 security association and PMK-R1 security associations. It then calculates the PMK-R0, PMKR0Name, and PMK-R1 and makes the PMK-R1 available to the R1KH of the AP with which the STA is associated.

11A.5 FT Protocol

11A.5.3 Over-the-DS FT Protocol authentication in an RSN

Change the second sentence of sixth paragraph in 11A.5.3 as follows:

If the contents of the RSNE do not indicate a negotiated AKM of Fast BSS Transition (suite type 00-0F-AC:3, ~~or~~ 00-0F-AC:4, or 00-0F-AC:9), the AP shall reject the FT Request frame with status code 43 (i.e., Invalid AKMP).

11A.8 FT authentication sequence

11A.8.4 FT authentication sequence: contents of third message

Change the third dashed list item after the third paragraph in 11A.8.4 as follows:

- When the negotiated AKM is 00-0F-AC:3, ~~or~~ 00-0F-AC:4, or 00-0F-AC:9, the MIC shall be calculated using the KCK and the AES-128-CMAC algorithm.The output of the AES-128-CMAC shall be 128 bits.

11A.8.5 FT authentication sequence: contents of fourth message

Change the fourth dashed list item after the fourth paragraph in 11A.8.5 as shown:

- When the negotiated AKM is 00-0F-AC:3, ~~or~~ 00-0F-AC:4, or 00-0F-AC:9, the MIC shall be calculated using the KCK and the AES-128-CMAC algorithm. The output of the AES-128-CMAC algorithm shall be 128 bits.

Insert the following new clause after Clause 11B:

11C. MLME mesh procedures

11C.1 Mesh STA dependencies

When dot11MeshActivated is true, the STA is a mesh STA.

When dot11MeshActivated is true, following MIB attributes shall be set to true.

- dot11QosOptionImplemented
- dot11ExtendedChannelSwitchEnabled
- dot11SpectrumManagementRequired

When dot11MeshActivated is true, following MIB attributes shall be set to false.

- dot11OCBEnabled
- dot11FastBSSTransitionEnabled

A mesh STA does not support functionalities that depend on AP or are only available in an infrastructure BSS, such as HCCA, Traffic specifications (TSPECs), Traffic stream (TS) management, Admission control, Automatic power save delivery (APSD), Direct-link setup (DLS), Tunneled direct-link setup (TDLS) or tunneled direct-link setup (TDLS).

An HT mesh STA does not support PSMP, STBC, or PCO.

11C.2 Mesh discovery

11C.2.1 General

A mesh STA shall perform either active scanning or passive scanning to discover an operating mesh BSS using the SCAN primitive (see 10.3.2). A mesh profile, a set of parameters identifying the mesh BSS configuration, is also obtained through the scanning process, and it is used to determine the scanning mesh STA's active mesh profile. Based on the result of the scan, the mesh STA may establish a new mesh BSS or become a member of the existing mesh BSS, using the START primitive (see 10.3.10). The MLME-START.request primitive triggers beaconing that facilitates the discovery of the mesh STA by the neighbor mesh STAs. A mesh STA that becomes a member of a mesh BSS should establish a mesh peering with one or more neighbor mesh STAs that are in the same mesh BSS.

11C.2.2 Mesh identifier

The Mesh ID represents the identity of an MBSS. The Mesh ID may be installed in mesh capable devices by a variety of means that are beyond the scope of this standard. For example, the Mesh ID might be set by the user, e.g., "Mike's Mesh." A mesh STA shall include the Mesh ID element (see 7.3.2.99) containing its Mesh ID in its Beacon and Probe Response frames, in order to advertise its identity. The mesh STA shall also include the Mesh ID element containing its Mesh ID in its Mesh Peering Open frames, Mesh Peering Confirm frames, and Mesh Peering Close frames.

The mesh STA shall set the SSID element (see 7.3.2.1) in Beacon, Probe Request, and Probe Response frames to the wildcard SSID.

NOTE—The wildcard SSID is used to notify non-mesh STAs that the mesh STA is neither a part of an infrastructure BSS nor an IBSS, so that the non-mesh STAs do not try to join the mesh BSS.

11C.2.3 Mesh profile

A mesh profile is a set of parameters that specifies the attributes of a mesh BSS. A mesh profile consists of the following:

a) A Mesh ID—specified by dot11MeshID

b) A path selection protocol identifier—specified by dot11MeshActivePathSelectionProtocol

c) A path selection metric identifier—specified by dot11MeshActivePathSelectionMetric

d) A congestion control mode identifier—specified by dot11MeshActiveCongestionControlMode

e) A synchronization method identifier—specified by dot11MeshActiveSynchronizationMethod

f) An authentication protocol identifier—specified by dot11MeshActiveAuthenticationProtocol

In a mesh BSS all mesh STAs use the same mesh profile. Mesh profiles are considered the same if all parameters in the mesh profiles match.

Before establishing a mesh BSS or becoming a member of a mesh BSS, a mesh STA shall configure one mesh profile. The mesh STA shall not change its mesh profile unless it leaves the mesh BSS of which it is a member. When the mesh STA leaves the mesh BSS of which it is a member, it should explicitly close all of its active mesh peerings using Mesh Peering Close frames (see 11C.3.8) and shall discard all session information obtained while the mesh profile was active, such as local forwarding information, security associations (and related keys), etc. The MLME receives the mesh STA's mesh profile from the SME upon receipt of the MLME-START.request primitive.

The mesh profile is signalled by means of the Mesh ID element and the Mesh Configuration element. The mesh profile is included in the Beacon and Probe Response frames, so that the mesh profile can be obtained by its neighbor mesh STAs through the scan. Mesh Peering Open and Mesh Peering Confirm frames also contain a mesh profile.

11C.2.4 Mesh STA configuration

The mesh STA configuration consists of the mesh profile (see 11C.2.3), the Supported Rates element, the Extended Supported Rates element, and the HT Operations element (if present).

Mesh STA configurations are identical if the following conditions hold:

- The mesh profiles are identical
- The BSSBasicRateSet parameters are identical
- For HT mesh STAs, the BSSBasicMCSSet parameters are identical

11C.2.5 Supplemental information for the mesh discovery

A mesh STA shall signal that it is able to establish additional mesh peerings by setting the Accepting Additional Mesh Peerings subfield in the Mesh Capability field in the Mesh Configuration element to 1 (see 7.3.2.98.8). The mesh STA sets the Accepting Additional Mesh Peerings subfield in the Mesh Capability field in the Mesh Configuration element to 0 when it is not able to accept new mesh peerings. This parameter is dynamically controlled by the SME and given to the MLME by dot11MeshAcceptingAdditionalPeerings.

NOTE—This control is driven by internal policies. When the Accepting Additional Mesh Peering subfield is 1, the mesh STA is assumed to have sufficient internal resources to accommodate more mesh peerings. The internal policy is outside the scope of this standard. For instance, a mesh STA might be configured to be able to maintain only two mesh peerings.

A mesh STA shall announce its topological information through the Mesh Formation Info field in the Mesh Configuration element. The contents of the Mesh Formation Info field shall be coded to reflect the current configuration.

11C.2.6 Scanning mesh BSSs

A mesh STA shall perform active scanning or passive scanning, depending on the value of the ScanMode parameter of the MLME-SCAN.request primitive (see 11.1.3), to discover neighbor mesh STAs. Upon receipt of an MLME-SCAN.request primitive with the Mesh ID parameter set to the wildcard Mesh ID, the STA shall passively scan for any Beacon frames, or actively transmit Probe Request frames containing the wildcard Mesh ID, as appropriate, depending on the value of ScanMode. Upon completion of scanning, an MLME-SCAN.confirm primitive is issued by the MLME indicating all of the discovery information received. Further, mesh STAs shall conform to the passive scan procedure as described in 11.1.3.1 and the active scan procedure as described in 11.1.3.2.

11C.2.7 Candidate peer mesh STA

When a mesh STA discovers a neighbor mesh STA through the scanning process and the discovered mesh STA is considered a candidate peer mesh STA, it may become a member of the mesh BSS of which the discovered mesh STA is a member and establish a mesh peering with the neighbor mesh STA.

The discovered neighbor mesh STA shall be considered a candidate peer mesh STA if and only if all of the following conditions are met:

a) The mesh STA uses the same mesh profile as the received Beacon or Probe Response frame indicates for the neighbor mesh STA.

   NOTE—If the scanning mesh STA has not become a member of any MBSS yet, it might simply activate the same mesh profile as the discovered neighbor mesh STA's profile to fulfill this condition.

b) The Accepting Additional Mesh Peerings subfield in the Mesh Capability field in the received Beacon or Probe Response frame equals 1.

c) The mesh STA supports the data rates indicated by the BSSBasicRateSet of the received Beacon or Probe Response frame.

d) If both the scanning mesh STA and the discovered neighbor STA are HT STAs, the mesh STA uses the same BSSBasicMCSSet as the received Beacon or Probe Response frame indicates for the neighbor mesh STA.

e) If the scanning mesh STA has dot11MeshSecurityActivated set to true and the dot11MeshActiveAuthenticationProtocol is ieee8021x (2), either the scanning mesh STA has an active connection to an AS or the discovered mesh STA has the Connected to AS subfield in the Mesh Formation field in the Mesh Configuration element equal to 1 in the received Beacon or Probe Response frame.

11C.2.8 Establishing or becoming a member of a mesh BSS

The Mesh Formation Info field in the Mesh Configuration element is available to assist scanning mesh STAs in choosing the mesh BSS of which to become a member. The details of the usage of this information are beyond the scope of this standard.

NOTE 1—Selection of the mesh BSS of which the scanning mesh STA becomes a new member is outside the scope of this standard. That is, the mesh STA might freely select the mesh BSS of a candidate peer mesh STA of which it becomes a new member.

After the determination of the active mesh profile, the mesh STA may establish a new mesh BSS or become a new member to an existing mesh BSS.

When dot11MBCAActivated is true, the mesh STA shall perform the TBTT selection procedure described in 11C.12.4.3 using TimeStamp, Local Time, Beacon Period, and Beacon Timing in the BSSDescription parameter given by the MLME-SCAN.confirm primitive, before starting its beaconing.

When dot11MCCAActivated is true, the mesh STA shall choose a DTIM interval with a duration of $2^n \times$ 100 TU with n a non-negative integer less than or equal to 17.

NOTE 2—It is allowed that a different value for the DTIM interval is used for mesh STAs that use MCCA in an MBSS that is centrally controlled and the central authority provides a coordination of the DTIM interval of mesh STAs that use MCCA in the MBSS.

When dot11MultiDomainCapabilityEnabled is true, the mesh STA shall not establish or become a member of a mesh BSS, unless a properly formed Beacon frame including a Country element is constructed, and dot11CountryString has been set.

A mesh STA shall include a Country element in its Beacon frames if either dot11MultiDomainCapabilityEnabled, dot11SpectrumManagementRequired, or dot11RadioMeasurementEnabled is true. See 7.2.3.1 for the description of a properly formed Beacon frame.

The mesh STA establishes a new mesh BSS by activating a mesh profile that is different from any mesh profile discovered during the scanning of mesh BSSs (see 11C.2.6).

The mesh STA becomes a new member of an existing mesh BSS by activating the same mesh profile as received from a candidate peer mesh STA of this mesh BSS (see 11C.2.6 and 11C.2.7).

In either case, the mesh STA shall start beaconing using the START primitive. Upon receipt of the MLME-START.request primitive, the mesh STA shall initialize and start its TSF timer as specified by its active synchronization method as described in 11C.12.2, and begin transmitting Beacon frames as described in 11C.12.3.

If the mesh STA has become a new member of an existing mesh BSS, it should establish a mesh peering with one or more candidate peer mesh STAs of this mesh BSS (see 11C.2.9) in order to form the MBSS.

If the mesh STA has become a new member of an existing mesh BSS, it shall adopt the BSSBasicRateSet parameter from a candidate peer mesh STA of this mesh BSS.

After establishing or becoming a member of an MBSS, the mesh STA may continue the discovery procedure described in 11C.2.6 to discover other candidate peer mesh STAs.

11C.2.9 Establishing mesh peerings

Mesh peerings shall be established only with candidate mesh STAs that are members of the same MBSS.

A mesh peering is established between the mesh STA and the candidate peer mesh STA after the successful completion of the mesh peering management (MPM) protocol (see 11C.3) or of the authenticated mesh peering exchange (AMPE) (see 11C.5). When establishing a secure mesh peering, mesh STAs authenticate each other and create a mesh PMKSA before processing the AMPE (see 11C.3.3).

A candidate peer mesh STA becomes a peer mesh STA when a mesh peering is established between the two mesh STAs.

11C.3 Mesh peering management (MPM)

11C.3.1 General

The mesh peering management (MPM) protocol is used to establish, maintain, and close mesh peerings between mesh STAs when dot11MeshSecurityActivated is false. When dot11MeshSecurityActivated is true, the peers establish an authenticated mesh peering using the authenticated mesh peering exchange

(AMPE) protocol. The AMPE protocol requires an existing mesh PMKSA. If a mesh PMKSA with the candidate peer mesh STA exists AMPE shall use that mesh PMKSA. If no mesh PMKSA exists the peers shall first authenticate to establish a mesh PMKSA, see 11C.5.

Figure 11C-1 shows the logical flow of protocol interactions in the peering management framework.

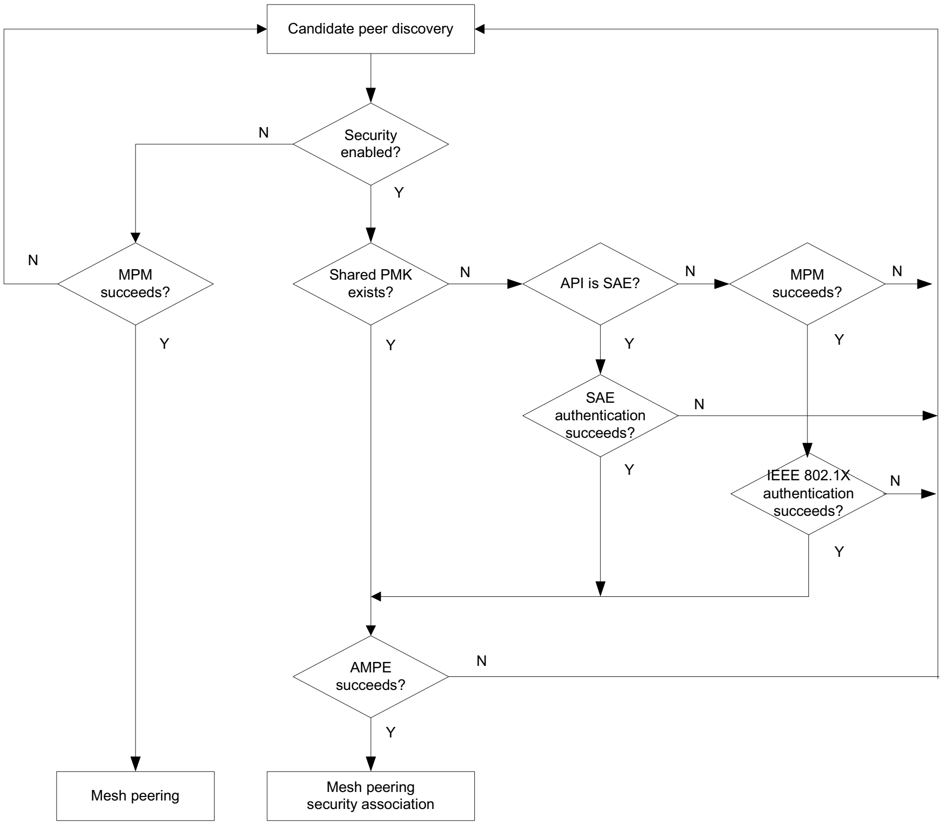


Figure 11C-1—Logical flowchart of protocol interaction in the mesh peering management framework

The MPM protocol uses Mesh Peering Open frames, Mesh Peering Confirm frames, and Mesh Peering Close frames to establish, manage, and tear down a mesh peering.

The protocol succeeds in establishing a mesh peering when the following requirements are satisfied: 1) both mesh STAs have sent and received (and correctly processed) a Mesh Peering Open frame for this mesh peering; 2) both mesh STAs have sent and received (and correctly processed) a corresponding Mesh Peering Confirm frame for this mesh peering.

A mesh STA that receives and accepts a Mesh Peering Open frame (see 11C.3.6.2) shall assign a unique AID among its neighbor peer mesh STAs to the transmitter of the frame. The AID is used in the encoding of the TIM element in the Beacon frame (see 7.3.2.6). AID 0 (zero) is reserved to indicate the presence of buffered groupcast MSDUs and MMPDUs (see 11C.13.4).

11C.3.2 State variable management

A mesh STA keeps an enumerated state variable (see 11.3) for each neighbor STA with which direct communication via the WM is needed. This state variable expresses the relationship between the local STA and a neighbor STA that varies depending on the active authentication protocol. It takes on the values shown in Table 11C-1.

Table 11C-1—State variables for mesh STAs

State	Active authentication		
	None	SAE	IEEE 802.1X
State 1	Initial start state, mesh peering not established	Initial start state, unauthenticated, mesh peering not established	Initial start state, unauthenticated, mesh peering not established
State 2	N/A	Authenticated, mesh peering not established	N/A
State 3	Mesh peering established	Authenticated, mesh peering established	Unauthenticated, mesh peering established (Pending IEEE 802.1X authentication)
State 4	N/A	N/A	Authenticated, mesh peering established

The state transitions in accordance with the protocol interaction shown in Figure 11C-1.

The current state existing between the neighbor STAs determines the IEEE 802.11 frame types that may be exchanged between that pair of STAs (see Clause 7). The allowed frame types are grouped into classes and the classes correspond to the STA state. The allowed frame types and the frame classes in each state are defined in 11.3.

Mesh STAs shall not transmit frames other than the ones used for candidate peer mesh STA discovery, MPM, and SAE to a neighboring mesh STA until a mesh peering has been established with the mesh STA.

11C.3.3 Mesh authentication

In order to create a secure peering, mesh STAs first authenticate each other and create a mesh PMKSA. This can be done using either SAE or IEEE 802.1X. Mesh STAs shall support SAE authentication (see 8.2a) using a pre-shared secret with the candidate peer mesh STA. Optionally, mesh STAs may support IEEE 802.1X authentication [see 5.8 (IEEE Std 802.11 and IEEE Std 802.1X-2004)].

When dot11MeshActiveAuthenticationProtocol is sae (1), the scanning mesh STA shall initiate SAE to the candidate mesh STA. If SAE terminates unsuccessfully, the scanning mesh STA shall terminate the peering establishment procedure. Otherwise, the PMK that results from successful SAE authentication shall be used to create a mesh PMKSA.

When dot11MeshActiveAuthenticationProtocol is ieee8021x (2), then the scanning mesh STA shall initiate the MPM protocol to establish a peering. If the MPM protocol fails then the scanning mesh STA shall terminate the peering establishment procedure. Otherwise, IEEE 802.1X authentication shall be performed between the two peers according to the following:

a) If only one mesh STA has the Connected to AS field set to 1, that STA shall act as the IEEE 802.1X authenticator and the other STA shall act as the IEEE 802.1X supplicant;

b) If both mesh STAs have the Connected to AS field set to 1, then the mesh STA with the higher MAC address shall act as the IEEE 802.1X authenticator and the other mesh STA shall act as the IEEE 802.1X supplicant (see 8.5.1 for MAC address comparison).

If IEEE 802.1X authentication fails, the peering establishment procedure shall be terminated and the peering established between the two mesh STAs shall be closed. Otherwise, the peering established between the two mesh STAs shall be closed and a mesh PMKSA shall be created using the PMK that resulted from the successful IEEE 802.1X authentication.

11C.3.4 Mesh peering instance controller

11C.3.4.1 Overview

A mesh STA uses a mesh peering instance controller to manage all mesh peering instances.

The mesh peering instance controller performs the following functions:

- Create and destroy MPM finite state machines and AMPE finite state machines
- Manage instance identifiers for each mesh peering instance
- Manage mesh TKSAs for each mesh peering instance when dot11MeshSecurityActivated is true
- Pre-process the incoming Mesh Peering Management frames and pass the frames to the finite state machine with matching instance identifier
- Pass internal commands to the finite state machine with matching instance identifier

A mesh peering instance is identified by a mesh peering instance identifier. The mesh peering instance identifier is the set of localLinkID, localMAC, and peerMAC.

A mesh peering instance consists of its identifier (the localLinkID, localMAC, peerMAC), a peerLinkID (an integer generated by the peer mesh STA or candidate peer mesh STA), and the configuration and capability negotiated and agreed upon by exchanging Mesh Peering Open frames (see 7.4.14.2) and Mesh Peering Confirm frames (see 7.4.14.3). If dot11MeshSecurityActivated is true, the mesh peering instance also contains a PMKID identifying the shared PMKSA, a localNonce chosen by the mesh STA and a peerNonce chosen by the peer mesh STA or candidate peer mesh STA.

The localMAC is the MAC address of the mesh STA that is managing this mesh peering instance. The peerMAC is the MAC address of the peer mesh STA or the candidate peer mesh STA. The localLinkID is an integer generated by the mesh STA. The localLinkID shall be unique among all existing link identifiers used by the mesh STA for its MPM finite state machines. The mesh STA selects the localLinkID to provide high assurance that the same number has not been used to identify a recent MPM finite state machine. The peerLinkID is the localLinkID of the peer mesh STA or candidate peer mesh STA and is supplied in the Mesh Peering Management element (see 7.3.2.102) of the Mesh Peering Open and Mesh Peering Confirm frames.

A mesh peering instance is controlled by an MPM finite state machine (see Table 11C-2) or an AMPE finite state machine (see Table 11C-3).

11C.3.4.2 Creating a new mesh peering instance

The mesh peering instance controller creates a new mesh peering instance after either of the following two events:

- The receipt of a Mesh Peering Open frame from a candidate peer mesh STA according to the rules of 11C.3.5
- The receipt of an MLME-MESHPEERINGMANAGEMENT.request primitive with a Mesh Peering Open frame

A unique localLinkID shall be generated for the mesh peering instance. If the mesh peering instance is established by AMPE, a random local nonce shall also be generated.

A mesh STA may create multiple mesh peering instances to establish a peering with the same candidate peer mesh STA.

11C.3.4.3 Deleting mesh peering instances

The mesh peering instance controller deletes a mesh peering instance after either:

- Expiry of a holding timer (see 11C.4.4).
- The acceptance of a peer's response to an existing request to close the peering (see 11C.4.3).
- Indication from the SME that the peer mesh STA, or candidate peer mesh STA, has deauthenticated.

When the deletion occurs, the mesh TKSA that is bound to the mesh peering shall be deleted.

11C.3.5 Mesh peering instance selection

The content of a Mesh Peering Management frame received from a candidate peer mesh STA, and the set of mesh peering instances in the mesh peering instance controller determine whether

- A new mesh peering instance is created (see 11C.3.4.2); or,
- An existing mesh peering instance is updated

If dot11MeshSecurityActivated is true and the mesh STA shares a PMK with the candidate peer mesh STA but the Mesh Peering Protocol Identifier field in the Mesh Peering Management element of the frame indicates "mesh peering management protocol," the frame shall be silently discarded.

If dot11MeshSecurityActivated is true and the mesh STA shares a PMK with the candidate peer mesh STA but either the Mesh Peering element or the MIC element are not present in the frame, the frame shall be silently discarded.

If dot11MeshSecurityActivated is false but the Mesh Peering Protocol Identifier field in the Mesh Peering Management element of the received frame indicates "authenticated mesh peering exchange," the frame shall be silently discarded.

If dot11MeshSecurityActivated is false but either the Mesh Peering element or the MIC element is present in the frame, the frame shall be silently discarded.

If the frame contains a group address in TA or RA, it shall be silently discarded.

If the incoming Mesh Peering Management frame is for AMPE and the Chosen PMK from the received frame contains a PMKID that does not identify a valid mesh PMKSA, the frame shall be silently discarded.

If the Mesh Peering Management frame has not been silently discarded, the mesh peering instance controller attempts to locate a matching mesh peering instance identifier. A match is determined by comparing the contents of the Mesh Peering Management frame with each peering instance. A match is found if all the following conditions are true:

- The transmitter's MAC address (Address 2) is the same as the peerMAC of the mesh peering instance
- The receiver's MAC address (Address 1) is the same as the localMAC of the mesh peering instance
- The value of the Peer Link ID field is the same as the localLinkID of the mesh peering instance

If the incoming frame is a Mesh Peering Open frame and no matching peering instance was found, a new mesh peering instance is created (and a new Mesh TSKA if dot11MeshSecurityActivated is true). See 11C.3.4.2.

If the incoming frame is a Mesh Peering Confirm or Mesh Peering Close frame and no matching mesh peering instance is found, it shall be silently discarded.

If the incoming Mesh Peering Management frame is for AMPE and has not been discarded it shall be further processed as follows:

- If the Peer Nonce field is present in the received frame, and the localNonce in the mesh peering instance is different than the Peer Nonce field of the received frame, the frame shall be dropped.
- If the peerNonce in the mesh peering instance exists and is different than the Local Nonce field of the received frame, the frame shall be dropped.

11C.3.6 Mesh peering open

11C.3.6.1 Generating Mesh Peering Open frames

A Mesh Peering Open frame is generated as a result of a sendOpen() action (see 11C.4.3).

The contents of the frame are described in 7.4.14.2.2.

11C.3.6.2 Mesh Peering Open frame processing

The mesh STA checks that the Mesh ID element and Mesh Configuration element of the Mesh Peering Open frame is identical to its own mesh STA configuration as specified in 11C.2.3 and 11C.2.4. If a mismatch is found the frame shall be rejected with a reason code of MESH-CONFIGURATION-POLICY-VIOLATION and the mesh peering establishment attempt shall be terminated.

When the mesh STA has established a mesh PMKSA with the candidate peer mesh STA, the mesh peering instance controller shall silently discard the Mesh Peering Open frame in the following two conditions:

- The Mesh Peering Open frame supports MPM protocol and the negotiated active authentication is SAE, or
- The Mesh Peering Open frame supports AMPE but the PMKID in the Chosen PMK field in the Authenticated Mesh Peering Exchange element does not identify a mesh PMKSA.

If the Mesh Peering Open frame is not discarded, the mesh peering instance controller actively rejects or accepts the mesh peering open request (see 11C.4). If dot11MeshAcceptingAdditionalPeerings is set to zero the Mesh Peering Open request shall be rejected with reason code MESH-MAX-PEERS.

If the peerLinkID in the mesh peering instance has not been set, the Local Link ID field of the Mesh Peering Open request shall be copied into the peerLinkID in the mesh peering instance. If the incoming Mesh Peering Open frame is for AMPE and the peerNonce in the mesh peering instance has not been set, the Local Nonce field in the incoming Mesh Peering Open frame shall be copied into the peerNonce in the mesh peering instance.

The mesh peering open request may be rejected due to an internal reason with a reason code of MESH-PEERING-CANCELED.

If the Mesh Peering Open request is rejected, the REQ_RJCT event shall be passed with the specified reason code to the protocol finite state machine to actively reject the mesh peering open request.

NOTE—Example internal reasons to reject new mesh peering request could be the mesh STA has reached its capacity to set up more mesh peering, the mesh STA is configured to reject mesh peering request from another specific peer mesh STA.

11C.3.7 Mesh peering confirm

11C.3.7.1 Generating Mesh Peering Confirm frames

A Mesh Peering Confirm frame is generated as a result of a sendConfirm() action (see 11C.4.3).

The contents of the frame are described in 7.4.14.3.2.

11C.3.7.2 Mesh Peering Confirm frame processing

The mesh STA shall check that the Mesh ID element and Mesh Configuration element of the Mesh Peering Confirm frame match its own mesh STA configuration as specified in 11C.2.3 and 11C.2.4. If a mismatch is found, the frame shall be rejected with the reason code of MESH-INCONSISTENT-PARAMETERS.

Otherwise, the mesh STA accepts the Mesh Peering Confirm frame and performs the actions described in 11C.4.

If the peerLinkID in the mesh peering instance has not been set, the Local Link ID field of the Mesh Peering Confirm request shall be copied into the peerLinkID in the mesh peering instance. If the incoming Mesh Peering Confirm frame is for AMPE and the peerNonce in the mesh peering instance has not been set, the Local Nonce field in the incoming Mesh Peering Confirm frame shall be copied into the peerNonce in the mesh peering instance.

11C.3.8 Mesh peering close

11C.3.8.1 Generating Mesh Peering Close frames

A Mesh Peering Close frame is generated as a result of a sendClose() action (see 11C.4.3).

The contents of the frame are described in 7.4.14.4.2.

When the Mesh Peering Close is generated as a result of a CNCL event, the reason code is MESH-PEERING-CANCELLED. When the Mesh Peering Close is generated as a result of a CLS_ACPT event, the reason code is MESH-CLOSE-RCVD.

11C.3.8.2 Mesh Peering Close frame processing

The mesh STA shall reject the Mesh Peering Close frame if the value in the Mesh ID element is not the same as the mesh STA's mesh profile. Otherwise, the mesh STA accepts the Mesh Peering Close frame and performs the actions described in 11C.4.

11C.4 Mesh peering management finite state machine (MPM FSM)

11C.4.1 General

Each mesh peering instance, including its states and resource, are managed by a mesh peering management finite state machine (MPM FSM). The MPM FSM uses MLME primitives to control the mesh STA to send and receive Mesh Peering Management frames.

11C.4.2 States

The MPM FSM uses the following six states:

- IDLE—IDLE state is a terminal state. In the IDLE state, the MPM FSM is ready to start a new mesh peering instance by either passively listening for an incoming Mesh Peering Open frame or actively initiating a mesh peering instance.
- OPN_SNT—In the OPN_SNT state, the finite state machine has sent a Mesh Peering Open frame and is waiting for a Mesh Peering Open frame and Mesh Peering Confirm frame from the candidate peer mesh STA.
- CNF_RCVD—In the CNF_RCVD state, the finite state machine has received a Mesh Peering Confirm frame, but has not received a Mesh Peering Open frame. The mesh STA has not sent the corresponding Mesh Peering Confirm frame yet.
- OPN_RCVD—In the OPN_RCVD state, the finite state machine has received only the Mesh Peering Open frame but not the Mesh Peering Confirm. The mesh STA has also sent a Mesh Peering Confirm frame upon receiving a Mesh Peering Open frame.
- ESTAB—In the ESTAB state, the finite state machine has received both the Mesh Peering Open and Mesh Peering Confirm frames. The mesh STA has also sent both the Mesh Peering Open frame and Mesh Peering Confirm frame. The mesh peering is established and configured for exchanging frames with the peer mesh STA in the ESTAB state.
- HOLDING—In the HOLDING state, the finite state machine is closing the mesh peering instance with the peer mesh STA or the candidate peer mesh STA.

11C.4.3 Events and actions

The finite state machine uses three types of events: 1) events for state machine transitions; 2) external events generated by frame processing; and 3) events associated with internal timers.

The events for state machine transitions are as follows:

- CNCL(localLinkID, peerMAC, ReasonCode)—Used to instruct the mesh peering instance to cancel the mesh peering with the peer mesh STA. localLinkID identifies the MPM FSM for the corresponding mesh peering instance. peerMAC is the MAC address of the peer mesh entity. ReasonCode is used to inform the reason to cancel the mesh peering instance. See 11C.3.8.2.
- ACTOPN(peerMAC, localLinkID)—The SME uses this event to create a new mesh peering instance to actively initiate the mesh peering establishment with the candidate peer mesh STA whose MAC address is peerMAC. localLinkID identifies the MPM FSM.

The events generated by frame processing are as follows:

- OPN_ACPT—PeeringOpen_Accept(peerMAC, peerLinkID) event indicates that a Mesh Peering Open frame meeting the correctness criteria of 11C.3.6 has been received from peerMAC for the mesh peering instance identified by peerLinkID.
- OPN_RJCT—PeeringOpen_Reject(peerMAC, peerLinkID, Configuration, reasonCode) event indicates that a Mesh Peering Open frame from peerMAC for the mesh peering instance identified by peerLinkID is rejected due to incomplete or erroneous configuration, as indicated by the

Configuration, with reasonCode being the specific reason for rejection of the Mesh Peering Open frame. See 11C.3.6.2.

- CNF_ACPT—PeeringConfirm_Accept(peerMAC, localLinkID, peerLinkID) event indicates that a Mesh Peering Confirm frame meeting the correctness criteria of 11C.3.7 has been received from peerMAC for the mesh peering instance identified by localLinkID and peerLinkID.
- CNF_RJCT—PeeringConfirm_Reject(peerMAC, localLinkID, peerLinkID, reasonCode) event indicates that a Mesh Peering Confirm frame from peerMAC for the mesh peering instance identified by localLinkID and peerLinkID is rejected due to incomplete or erroneous configuration, and reasonCode is the specific reason for rejection of the Confirm frame. See 11C.3.7.2.
- CLS_ACPT—PeeringClose_Accept(peerMAC, localLinkID, peerLinkID, reasonCode) event indicates that a Mesh Peering Close frame meeting the correctness criteria of 11C.3.8 has been received from peerMAC for the mesh peering instance identified by localLinkID and peerLinkID. The reasonCode specifies the reason that caused the generation of the Mesh Peering Close frame. See 11C.3.8.2.
- REQ_RJCT—PeeringRequest_Reject(peerMAC, peerLinkID, reasonCode) event indicates a special incidence that the mesh STA rejects the incoming Mesh Peering Open frame requesting to set up a new mesh peering for some specified reason. The incoming request is identified by the peerMAC, peerLinkID is the peerLinkID received from the Mesh Peering Open frame, and reasonCode is the specific reason for rejection of the Mesh Peering Open frame. See 11C.3.6.2.

The finite state machine may take an action triggered by an event. It uses two types of actions: sending a Mesh Peering Management frame and handling a timer.

Actions related to sending a Mesh Peering Management frame are as follows:

- sndOPN—sendOpen(peerMAC, localLinkID, Configuration) is the action that the mesh STA takes to send a Mesh Peering Open frame to the candidate peer mesh STA, whose MAC address is peerMAC. The MLME-MESHPEERINGMANAGEMENT.request primitive shall be invoked to send the frame to the peer mesh entity.
- sndCNF—sendConfirm(peerMAC, localLinkID, peerLinkID, Configuration) is the action that the mesh STA takes to send a Mesh Peering Confirm frame to the candidate peer mesh STA, whose MAC address is peerMAC. The MLME-MESHPEERINGMANAGEMENT.request primitive shall be invoked to send the frame to the peer mesh entity.
- sndCLS—sendClose(peerMAC, localLinkID, peerLinkID, reasonCode) is the action that the mesh STA takes to send a Mesh Peering Close frame to the peer mesh STA or candidate peer mesh STA, whose MAC address is peerMAC. The MLME-MESHPEERINGMANAGEMENT.request primitive shall be invoked to send the frame to the peer mesh entity.

11C.4.4 Timers

The following three timers are used by the finite state machine:

a) The retryTimer triggers a resend of the Mesh Peering Open frame when a Mesh Peering Confirm frame was not received as a response. The retryTimer is set to the dot11MeshRetryTimeout.

b) The confirmTimer signals that a link establishment attempt should be aborted because a Mesh Peering Confirm frame responding to a Mesh Peering Open frame was never received. The confirmTimer is set to the value of dot11MeshConfirmTimeout.

c) The holdingTimer signals that its mesh peering instance may be completely closed and facilitates graceful shutdown. The holdingTimer is set to the value of dot11MeshHoldingTimeout.

The events associated with internal timers are indicated in the state machine as acronyms that indicate timer expiry. With each timer event there is an associated action.

- TOR1—This event indicates that the retryTimer has expired and dot11MeshMaxRetries has not been reached. The Mesh Peering Open frame shall be resent, an action indicated in the state machine by setR.
- TOR2—This event indicates that the retryTimer has expired and dot11MeshMaxRetries has been reached. The mesh peering instance shall be closed when TOR2 occurs.
- TOC—This event indicates that the confirmTimer has expired. When TOC event occurs, the mesh peering instance shall be closed, an action indicated in the state machine as setC.
- TOH—This event indicates that the holdingTimer has expired. When TOH occurs, the mesh peering instance shall be closed and the finite state machine shall transition to IDLE state, an action indicated in the state machine as setH.

11C.4.5 State transitions

Table 11C-2 and Figure 11C-2 summarize the state transitions for the MPM protocol.

In Table 11C-2, each row represents state transitions from the state to all other states. A blank entry indicates an impossible transition.

Table 11C-2—MPM finite state machine

		To State					
		IDLE	OPN_SNT	CNF_RCVD	OPN_RCVD	ESTAB	HOLDING
From State	IDLE	REQ_RJCT / sndCLS	ACTOPN/ (sndOPN, setR)		OPN_ACPT / (sndOPN, sndCNF, setR)		
	OPN_SNT		TOR1/ (sndOPN, setR)	CNF_ACPT/ (clR, setC)	OPN_ACPT / (sndCNF)		CLS_ACPT, OPN_RJCT, CNF_RJCT, TOR2, CNCL/ (sndCLS, clR, setH)
	CNF_RCVD					OPN_ACPT / (clC, sndCNF)	CLS_ACPT, OPN_RJCT, CNF_RJCT, CNCL/ (sndCLS, clC, setH) TOC / (sndCLS, setH)
	OPN_RCVD				OPN_ACPT / sndCNF TOR1 / (sndOPN, setR)	CNF_ACPT / clR	CLS_ACPT, OPN_RJCT, CNF_RJCT, TOR2, CNCL/ (sndCLS, clR, setH)
	ESTAB					OPN_ACPT / sndCNF	CLS_ACPT, OPN_RJCT, CNF_RJCT, CNCL/ (sndCLS, setH)
	HOLDING	TOH/—, CLS_ACPT /clH					OPN_ACPT, CNF_ACPT, OPN_RJCT, CNF_RJCT/ sndCLS

In Figure 11C-2, each arrow represents a state transition.

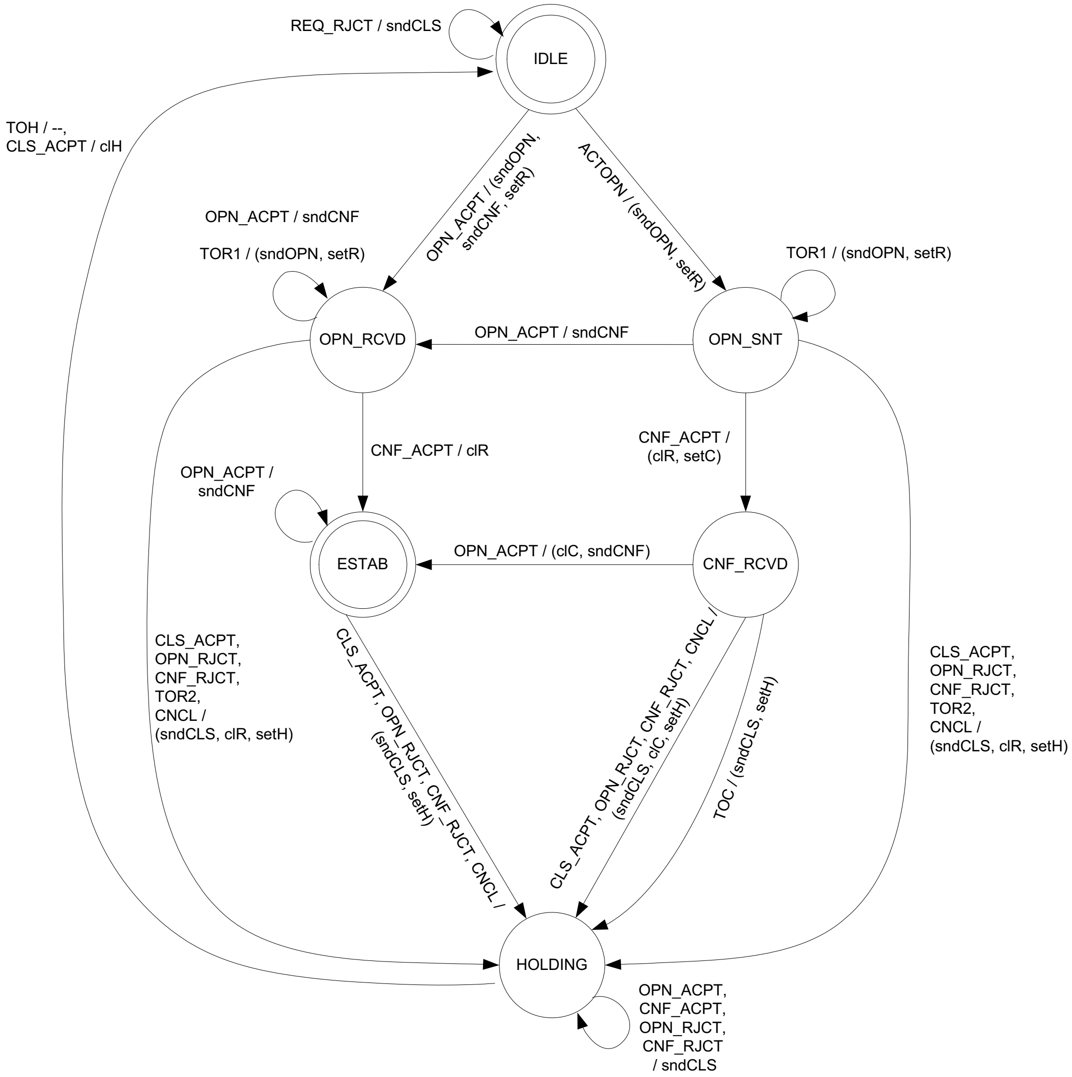

REQ_RJCT / sndCLS
IDLE
TOH / --,
CLS_ACPT / clH
OPN_ACPT / (sndOPN, sndCNF, setR)
ACTOPN / (sndOPN, setR)
OPN_ACPT / sndCNF
TOR1 / (sndOPN, setR)
TOR1 / (sndOPN, setR)
OPN_RCVD
OPN_ACPT / sndCNF
OPN_SNT
CNF_ACPT / clR
CNF_ACPT / (clR, setC)
OPN_ACPT / sndCNF
ESTAB
OPN_ACPT / (clC, sndCNF)
CNF_RCVD
CLS_ACPT, OPN_RJCT, CNF_RJCT, TOR2, CNCL / (sndCLS, clR, setH)
CLS_ACPT, OPN_RJCT, CNF_RJCT, CNCL / (sndCLS, setH)
CLS_ACPT, OPN_RJCT, CNF_RJCT, CNCL / (sndCLS, clC, setH)
TOC / (sndCLS, setH)
CLS_ACPT, OPN_RJCT, CNF_RJCT, TOR2, CNCL / (sndCLS, clR, setH)
HOLDING
OPN_ACPT, CNF_ACPT, OPN_RJCT, CNF_RJCT / sndCLS

Figure 11C-2—Finite state machine of the MPM protocol

The event/action representation is defined as the following. "E/A" string represents that the action A is taken given that the event E occurs. "E1, E2/A" string represents that the action A is taken given that the event E1 or event E2 occurs. "E/(A1, A2)" string represents that the action A1 and A2 are taken at a time when event E occurs.

Note that Table 11C-2 and Figure 11C-2 are used for illustration purpose. The protocol behavior is in the following subclauses.

11C.4.6 IDLE state

IDLE is a quiescent state the finite state machine enters prior to establishing a new mesh peering.

When ACTOPN event occurs, the mesh STA shall set the retryCounter to zero, and perform a sndOPN action. The retryTimer shall be set and the finite state machine shall transition to OPN_SNT state.

When an OPN_ACPT event occurs, the mesh STA shall perform a sndOPN action and sndCNF action, and set the retryTimer. The finite state machine shall transition to OPN_RCVD state.

When an REQ_RJCT event occurs, a Mesh Peering Close frame shall be sent to reject the mesh peering open request. The reason code in the Mesh Peering Close frame shall be set to the reason code in REQ_RJCT event. The finite state machine shall stay in the IDLE state.

All other events shall be ignored in this state.

11C.4.7 OPN_SNT state

In the OPN_SNT state, the mesh STA waits for a Mesh Peering Confirm frame. In this state, the retryTimer is set.

When a CNCL event occurs, the mesh STA shall clear the retryTimer, perform a sndCLS using the reason code specified by the CNCL event, and set the holdingTimer. The finite state machine shall transition to HOLDING state.

When a CLS_ACPT event occurs, the mesh STA shall clear the retryTimer, perform a sndCLS using the reason code specified by the CLS_ACPT event, and set the holdingTimer. The finite state machine shall transition to HOLDING state.

When an OPN_ACPT event occurs, the mesh STA shall send perform a sndCNF action. The finite state machine shall transition to OPN_RCVD state.

NOTE—The retryTimer is still in effect after the state transition.

When an OPN_RJCT event occurs, the mesh STA shall clear the retryTimer, perform a sndCLS using the reason code specified by the OPN_RJCT event, and set the holdingTimer. The finite state machine shall transition to HOLDING state.

When a CNF_ACPT event occurs, the mesh STA shall clear the retryTimer and shall set the confirmTimer and the finite state machine shall transition to CNF_RCVD state.

When a CNF_RJCT event occurs, the mesh STA shall clear the retryTimer, perform a sndCLS action using the reason code specified by the CNF_RJCT event, and set the holdingTimer. The finite state machine shall transition to HOLDING state.

When a TOR1 event occurs, the Mesh STA shall perform a sndOPN action and the retryCounter shall be incremented. The retryTimer shall be and the finite state machine shall stay in the OPN_SNT state.

When a TOR2 event occurs, the mesh STA shall perform a sndCLS action using the reason code MESH-MAX-RETRIES. The holdingTimer shall be set, and the finite state machine shall transition to HOLDING state.

All other events shall be ignored in this state.

11C.4.8 CNF_RCVD state

In the CNF_RCVD state, the mesh STA has received a Mesh Peering Confirm frame and is waiting for a Mesh Peering Open frame.

When a CNCL event occurs, the mesh STA shall clear the confirmTimer, perform a sndCLS action using the reason code MESH-PEERING-CANCELLED, and set the holdingTimer. The finite state machine shall transition to HOLDING state.

When a CLS_ACPT event occurs, the mesh STA shall clear the confirmTimer, perform a sndCLS using the reason code MESH-CLOSE-RCVD, and set the holdingTimer. The finite state machine shall transition to HOLDING state.

When an OPN_ACPT event occurs, the mesh STA shall clear the confirmTimer and shall perform a sndCNF action. The finite state machine shall transition to ESTAB state.

When an OPN_RJCT event occurs, the mesh STA shall clear the confirmTimer, perform a sndCLS action using the reason code as specified by the OPN_RJCT event, and set the holdingTimer. The finite state machine shall transition to HOLDING state.

When a CNF_RJCT event occurs, the mesh STA shall clear the confirmTimer, perform a sndCLS action using the reason code as specified by the CNF_RJCT event, and set the holdingTimer The finite state machine shall transition to HOLDING state.

When TOC event occurs, the mesh STA shall perform a sndCLS action using the reason code MESH-CONFIRM-TIMEOUT and set the holdingTimer. The finite state machine shall transition to HOLDING state.

All other events shall be ignored in this state.

11C.4.9 OPN_RCVD state

In the OPN_RCVD state, the mesh STA has received a Mesh Peering Open frame and sent a Mesh Peering Open frame and the corresponding Mesh Peering Confirm frame. An incoming Mesh Peering Confirm is expected.

When a CNCL event occurs, the mesh STA shall clear the retryTimer, perform a sndCLS action using the reason code MESH-PEERING-CANCELLED, and set the holdingTimer. The finite state machine shall transition to HOLDING state.

When a CLS_ACPT event occurs, the mesh STA shall clear the retryTimer, perform a sndCLS action, and set the holdingTimer. The finite state machine shall transition to HOLDING state.

When an OPN_ACPT event occurs, the mesh STA shall perform a sndCNF action. The finite state machine shall stay in the OPN_RCVD state.

When an OPN_RJCT event occurs, the mesh STA shall clear the retryTimer, perform a sndCLS action using the reason code as specified by the OPN_RJCT event, and set the holdingTimer. The finite state machine shall transition to HOLDING state.

When a CNF_ACPT event occurs, the retryTimer shall be cleared. The finite state machine shall transition to ESTAB state.

When a CNF_RJCT event occurs, the mesh STA shall clear the retryTimer, perform a sndCLS action using the reason code as specified by the CNF_RJCT event, and set the holdingTimer. The finite state machine shall transition to HOLDING state.

When a TOR1 event occurs, the Mesh STA shall perform a sndOPN action, increment the retryCounter, and set the retryTimer. The finite state machine shall stay in the OPN_RCVD state.

When a TOR2 event occurs, the mesh STA shall perform a sndCLS action using the reason code MESH-MAX-RETRIES. The holdingTimer shall be set, and the finite state machine shall transition to HOLDING state.

All other events shall be ignored in this state.

11C.4.10 ESTAB state

In the ESTAB state, mesh peering has been successfully established with the peer mesh STA.

When a CNCL event occurs, the mesh STA shall perform a sndCLS action using the reason code MESH-PEERING-CANCELLED, and set the holdingTimer. The finite state machine shall transition to HOLDING state.

When a CLS_ACPT event occurs, the mesh STA shall perform a sndCLS action using the reason code MESH-CLOSE-RCVD, and set the holdingTimer. The finite state machine shall transition to HOLDING state.

When an OPN_ACPT event occurs, the mesh STA shall respond by performing a sndCNF action. The finite state machine shall stay in the ESTAB state.

All other events shall be ignored in this state.

11C.4.11 HOLDING state

In HOLDING state, the mesh STA is closing the mesh peering. The holdingTimer has been set according to the value of dot11MeshHoldingTimeOut.

When a CLS_ACPT event occurs, the holdingTimer shall be cleared. The finite state machine shall transition to IDLE state.

When any of the following four events occurs—OPN ACPT, CNF_ACPT, OPN_RJCT, CNF_RJCT—the mesh STA shall send a Mesh Peering Close frame. The finite state machine shall stay in the HOLDING state.

When a TOH event occurs, the finite state machine shall transition to IDLE state.

All other events are ignored in this state.

11C.5 Authenticated mesh peering exchange (AMPE)

11C.5.1 Overview

The authenticated mesh peering exchange (AMPE) establishes an authenticated mesh peering between the mesh STAs, under the assumption that mesh PMKSA has already been established before the initiation of the protocol. An authenticated mesh peering includes a mesh peering, corresponding mesh TKSA, and the two mesh STAs mesh GTKSAs.

The AMPE uses Mesh Peering Management frames. Parameters are exchanged via the RSN element, the Authenticated Mesh Peering Exchange element, and the MIC element.

The major functions provided by AMPE are security capabilities selection, key confirmation, and key management.

- The security capabilities selection function (specified in 11C.5.2) is performed by agreeing on the security parameters used for the protocol instance.
- Key confirmation using the shared Mesh PMK is performed by verifying that the protection on the Mesh Peering Management frames is correct.
- Key management (specified in 8.8.1) is performed by the derivation of the temporal key in the mesh TKSA and the exchange of each mesh STA's MGTK.

During the AMPE handshake, the mesh STAs generate nonces and transmit them via Mesh Peering Management frames. The mesh STA shall generate a random value for its localNonce, as specified in 8.5.7. The candidate peer mesh STA is expected to generate a random value for the peerNonce, which the mesh STA receives from the candidate peer mesh STA in Confirm and Close Action frames.

Mesh Peering Management frames used in the AMPE are protected using the deterministic authenticated encryption mode of AES-SIV (IETF RFC 5297).

11C.5.2 Security capabilities selection

11C.5.2.1 Instance Pairwise Cipher Suite selection

Pairwise cipher suite selectors WEP-40, WEP-104, and TKIP shall not be used as the pairwise cipher suite when dot11MeshSecurityActivated is enabled.

If the pairwise cipher suite has not been selected, mesh STAs shall attempt to reach the agreement on the pairwise cipher suite using the following procedure in four steps:

a) The mesh STA shall announce the list of pairwise cipher suites it supports using an ordered list in the RSN element in the Mesh Peering Open frame. The first value in the list is the mesh STA's most preferred cipher suite, and the last value the least preferred.

b) If the mesh STA receives a Mesh Peering Open frame from the candidate peer mesh STA, the mesh STA shall make its decision on the selected pairwise cipher suite based on the intersection of its own ordered list and the received ordered list.

   1) If the intersection is empty, the pairwise cipher suite selection fails and the mesh STA generates the failure reason code MESH-INVALID-SECURITY-CAPABILITY and then takes the corresponding actions specified in 11C.5.6.

   2) If the intersection contains more than one value, the selected cipher suite shall be the entry in the intersection list most preferred by the mesh STA that has the largest MAC address in the lexicographic ordering.

c) If the mesh STA receives a Mesh Peering Confirm frame from the candidate peer mesh STA before receiving a Mesh Peering Open frame, the mesh STA shall verify that it supports the pairwise cipher suite chosen by the candidate peer mesh STA. Otherwise, the selection fails and the mesh STA shall generate the failure reason code MESH-INVALID-SECURITY-CAPABILITY.
   Furthermore, upon receiving a Mesh Peering Open frame, the mesh STA shall verify that the accepted selected pairwise cipher suite matches the pairwise cipher suite chosen in step b). If they do not match, the selection fails and the mesh STA shall generate the failure reason code MESH-INVALID-SECURITY-CAPABILITY. Otherwise, the pairwise cipher suite selection succeeds, and the mesh STA shall proceed to step d).

d) If the mesh STA is generating a Mesh Peering Confirm frame, it shall set the Selected Pairwise Cipher Suite to the selected pairwise cipher suite upon successful pairwise cipher suite selection.

11C.5.2.2 Group cipher suite selection

Group cipher suite selectors WEP-40, WEP-104, and TKIP shall not be used as the group cipher suite when dot11MeshSecurityActivated is true.

The mesh STA shall not use a different group cipher suite than the one used by the peer mesh STA or candidate peer mesh STA in the same MBSS.

A mesh STA shall announce in a Mesh Peering Open action frame the group cipher suite it uses for broadcast protection. When it receives a Mesh Peering Open frame from a candidate peer, it shall verify that it supports the candidate's announced group cipher suite. In addition, if the mesh STA receives a Mesh Peering Confirm frame, it shall verify that it supports the group cipher suite listed in that frame. If either selection fails, the mesh STA shall issue the appropriate reply frame with the MESH-INVALID-SECURITY-CAPABILITY reason code.

11C.5.3 Construction and processing AES-SIV-protected Mesh Peering Management frames

AES-SIV performs deterministic authenticated encryption and takes additional data that is authenticated but not encrypted (AAD). When encrypting and authenticating, AES-SIV takes a key, plaintext data to protect, and multiple distinct components of AAD, to produce a synthetic initialization vector and a ciphertext. When verifying encrypted and authenticated data AES-SIV takes a key, a synthetic initialization vector, ciphertext data to decrypt and verify, and AAD, to produce either plaintext or the symbol "FAIL", indicating failure to decrypt and verify. Note that the AAD used in the encryption process shall be identical to the AAD used in the decryption process and the synthetic initialization vector produced by the encryption process shall be used in the decryption process.

When the mesh STA constructs a Mesh Peering Management frame, it shall follow the following procedure:

- The input key shall be the AEK
- The input plaintext shall be the Authenticated Mesh Peering element (see 7.4.14.2, 7.4.14.3, 7.4.14.4)
- The input AAD shall be three distinct components consisting of
  1) The localMAC
  2) The peerMAC
  3) The contents of the Mesh Peering Management frame from the category (inclusive) to the MIC element (exclusive)
- The output synthetic initialization vector shall be copied into the MIC field of the MIC element in the Mesh Peering Management frame
- The output ciphertext shall become the remainder of the Mesh Peering Management frame after the MIC element

When the mesh STA verifies a Mesh Peering Management frame, it shall follow the following procedure:

- The input key shall be the AEK
- The input synthetic initialization vector shall be the MIC field of the MIC element in the Mesh Peering Management frame
- The input ciphertext shall be the part of the Mesh Peering Management frame following the MIC element
- The input AAD shall be three distinct components consisting of
  1) The peerMAC
  2) The localMAC
  3) The contents of the Mesh Peering Management frame from the category (inclusive) to the MIC element (exclusive)
- If AES-SIV returns the symbol "FAIL" processing of the frame shall be deemed a failure with a behavior dependent on the type of Mesh Peering Management frame

- If AES-SIV returns plaintext it shall be treated as the components of the Mesh Peering Management frame and processed accordingly

11C.5.4 MGTK distribution

The mesh STA shall distribute the MGTK to the peer mesh STA using the Mesh Peering Open frame during the AMPE. Upon successful completion of AMPE, each mesh STA shall establish states for the peer mesh STA's mesh GTKSA. The GTKData subfield in the Authenticated Mesh Peering Exchange element shall contain the MGTK concatenated by the Key RSC and the GTKExpirationTime (as indicated in 7.3.2.118).

11C.5.5 Mesh Peering Management frames for AMPE

11C.5.5.1 General

The AMPE is inclusive of the mesh peering management (MPM) protocol. Mesh Peering Management frames for AMPE have additional processing and construction requirements on top of those for Mesh Peering Management frames.

The Mesh Peering Management frames shall be generated with additional information using the RSN element and the Authenticated Mesh Peering Exchange element to support AMPE.

11C.5.5.2 Mesh peering open for AMPE

11C.5.5.2.1 Generating Mesh Peering Open frames for AMPE

In addition to contents for establishing a mesh peering as specified in 11C.3.6.1, the Mesh Peering Open frame, when used for the AMPE, shall contain the following:

- In the Mesh Peering Management element, the Mesh Peering Protocol Identifier shall be set to 1 "authenticated mesh peering exchange protocol."
- In the Mesh Peering Management element, the Chosen PMK field shall be set to PMKID that identifies the mesh PMKSA the mesh STA established with the candidate peer mesh STA.
- The RSN element shall be identical to the RSN element in the STA's Beacon and Probe Response frames.
- In the Authenticated Mesh Peering Exchange element:
  - The Selected Pairwise Cipher Suite field shall be set to the first cipher suite selector in the Pairwise Cipher Suite List field in RSN element.
  - The Local Nonce field shall be set to the localNonce value generated by the mesh STA for identifying the current mesh peering instance.
  - The Peer Nonce field shall be set to 0.
  - The GTKdata field shall be present and shall contain the data for the mesh STA's MGTK. The components of the GTKdata are specified in 11C.5.4.

The Mesh Peering Open frame shall be protected using AES-SIV as specified in 11C.5.3.

11C.5.5.2.2 Processing Mesh Peering Open frames for AMPE

On receiving a Mesh Peering Open frame, the mesh STA shall verify the received frame. If AES-SIV returns the symbol "FAIL" the OPN_RJCT event shall be invoked to the corresponding AMPE finite state machine and the reason code "MESH-INVALID-GTK" is generated. Otherwise, processing continues.

The received frame shall be rejected if the security capability selection fails (see 11C.5.2). The OPN_RJCT event shall be invoked to the corresponding AMPE finite state machine.

The peer mesh STA's MGTK extracted from the Mesh Peering Open frame shall be added to the Receive MGTK SA in which the peer's MAC address equals the MGTK Source mesh STA MAC address.

If all operations succeed, the mesh STA shall proceed to process the Mesh Peering Open frame on basic parameters as specified in 11C.3.6.2.

11C.5.5.3 Mesh peering confirm for AMPE

11C.5.5.3.1 Generating Mesh Peering Confirm frames for AMPE

In addition to contents for establishing a mesh peering as specified in 11C.3.7.1, the Mesh Peering Confirm frame, when used with the AMPE, shall contain the following:

- In the Mesh Peering Management element, the Mesh Peering Protocol Identifier shall be set to 1 "authenticated mesh peering exchange protocol."
- The RSN element shall be the same as sent in the Mesh Peering Open frame.
- In the Authenticated Mesh Peering Exchange element:
  - The Selected Pairwise Cipher Suite field shall be set to the cipher suite selector that indicates the successfully selected pairwise cipher suite (specified in 11C.5.2.1).
  - The Peer Nonce field shall be set to the nonce value chosen by the peer mesh STA as received in the Local Nonce field in the Mesh Peering Open frame from the candidate peer mesh STA.
  - The GTKdata field shall not be present.
  - The rest of fields are set to the same values sent in the Mesh Peering Open frame.

The Mesh Peering Confirm frame shall be protected using AES-SIV as specified in 11C.5.3.

11C.5.5.3.2 Processing Mesh Peering Confirm frames for AMPE

On receiving a Mesh Peering Confirm frame, the mesh STA shall verify the received frame. The received frame shall be discarded if AES-SIV returns the symbol "FAIL."

If AES-SIV returns plaintext, the following operations shall be performed in order:

a) The Selected Pairwise Cipher Suite is checked. If the security capability selection has been done and the received value from Chosen Pairwise Cipher Suite field is not the same as the agreed pairwise cipher suite, the mesh STA shall reject the received frame and the CNF_RJCT event is invoked to the corresponding AMPE finite state machine with the failure reason code MESH-INVALID-SECURITY-CAPABILITY.

b) The Group Cipher Suite is checked. If the received group cipher suite is not supported by the mesh STA, the mesh STA shall reject the received Mesh Peering Confirm frame and the CNF_RJCT event is invoked to the corresponding AMPE finite state machine with the failure reason code MESH-INVALID-SECURITY-CAPABILITY.

If none of the cases is true, the mesh STA shall proceed to process the Mesh Peering Confirm Action frame on basic parameters as specified in 11C.3.7.2.

11C.5.5.4 Mesh peering close for AMPE

11C.5.5.4.1 Generating Mesh Peering Close frames for AMPE

In addition to contents for closing a mesh peering as specified in 11C.3.8.1, the Mesh Peering Close frame, when used for the AMPE, shall contain the following:

- In the Mesh Peering Management element, the Mesh Peering Protocol Identifier shall be set to 1 "authenticated mesh peering exchange protocol."

- In the Mesh Peering Management element, the Chosen PMK field shall be set to the same value as sent in the Mesh Peering Open frame.
- In the Authenticated Mesh Peering Exchange element:
  - The Selected Pairwise Cipher Suite field shall be set to the same value as sent in the Mesh Peering Open frame.

    NOTE—If the reason for sending the Mesh Peering Close is the pairwise cipher suite selection failure, the information in this field is used to inform the candidate peer mesh STA what was announced by the mesh STA for the mesh peering instance.
  - The Local Nonce field shall be set to the same value as sent in the Mesh Peering Open frame.
  - The Peer Nonce field shall be set to the same value as received in the Local Nonce field of the Authenticated Mesh Peering Exchange element of the incoming Mesh Peering Management frame from the candidate peer mesh STA.

The Mesh Peering Close frame shall be protected using AES-SIV as specified in 11C.5.3.

11C.5.5.4.2 Processing Mesh Peering Close frames for AMPE

On receiving a Mesh Peering Close frame, the mesh STA shall verify the received frame. The received frame shall be discarded if AES-SIV returns the symbol “FAIL.”

If AES-SIV returns plaintext, the mesh STA shall proceed to process the Mesh Peering Close frame on basic parameters as specified in 11C.3.8.2.

11C.5.6 AMPE finite state machine

11C.5.6.1 Overview

The finite state machine for AMPE supports all the states, events, and actions defined for the finite state machine for the MPM protocol. In addition, new events, actions, and state transitions are added to specify the security functions for AMPE.

When a finite state machine is generated and activated for an AMPE instance, the localNonce shall be generated and used together with a new localLinkID to identify the instance.

11C.5.6.2 Additional events and actions to MPM FSM

All events for rejecting or ignoring received Action frames shall report the corresponding reason code related to AMPE functions as described in 11C.5.5.

In addition, there is one new event as follows:

- TOR3—This event indicates that the retryTimer has expired, the dot11MeshMaxRetries has been reached, the AMPE is enabled, but the mesh STA failed to confirm the selection of the shared mesh PMKSA. When this event triggers, the protocol instance shall be closed, but no Mesh Peering Close frame shall be sent.

The actions of sending Mesh Peering Management frames are updated as the following:

- sndOPN—Generate a Mesh Peering Open frame for the current AMPE protocol instance (as specified in 11C.5.5.2.1) and send it to the candidate peer mesh STA.
- sndCNF—Generate a Mesh Peering Confirm frame for the current AMPE protocol instance (as specified in 11C.5.5.3.1) and send it to the candidate peer mesh STA.
- sndClose—Generate a Mesh Peering Close frame for the current AMPE protocol instance (as specified in 11C.5.5.4.1) and send it to the candidate peer mesh STA.

11C.5.6.3 State transitions

All state transitions specified in MPM FSM shall be used for AMPE finite state machine.

In OPN_SNT state, the following are additional state transitions and actions:

> When TOR3 event occurs, the retryTimer shall be cleared and the holdingTimer shall be set. The finite state machine shall transition to HOLDING state.

In OPN_RCVD state, the following are the additional actions:

> When CNF_ACPT event occurs, in addition to the actions for MPM protocol, the mesh STA shall signal the completion of key management by utilizing the MLME-SETKEYS.request primitive to configure the agreed-upon mesh temporal pairwise key into the IEEE 802.11 MAC and by calling the MLME-SETPROTECTION.request primitive to enable its use.

In CNF_RCVD state, the following are the additional actions:

> When OPN_ACPT event occurs, in addition to the actions for MPM protocol, the mesh STA shall signal the completion of key management by utilizing the MLME-SETKEYS.request primitive to configure the agreed-upon mesh temporal pairwise key into the IEEE 802.11 MAC and received MGTK and by calling the MLME-SETPROTECTION.request primitive to enable the usage.

Table 11C-3 and Figure 11C-3 specify the state transitions of the finite state machine for AMPE.

Table 11C-3—AMPE finite state machine

		To State					
		IDLE	OPN_SNT	CNF_RCVD	OPN_RCVD	ESTAB	HOLDING
From State	IDLE	REQ_RJCT / sndCLS	ACTOPN/ (sndOPN, setR)		OPN_ACPT/ (sndOPN, sndCNF, setR)		
	OPN_SNT		TOR1/ (sndOPN, setR)	CNF_ACPT/ (clR, setC)	OPN_ACPT/ (sndCNF)		CLS_ACPT, OPN_RJCT, CNF_RJCT, TOR2, CNCL/ (sndCLS, clR, setH) TOR3 / (clR, setH)
	CNF_RCVD					OPN_ACPT / (clC, sndCNF)	CLS_ACPT, OPN_RJCT, CNF_RJCT, CNCL/ (snd-CLS, clC, setH) TOC / (snd-CLS, setH)
	OPN_RCVD				TOR1 / (sndOPN, setR) OPN_ACPT / sndCNF	CNF_ACPT / clR	CLS_ACPT, OPN_RJCT, CNF_RJCT,TOR2, CNCL/ (sndCLS, clR, setH)
	ESTAB					OPN_ACPT / sndCNF	CLS_ACPT, OPN_RJCT, CNF_RJCT,CNCL/ (snd-CLS, setH)
	HOLDING	TOH/—, CLS_ACPT / clH-					OPN_ACPT, CNF_ACPT, OPN_RJCT, CNF_RJCT / sndCLS

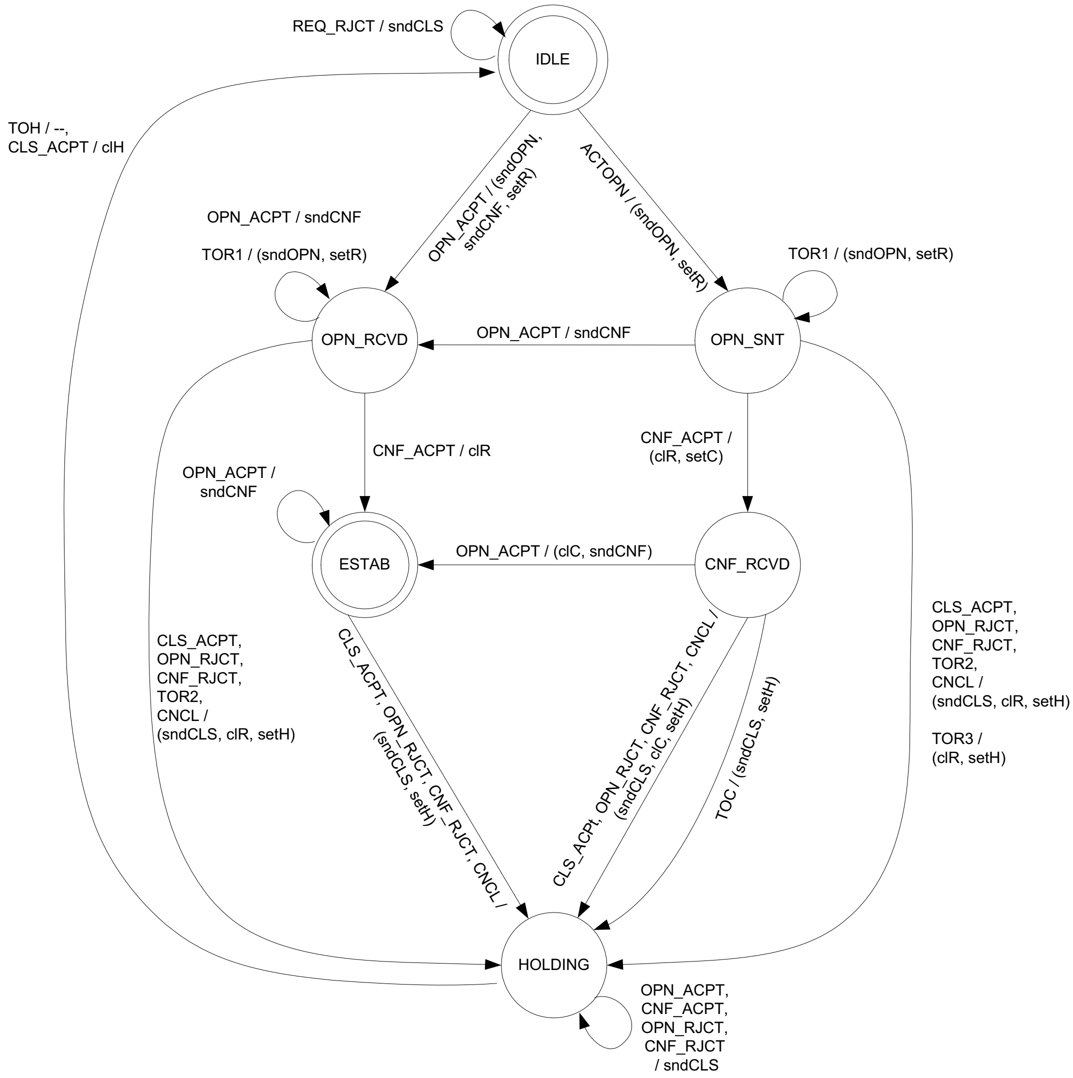

REQ_RJCT / sndCLS
IDLE
TOH / --,
CLS_ACPT / clH
OPN_ACPT / (sndOPN, sndCNF, setR)
ACTOPN / (sndOPN, setR)
OPN_ACPT / sndCNF
TOR1 / (sndOPN, setR)
TOR1 / (sndOPN, setR)
OPN_RCVD
OPN_ACPT / sndCNF
OPN_SNT
CNF_ACPT / clR
CNF_ACPT / (clR, setC)
OPN_ACPT / sndCNF
ESTAB
OPN_ACPT / (clC, sndCNF)
CNF_RCVD
CLS_ACPT, OPN_RJCT, CNF_RJCT, TOR2, CNCL / (sndCLS, clR, setH)
CLS_ACPT, OPN_RJCT, CNF_RJCT, CNCL / (sndCLS, setH)
CLS_ACPT, OPN_RJCT, CNF_RJCT, CNCL / (sndCLS, clC, setH)
TOC / (sndCLS, setH)
CLS_ACPT, OPN_RJCT, CNF_RJCT, TOR2, CNCL / (sndCLS, clR, setH)
TOR3 / (clR, setH)
HOLDING
OPN_ACPT, CNF_ACPT, OPN_RJCT, CNF_RJCT / sndCLS

Figure 11C-3—Finite state machine of the AMPE protocol

11C.6 Mesh group key handshake

11C.6.1 General

The mesh group key handshake may be used by either mesh STA, after a secure mesh peering has been established, to update the MGTK that it uses to protect group addressed MPDUs that it transmits to its peer mesh STAs.

The mesh STA may update its MGTK when a mesh peering is terminated.

To update the MGTK, the mesh STA shall execute the mesh group key handshake with each of its current peer mesh STAs. The “MGTK source” is the mesh STA that is sending the MGTK to a peer mesh STA using this protocol. A “MGTK recipient” is a mesh STA receiving the MGTK being sent by the MGTK Source.

The mesh group key handshake exchange shall include the following two messages:

- Message 1: Mesh Group Key Inform frame
- Message 2: Mesh Group Key Acknowledge frame

Mesh Group Key Inform frame and Mesh Group Key Acknowledge frame are conventionally referred to as "mesh group key handshake frames."

The mesh STA shall do an AMPE handshake before a mesh group key handshake if both are required to be done.

NOTE—It is impossible that the MGTK source initiates the mesh group key handshake before the authenticated mesh peering exchange (AMPE) completes successfully.

11C.6.2 Protection on mesh group key handshake frames

Mesh group key handshake frames used in mesh group key handshake are protected using the deterministic authenticated encryption mode of AES-SIV (RFC 5297) when dot11MeshSecurityActivated is true.

When constructing protection on mesh group handshake frames, the following procedure shall be used:

- The key shall be the AEK from the current active security association with the peer mesh STA that receives the mesh group key handshake frame.
- The input plaintext shall be the AMPE Authenticated Mesh Peering element (see 7.4.14.5 and 7.4.14.6).
- The plaintext shall be the Authenticated Mesh Peering Exchange element.
- AAD shall be three distinct components as follows:
  1) The localMAC
  2) The peerMAC
  3) The contents of the mesh group key handshake frame from the category (inclusive) to the MIC element (exclusive)
- The synthetic initialization vector produced by AES-SIV shall be copied into the MIC field of the MIC element in the frame.
- The produced ciphertext shall become the remainder of the mesh group key handshake frame after the MIC element.

When verifying the protection on the mesh group handshake frames, the following procedure shall be used:

- The key shall be the AEK from the current active security association with the peer mesh STA that receives the mesh group key handshake frame.
- AAD shall be three distinct components as follows:
  1) The peerMAC
  2) The localMAC
  3) The contents of the mesh group key handshake frame from the category (inclusive) to the MIC element (exclusive)
- The synthetic initialization vector shall be the MIC field of the MIC element in the frame.
- The ciphertext shall be the content after the MIC element in the frame.
- If AES-SIV validation function takes above input.
  - If the function returns the special symbol "FAIL," the frame shall be discarded.
  - If the plaintext is returned successfully, the produced plaintext shall be treated as the contents after the MIC element in the frame.

11C.6.3 Mesh Group Key Inform frame construction and processing

Mesh Group Key Inform frame shall be constructed as follows:

- The Authenticated Mesh Peering Exchange element shall be set as the following:
  - The Selected Pairwise Cipher Suite field shall be left blank.
  - The Local Nonce field shall be set to the same value as sent in the Mesh Peering Open frame that established the mesh peering instance.
  - The Peer Nonce field shall be set to the same value as received in the Local Nonce field of the Authenticated Mesh Peering Exchange element of the incoming Mesh Peering Open frame that established the peering instance.
  - The Key Replay Counter field shall be set to the mesh STA's local replay counter value, incremented by 1, for the mesh peering. After setting this field, the local replay counter shall also be incremented by 1.
  - The GTKdata field shall be present and shall contain the data for the MGTK from MGTK source. The components of the GTKdata are specified in 11C.5.4.
- The MIC element shall be set according to the protection mechanism in 11C.6.2.

The construction of AES-SIV protection on Mesh Group Key Inform frame shall use the construction procedure as in 11C.6.2.

The MGTK source sends the Mesh Group Key Inform frame to the MGTK recipient.

On reception of Mesh Group Key Inform frame, the MGTK recipient shall use the verification procedure in 11C.6.2 to validate the AES-SIV construction.

- If the validation recovers the plaintext successfully, the MGTK recipient shall proceed with the following procedure:
  - Verify that values in the Local Nonce field and the Peer Nonce field in the Authenticated Mesh Peering Exchange element are the same as in the current valid mesh TKSA that the MGTK recipient established with the sender of the Mesh Group Key Inform frame. If there is any mismatch, the received Mesh Group Key Inform frame shall be discarded and no further action shall be taken.
  - Verify that the Key Replay Counter has not yet been seen before, i.e., its value is strictly larger than that in any other mesh Group Key Inform frame received thus far during this security association. If this verification fails, the received Mesh Group Key Inform frame shall be discarded and no further action shall be taken.
  - Use the MLME-SETKEYS.request primitive to configure the temporal MGTK into its IEEE 802.11 MAC.
  - Respond by constructing and sending mesh group key handshake acknowledge to the MGTK source and incrementing the replay counter.

  NOTE—The MGTK source increments and uses a new Key Replay Counter field value on every Mesh Group Key Inform frame, even retries, because the Mesh Group Key Acknowledge responding to an earlier Mesh Group Key Inform frame might have been lost. If the MGTK source did not increment the replay counter, the MGTK receiver discards the retry, and no responding Mesh Group Key Acknowledge frame will ever arrive.
- If the AES-SIV validation returns a special symbol "FAIL", the Mesh Group Key Inform frame shall be discarded. No further action shall be taken.

11C.6.4 Mesh Group Key Acknowledge frame construction and processing

Mesh Group Key Acknowledge frame shall be constructed as follows:

- The Authenticated Mesh Peering Exchange element shall be set as follows:
  - The Selected Pairwise Cipher Suite field shall be left blank.

  - The Local Nonce field shall be set to the same value as sent in the Mesh Peering Open frame that established the mesh peering instance.
  - The Peer Nonce field shall be set to the same value as received in the Local Nonce field of the Authenticated Mesh Peering Exchange element of the incoming Mesh Peering Open frame that established the peering instance.
  - The Key Replay Counter shall be set to the same value as received in the Mesh Group Key Inform frame.
  - The GTKdata field shall be blank.
- The MIC element shall be set according to the protection mechanism in 11C.6.2.

The construction of AES-SIV protection on Mesh Group Key Acknowledge frame shall use the construction procedure as in 11C.6.2.

The MGTK recipient sends the Mesh Group Key Acknowledge frame to the MGTK source.

On reception of Mesh Group Key Acknowledge frame, the MGTK source shall use the verification procedure in 11C.6.2 to validate the AES-SIV construction.

- If the validation recovers the plaintext successfully, the MGTK source shall set the content of the Authenticated Mesh Peering Exchange element using the recovered plaintext and proceed with the following procedure:
  - Verify that values in the Local Nonce field and the Peer Nonce field in the Authenticated Mesh Peering Exchange element are the same as in the current valid mesh TKSA that the MGTK source established with the sender of the Mesh Group Key Acknowledge frame. If there is any mismatch, the received Mesh Group Key Acknowledge frame shall be discarded and no further action shall be taken.
  - Verify that the Key Replay Counter value matches the one that it has used for the mesh group key handshake. If this verification fails, the received Mesh Group Key Acknowledge frame shall be discarded and the MGTK source may invoke a retry to send a new Mesh Group Key Inform frame with a new Key Replay Counter value.
- If the validation returns a special symbol “FAIL,” the Mesh Group Key Acknowledge frame shall be discarded and the MGTK source may invoke a retry to send a new Mesh Group Key Inform frame with a new Key Replay Counter value.

11C.6.5 Mesh group key implementation considerations

If the MGTK source does not receive a Mesh Group Key Acknowledge frame to its Mesh Group Key Inform frames, it shall attempt dot11MeshConfigGroupUpdateCount additional transmissions of the Mesh Group Key Inform frame. The retransmit timeout value shall be 100 ms for the first timeout, half the listen interval for the second timeout, and the listen interval for subsequent timeouts. If there is no listen interval, then 100 ms shall be used for all timeout values. If it still has not received a response after this, then the MGTK source shall tear down the mesh peering and mesh TKSA with this MGTK recipient, by generating a CNCL event for the peering instance and pass the event to the mesh peering instance controller.

11C.7 Mesh path selection and metric framework

11C.7.1 General

The term “mesh path selection” is used to describe selection of multi-hop paths between mesh STAs at the link layer. Mesh path selection creates forwarding information that is utilized for MSDU/MMPDU forwarding as described in 9.22.

11C.7.2 Extensible path selection framework

This standard allows for alternative and flexible implementations of path selection protocols and metrics.

A mesh STA may include multiple protocol implementations (that is, the default protocol, vendor-specific protocols, etc.) as well as multiple metric implementations, but only one path selection protocol and only one path selection metric shall be used by a mesh STA at a time.

As described in 11C.2.3 and 11C.2.7, mesh STAs use the Mesh Configuration element (7.3.2.98) to announce the active path selection protocol and active path selection metric of the MBSS. This allows a neighbor mesh STA to identify if it should become a member of the MBSS and how it should establish mesh peerings with its members. This standard does not force an existing MBSS that is using a protocol other than the default protocol to switch to the default protocol when a new mesh STA requests mesh peering establishment. While it is possible, in principle, to implement such behavior, an algorithm to coordinate such reconfiguration is beyond the scope of this standard.

Path selection protocol and path selection metric are identified by a unique identifier as defined in 7.3.2.98.2 and 7.3.2.98.3, respectively. Also, each path selection protocol and each path selection metric specifies the following:

- Data type of metric values
- Length of the metric field
- Operator for aggregation of link metrics to a path metric; the symbol $\oplus$ is used to identify an arbitrary operator for aggregation
- Comparison operator for determining a better or worse path; how this is performed depends on the actual comparison operator
- Initial value of the path metric (path selection metric only)

The standard defines a default mandatory path selection protocol (HWMP, 11C.9) and a default mandatory path selection metric (airtime link metric, 11C.8). Both shall be implemented on all mesh STAs to ensure interoperability.

11C.7.3 Link metric reporting

A mesh STA may submit a link metric report to or request a link metric report from its neighbor peer mesh STA by transmitting a Mesh Link Metric Report frame. A mesh STA receiving a Mesh Link Metric Report element with the Request subfield of the Flags field equal to 1 shall reply with a Mesh Link Metric Report frame containing the link metric value for the corresponding link.

Upon reception of a Mesh Link Metric Report frame, the mesh STA may update its local link metric information using the link metric information received. The procedure to update the local link metric information with the link metric information received from a neighbor peer mesh STA is outside the scope of the standard.

11C.8 Airtime link metric

This subclause defines a default link metric that may be used by a path selection protocol to identify an efficient radio-aware path. The extensibility framework allows this metric to be overridden by any path selection metric as specified in the mesh profile.

Airtime reflects the amount of channel resources consumed by transmitting the frame over a particular link. This measure is approximate and designed for ease of implementation and interoperability.

The airtime for each link is calculated as follows:

$$c_a = \left[O + \frac{B_t}{r}\right]\frac{1}{1 - e_f}$$

Where O and B_t are constants listed in Table 11C-4, and the input parameters r and e_f are the data rate in Mb/s and the frame error rate for the test frame size B_t respectively. The rate r represents the data rate at which the mesh STA would transmit a frame of standard size B_t based on current conditions and its estimation is dependent on local implementation of rate adaptation. The frame error rate e_f is the probability that when a frame of standard size B_t is transmitted at the current transmission bit rate r, the frame is corrupted due to transmission error; its estimation is a local implementation choice. Frame failures due to exceeding Mesh TTL should not be included in this estimate as they are not correlated with link performance.

The airtime link metric shall be encoded as an unsigned integer in units of 0.01 TU.

Table 11C-4—Airtime cost constants

Parameter	Recommended value	Description
O	Varies depending on PHY	Channel access overhead, which includes frame headers, training sequences, access protocol frames, etc.
B_t	8192	Number of bits in test frame

Table 11C-5 gives the parameters of the airtime link metric for the extensible path selection framework.

Table 11C-5—Parameters of the airtime link metric for extensible path selection framework

Path Selection Metric ID	See Table 7-43bj2 in 7.3.2.98.3
Data type	Unsigned integer, 0 ≤ metric value< 4 294 967 296
Length of metric field	4 octets
Operator for metric aggregation	addition (+)
Comparison operator	*less than, equal to, greater than* as used with integers — metric a is *better than* metric b iff $a < b$ — metric a is *equal to* metric b iff $a = b$ — metric a is *worse than* metric b iff $a > b$
Initial value of path metric	0

An example of the airtime link metric is shown in Y.5.

11C.9 Hybrid wireless mesh protocol (HWMP)

11C.9.1 General

The hybrid wireless mesh protocol (HWMP) is a mesh path selection protocol that combines the flexibility of on-demand path selection with proactive topology tree extensions. The combination of reactive and proactive elements of HWMP enables efficient path selection in a wide variety of mesh networks (with or without access to the infrastructure).

HWMP uses a common set of protocol elements, generation and processing rules inspired by Ad Hoc On-Demand Distance Vector (AODV) protocol (IETF RFC 3561) adapted for MAC address-based path selection and link metric awareness. HWMP is completely specified herein and does not require reference to AODV specifications or descriptions.

HWMP supports two modes of operation depending on the configuration. These modes provide different levels of functionality as follows:

— On-demand mode: The functionality of this mode is always available, independent of whether a root mesh STA is configured in the MBSS or not. It allows mesh STAs to communicate using peer-to-peer paths.

— Proactive tree building mode: In this mode, additional proactive tree building functionality is added to the on-demand mode. This can be performed by configuring a mesh STA as root mesh STA using either the proactive PREQ or RANN mechanism.The proactive PREQ mechanism creates paths from the mesh STAs to the root, using only group-addressed communication. The RANN mechanism creates paths between the root and each mesh STA using acknowledged communication.

These modes are not exclusive. On-demand and proactive modes are used concurrently, because the proactive modes are extensions of the on-demand mode.

NOTE—One example of concurrent usage of on-demand and proactive mode is for two mesh STAs that are part of the same mesh BSS (or STAs that are proxied by mesh STAs in the same MBSS) to begin communicating using the proactively built tree but subsequently to perform an on-demand discovery for a direct path. This type of concurrent usage of the proactive and on-demand modes allows communication to begin immediately (by forwarding all traffic to the root, which knows all mesh STAs and addresses proxied by mesh STAs in the MBSS) while an on-demand discovery finds a shorter path between two mesh STAs (or STAs that are proxied by mesh STAs in the same MBSS).

All HWMP modes of operation utilize common processing rules and primitives. HWMP elements are the path request (PREQ), path reply (PREP), path error (PERR), and root announcement (RANN). The metric cost of the links determines which paths HWMP builds. In order to propagate the metric information between mesh STAs, a Metric field is used in the PREQ, PREP, and RANN elements.

Path selection in HWMP uses a sequence number mechanism to ensure that mesh STAs can distinguish current path information from stale path information at all times in order to maintain loop-free connectivity. Each mesh STA maintains its own HWMP sequence number, which is propagated to other mesh STAs in the HWMP elements. Rules for maintaining HWMP sequence numbers are given in 11C.9.8.3.

11C.9.2 Terminology

This subclause describes terminology for HWMP, especially for the process of path discovery. Terms such as Path Originator or Path Target designate very specific entities within the path discovery process. They stay with the same assigned entity for the whole path discovery process and other procedures related to this path discovery. Figure 11C-4 illustrates an example utilizing this terminology.

NOTE—Both the path target and path originator are a path destination for the forward path and the reverse path respectively.

The following terms are used within the context of a single PREQ/PREP pair, a so-called HWMP path discovery:

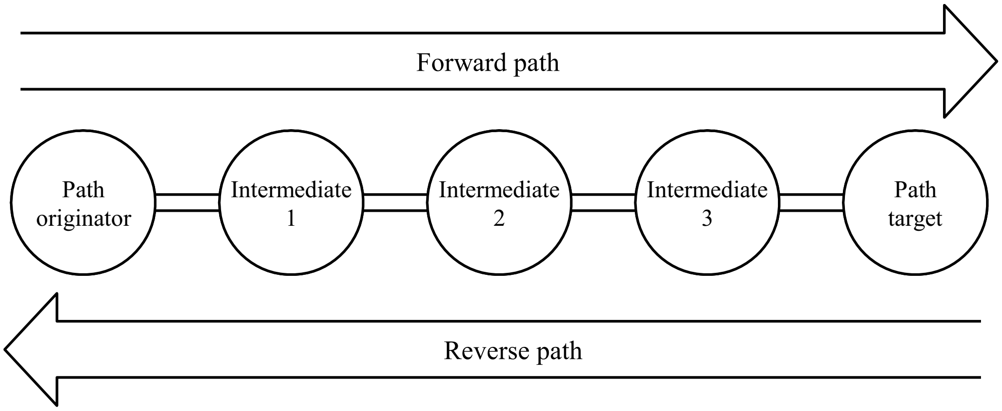


Figure 11C-4—Illustration of definitions

- **path originator:** The path originator is the mesh STA that triggers the path discovery.
- **path originator address:** The MAC address of the path originator.
- **path target:** The path target is the entity to which the path originator attempts to establish a path.

  NOTE—When an originator mesh STA initially attempts to establish a path to a target, it does not know whether the target is a mesh STA in the mesh BSS or not. Only when the Originator receives a PREP does it learn if the target is a mesh STA in the mesh BSS or not. If the target is in the mesh BSS, it is referred to as a target mesh STA. If the target is outside the mesh BSS, the term target proxy mesh gate refers to the mesh gate proxying for the target.

- **path target address:** The MAC address of the path target.
- **intermediate mesh STA:** The intermediate mesh STA is the mesh STA that participates in path selection and is neither path originator nor path target.
- **intermediate mesh STA address:** The MAC address of the intermediate mesh STA.
- **forward path:** The forward path is the mesh path to the path target, set up at the path originator and intermediate mesh STAs.
- **reverse path:** The reverse path is the mesh path to the path originator, set up at the path target and intermediate mesh STAs.
- **HWMP Sequence Number (HWMP SN):** Each mesh HWMP path selection element contains an HWMP sequence number that allows recipients to distinguish newer from stale information. An HWMP sequence number is specific to a mesh STA. See also 11C.9.8.3.
- **forwarding information:** The forwarding information maintained by an originator mesh STA, an intermediate mesh STA, or a target mesh STA that allows the mesh STA to perform its path selection and forwarding functions.

  The terminology used when discussing forwarding information is relative to the mesh STA (reference mesh STA, given mesh STA or local mesh STA) and a particular mesh destination of the path. The following terms are specific to a given instance of the forwarding information:

  - **destination mesh STA:** The end station (mesh STA) of a (forward or reverse) path.
  - **destination mesh STA address:** The MAC address of the destination mesh STA.
  - **destination HWMP sequence number:** The HWMP sequence number of the destination mesh STA.
  - **next-hop mesh STA:** The next-hop mesh STA is the next peer mesh STA on the mesh path to the destination mesh STA.
  - **next-hop mesh STA address:** The MAC address of the next-hop mesh STA.

  - — **precursor mesh STA:** A precursor mesh STA is a neighbor peer mesh STA on the mesh path that identifies a given mesh STA as the next-hop mesh STA to the destination mesh STA.
  - — **precursor mesh STA address:** The MAC address of the precursor mesh STA.
  - — **lifetime:** The time during which forwarding information remains active (see 11C.9.8.4)

- — **unreachable destination:** A destination mesh STA is considered unreachable by a source mesh STA or an intermediate mesh STA if the link to the next hop of the mesh path to this destination mesh STA, as derived from its forwarding information, is no longer usable.
- — **element time to live (Element TTL)**: An integer number that is used to limit the number of hops an HWMP element may be processed and propagated. Note that this Element TTL is different from the Mesh TTL in the Mesh Control field (see 7.1.3.6.3).
- — **root mesh STA:** A root mesh STA is configured to originate pro-active PREQs or RANNs. It is the root of a path selection tree.

Table 11C-6 and Table 11C-7 shows the roles of the various mesh STAs in the forward path and reverse path generated as a result of the full PREQ and PREP processing as shown in Figure 11C-4. Each row in the table contains the roles of a forward/reverse path from the reference mesh STA's perspective.

Table 11C-6—Precursor and next hop examples (forward path)

Forward path (to Path Target)			
Reference mesh STA	**Precursor mesh STA**	**Next-hop mesh STA**	**Destination mesh STA**
Path Originator	N/A	Intermediate 1	Path Target
Intermediate 2	Intermediate 1	Intermediate 3	Path Target
Path Target	Intermediate 3	N/A	Path Target

Table 11C-7—Precursor and next hop examples (reverse path)

Reverse path (to Path Originator)			
Reference mesh STA	**Precursor mesh STA**	**Next-hop mesh STA**	**Destination mesh STA**
Path Originator	Intermediate 1	N/A	Path Originator
Intermediate 2	Intermediate 3	Intermediate 1	Path Originator
Path Target	N/A	Intermediate 3	Path Originator

11C.9.3 On-demand path selection mode

If a source mesh STA needs to find a path to a destination mesh STA using the on-demand path selection mode, it broadcasts a PREQ with the path target specified in the list of targets and the metric field initialized to the initial value of the active path selection metric.

When a mesh STA receives a new PREQ, it creates or updates its path information to the originator mesh STA and propagates the PREQ to its neighbor peer mesh STAs if the PREQ contains a greater HWMP sequence number, or the HWMP sequence number is the same as the current path and the PREQ offers a better metric than the current path. Each mesh STA may receive multiple copies of the same PREQ that originated at the originator mesh STA, each PREQ traversing a unique path.

Whenever a mesh STA propagates a PREQ, the metric field in the PREQ is updated to reflect the cumulative metric of the path to the originator mesh STA. After creating or updating a path to the originator mesh STA, the target mesh STA sends an individually addressed PREP back to the originator mesh STA.

If the mesh STA that received a PREQ is the target mesh STA, it sends an individually addressed PREP back to the originator mesh STA after creating or updating a path to the originator mesh STA.

The PREQ provides the TO (Target Only) subfield that allows path selection to take advantage of existing paths to the target mesh STA by allowing an intermediate mesh STA to return a PREP to the originator mesh STA. If the TO (Target Only) subfield is 1, only the target mesh STA responds with a PREP. The effect of setting the TO (Target Only) subfield to 0 is the quick establishment of a path using the PREP generated by an intermediate mesh STA, allowing the forwarding of MSDUs with a low path selection delay. In order to select (or validate) the best path during the path selection procedure, the intermediate mesh STA that responded with a PREP propagates the PREQ with the TO (Target Only) subfield set to 1. This prevents all other intermediate mesh STAs on the way to the target from sending a PREP.

Intermediate mesh STAs create a path to the target mesh STA on receiving the PREP, and also forward the PREP toward the originator. When the originator receives the PREP, it creates a path to the target mesh STA. If the target mesh STA receives further PREQs with a better metric, then the target updates its path to the originator with the new path and also sends a new PREP to the originator along the updated path. A bidirectional, best metric end-to-end path is established between the originator and target mesh STA.

11C.9.4 Proactive tree building mode

11C.9.4.1 General

There are two mechanisms for proactively disseminating path selection information for reaching the root mesh STA. The first method uses a *proactive* Path Request (PREQ) element and is intended to create paths between all mesh STAs and the root mesh STA in the network proactively. The second method uses a Root Announcement (RANN) element and is intended to distribute path information for reaching the root mesh STA but there is no forwarding information created.

A mesh STA configured as root mesh STA sends either proactive PREQ or RANN elements periodically.

11C.9.4.2 Proactive PREQ mechanism

The PREQ tree building process begins with a proactive PREQ element sent by the root mesh STA, with the Target Address set to all ones and the TO subfield set to 1. The PREQ contains the path metric (set to the initial value of the active path selection metric by the root mesh STA) and an HWMP sequence number. The proactive PREQ is sent periodically by the root mesh STA, with increasing HWMP sequence numbers.

A mesh STA receiving a proactive PREQ creates or updates its forwarding information to the root mesh STA, updates the metric and hop count of the PREQ, records the metric and hop count to the root mesh STA, and then transmits the updated PREQ. Information about the presence of and distance to available root mesh STA(s) is disseminated to all mesh STAs in the network.

Each mesh STA may receive multiple copies of a proactive PREQ, each traversing a unique path from the root mesh STA to the mesh STA. A mesh STA updates its current path to the root mesh STA if and only if the PREQ contains a greater HWMP sequence number, or the HWMP sequence number is the same as the current path and the PREQ offers a better metric than the current path to the root mesh STA. The processing of the proactive PREQ is the same as the processing of the PREQ in the on-demand mode described in 11C.9.3.

If the proactive PREQ is sent with the Proactive PREP subfield set to 0, the recipient mesh STA may send a proactive PREP. A proactive PREP is necessary, for example, if the mesh STA has data to send to the root mesh STA, thus requiring the establishment of a forward path from the root mesh STA. During the time the forward path is required, the recipient mesh STA shall send a proactive PREP even if the Proactive PREP subfield is set to 0. Guidance on controlling the generation of proactive PREQs in such a case is given in Y.6.

If the PREQ is sent with a Proactive PREP subfield set to 1, the recipient mesh STA shall send a proactive PREP. The proactive PREP establishes the path from the root mesh STA to the mesh STA.

11C.9.4.3 Proactive RANN mechanism

The root mesh STA periodically propagates a RANN element into the network. The information contained in the RANN is used to disseminate path metrics to the root mesh STA, but reception of a RANN does not establish a path.

Upon reception of a RANN, each mesh STA that has to create or refresh a path to the root mesh STA sends an individually addressed PREQ to the root mesh STA via the mesh STA from which it received the RANN.

The root mesh STA sends a PREP in response to each PREQ. The individually addressed PREQ creates the reverse path from the root mesh STA to the originator mesh STA, while the PREP creates the forward path from the mesh STA to the root mesh STA.

11C.9.5 Collocated STAs

HWMP terminology strictly refers to a STA whose address is used for destination mapping (i.e., the originator address, the intermediate mesh STA address, or the target address). HWMP terminology does not make any assumption about the address used for communication over the WM (i.e., the Transmitter Address and the Receiver Address). For example, there is no requirement that the Path Originator or the Path Target of an HWMP path are the same as the address used for transmitting the first and receiving the last (respectively) HWMP Mesh Path Selection frame containing a PREQ.

The corollary to this is that the first hop of a PREQ may have a transmitter address that is not the same as the Originator address in the PREQ element and that the first hop may have a transmitter address that is not the same as the source address. In order to determine whether a transmission is a first hop or not, mesh STAs should not compare the source and transmitter addresses. Instead, this determination can be made by looking at the hop count field of the PREQ element (which is not used as an acceptance criterion).

11C.9.6 Parameters for extensible path selection framework

Table 11C-8 gives the parameters of HWMP for the extensible path selection framework (see 11C.7.2).

Table 11C-8—Parameters of HWMP for extensible path selection framework

Path Selection Protocol ID	See Table 7-43bj1 in 7.3.2.98.2
Data type of metric field	As defined by active path selection metric
Length of metric field	4 octets
Operator for metric aggregation	As defined by active path selection metric
Comparison operator	As defined by active path selection metric
Initial value of path metric	As defined by active path selection metric

11C.9.7 Addressing of HWMP Mesh Path Selection frame

All HWMP elements are sent in an HWMP Mesh Path Selection frame (see 7.4.15.3). The RANN element may also be sent in a Beacon frame. "Cases" refer to the different conditions that trigger the transmission of an HWMP Mesh Path Selection frame. PREQ cases are specified in 11C.9.9.3. PREP cases are specified in 11C.9.10.3. PERR cases are specified in 11C.9.11.3. RANN cases are specified in 11C.9.12.3.

Note that the PREQ Addressing Mode subfield in the Flags field identifies the propagation mode of the PREQ; an Addressing Mode subfield of 0 indicates that the PREQ is group addressed, an Addressing Mode subfield of 1 indicates that the PREQ is individually addressed.

The addresses of the HWMP Mesh Path Selection frame shall be as follows:

- PREQ group addressed—Addressing Mode subfield = 0 [Case A: Path Discovery (Original transmission), Case B: Path Maintenance (Original transmission), Case C: Proactive PREQ (Original transmission), and Case E: PREQ Propagation]:
  - Address 1: Group address
  - Address 2: Address of the mesh STA sending the PREQ
  - Address 3: Same as Address 2
- PREQ individually addressed—Addressing Mode subfield = 1 [Case D: Root Path Confirmation (Original transmission)]:
  - Address 1: Address 2 of the frame containing the RANN element that triggered the PREQ
  - Address 2: Address of the mesh STA sending the PREQ
  - Address 3: Same as Address 2
- PREQ individually addressed—Addressing Mode subfield = 1 [Case E: PREQ Propagation]:
  - Address 1: Next-hop MAC address to the mesh STA identified as the Target MAC address in the PREQ element
  - Address 2: Address of the mesh STA sending the PREQ
  - Address 3: Same as Address 2
- PREP Case A: Original transmission, Case C: Intermediate reply, Case D: Proactive PREP in Proactive PREQ mode:
  - Address 1: Address of the next hop to the Originator Mesh STA Address in the PREQ that triggered the PREP
  - Address 2: Address of the mesh STA sending the PREP
  - Address 3: Same as Address 2
- PREP Case B: PREP Propagation:
  - Address 1: Address of the next hop to the Originator Mesh STA Address in the PREP that triggered the PREP
  - Address 2: Address of the mesh STA sending the PREP
  - Address 3: Same as Address 2
- PERR individually addressed [Case A: Original transmission—next hop is unusable]:
  - Address 1: Address of each one of the precursors for which the active forwarding information has been invalidated [see Case A, 11C.9.11.3]
  - Address 2: Address of the mesh STA sending the PERR
  - Address 3: Same as Address 2
- PERR individually addressed [Case B: Original transmission—missing forwarding information]:
  - Address 1: Address of the transmitter of the frame that triggered the PERR (see Case B, 11C.9.11.3)
  - Address 2: Address of the mesh STA sending the PERR
  - Address 3: Same as Address 2
- PERR individually addressed [Case C: Original transmission (proxy information is unusable)]:

  - Address 1: Address of each one of the neighbor peer mesh STAs
  - Address 2: Address of the mesh STA sending the PERR
  - Address 3: Same as Address 2
- PERR individually addressed [Case D: PERR propagation]:
  - Address 1: Address of each one of the precursors for which the active forwarding information has been invalidated (see 11C.9.11.4.3)
  - Address 2: Address of the mesh STA sending the PERR
  - Address 3: Same as Address 2
- PERR group addressed [all cases]:
  - Address 1: group address
  - Address 2: Address of the mesh STA sending the PERR
  - Address 3: Same as Address 2
- RANN all cases:
  - Address 1: group address
  - Address 2: Address of the mesh STA sending the RANN
  - Address 3: Same as Address 2

Multiple HWMP elements may be sent in the same HWMP Mesh Path Selection frame if they share the same intended Address 1.

11C.9.8 General rules for processing HWMP elements

11C.9.8.1 General

This subclause describes the rules for the processing of the following components of the HWMP elements:

- HWMP Sequence Number
- Element TTL
- Metric

11C.9.8.2 HWMP propagation

The term “propagate” is used to describe the means by which elements are not transmitted “as is” across the network but are processed and modified along the way. Many HWMP elements are intended to be processed and propagated across an MBSS by mesh STAs. Each propagation is subject to certain rules or limitations as explained in the following subclauses. Certain parameters in the HWMP elements are updated during the propagation. See 11C.9.9, 11C.9.10, 11C.9.11, and 11C.9.12.

The originator of an HWMP element sets the initial value of the Element TTL. The mesh STA that receives the HWMP element shall propagate it if the received value of Element TTL is greater than 1. Before propagating the HWMP element, the mesh STA decrements the Element TTL value.

In general, the propagation of an HWMP element is not subject to a delay. Exception exists for the RANN element as described in 11C.9.12.

11C.9.8.3 HWMP sequence numbering

HWMP uses sequence numbers to prevent the creation of path loops and to distinguish stale and fresh path information. Each mesh STA keeps its own HWMP sequence number that it increments, uses in HWMP elements, and processes according to the HWMP rules. HWMP sequence numbers for other mesh STAs are maintained in the forwarding information (see 11C.9.8.4).

An HWMP sequence number is included in the PREQ, PREP, PERR, and RANN elements. The HWMP sequence number in the forwarding information is updated whenever a mesh STA receives new (i.e., not stale) information about the HWMP sequence number from a PREQ, PREP, or PERR that may be received relative to that originator mesh STA, target mesh STA, or destination mesh STA.

HWMP depends on each mesh STA in the network to own and maintain its HWMP sequence number to guarantee the loop-freedom of all paths towards that mesh STA. A mesh STA increments its own HWMP sequence number in the following two circumstances:

- If it is an originator mesh STA, it shall increment its own HWMP sequence number immediately before it starts a path discovery. This prevents conflicts with previously established reverse paths towards the originator mesh STA. However, it might be advantageous not to increment the HWMP sequence number too frequently. An optional mechanism for achieving this is described in 11C.9.8.6.
- If it is a target mesh STA, it shall update its own HWMP sequence number to maximum (current HWMP sequence number, target HWMP sequence number in the PREQ) + 1 immediately before it generates a PREP in response to a PREQ. The target HWMP sequence number of the PREQ is relevant when a link was broken along the path and the stored sequence number was increased at an intermediate mesh STA.

HWMP sequence numbers are processed as follows:

a) HWMP sequence numbers are incremented monotonically as unsigned integers.

b) Comparing HWMP sequence numbers is done using a circular modulo 2^{32} comparison.

In general, when a mesh STA receives an element with an HWMP sequence number that is less than the HWMP sequence number in the corresponding forwarding information, it discards the received element. If they are the same, the outcome (element processed or not) depends on the type of the element and some additional conditions. These cases are noted in the applicable element descriptions.

The only circumstance in which a mesh STA may change the HWMP sequence number of another mesh STA in the forwarding information independently of the reception of an HWMP element originated by this mesh STA is in response to a broken or no longer usable link to the next hop towards that destination mesh STA. The mesh STA determines which destinations use a particular next hop by consulting its forwarding information. In this case, for each destination that uses the next hop, the mesh STA increments the HWMP sequence number in the forwarding information and marks the path as invalid (see also 11C.9.11). Whenever any forwarding information containing an HWMP sequence number greater than the recorded HWMP sequence number for an affected destination is received by a mesh STA that has marked that recorded forwarding information as invalid, the mesh STA shall update its forwarding information according to the information contained in the update.

11C.9.8.4 Forwarding information

In addition to the parameters contained in the basic forwarding information as described in 9.22.2, the forwarding information to a destination defined by HWMP also contains at least the destination HWMP sequence number (HWMP SN), the path metric, and the number of hops.

PREQ elements and PREP elements create or update the forwarding information of the mesh STAs that process these elements as follows:

- The mesh STA may create or update its forwarding information to the transmitter of the element if the path metric improves.
- The mesh STA shall create or update its forwarding information to the originator mesh STA, if it received a PREQ, and one of the following conditions is met:

  - The Originator HWMP sequence number > HWMP sequence number in the forwarding information for this originator mesh STA, or
  - The Originator HWMP sequence number = HWMP sequence number in the forwarding information for this originator mesh STA AND the updated path metric is better than the path metric in the forwarding information.
- The mesh STA shall create or update its forwarding information to the target mesh STA, if it received a PREP, and one of the following conditions is met:
  - The Target HWMP sequence number > HWMP sequence number in the forwarding information for this target mesh STA, or
  - The Target HWMP sequence number = HWMP sequence number in the forwarding information for this target mesh STA AND the updated path metric is better than the path metric in the forwarding information.

Table 11C-9 defines the values to be stored in the different fields of the forwarding information after a PREQ or PREP has been received.

Table 11C-9—Data for creation and update of forwarding information due to PREQ and PREP

	Received PREQ		Received PREP	
Field of forwarding information	**Forwarding information for transmitter of PREQ**	**Forwarding information for originator mesh STA**	**Forwarding information for transmitter of PREP**	**Forwarding information for target mesh STA**
HWMP sequence number	Invalid if created, no change if updated	PREQ field Originator HWMP Sequence Number	Invalid if created, no change if updated	PREP field Target HWMP Sequence Number
Next hop	Transmitter address of the management frame containing the PREQ element	Transmitter address of the management frame containing the PREQ element	Transmitter address of the management frame containing the PREP element	Transmitter address of the management frame containing the PREP element
Path metric	Accumulation of the initial value of the path metric with the metric of the link to the transmitter of the PREQ element	Accumulation of the value of PREQ field Metric with the metric of the link to the transmitter of the PREQ element	Accumulation of the initial value of the path metric with the metric of the link to the transmitter of the PREP element	Accumulation of the value of PREP field Metric with the metric of the link to the transmitter of the PREP element
Number of hops	1	Value of PREQ field Hop Count + 1	1	Value of PREP field Hop Count + 1
Precursor list	No change	No change except in case of an intermediate reply [see 11C.9.4.3 step f)]	No change	See 11C.9.10.4.3 step d)
Lifetime	The longer one of the lifetime of the stored forwarding information and the value of PREQ field Lifetime	The longer one of the lifetime of the stored forwarding information and the value of PREQ field Lifetime	The longer one of the lifetime of the stored forwarding information and the value of PREP field Lifetime	The longer one of the lifetime of the stored forwarding information and the value of PREP field Lifetime

Changes to the forwarding information in other situations, for instance, when processing a PERR element (see 11C.9.11), are described in the corresponding clauses.

11C.9.8.5 Repeated attempts at path discovery

Repeated attempts by a mesh STA at path discovery towards a single target shall be limited to dot11MeshHWMPmaxPREQretries. The minimum waiting time for the repeated attempt at path discovery to a single target is 2 × dot11MeshHWMPnetDiameterTraversalTime. For each attempt, the HWMP sequence number is incremented and a new Path Discovery ID is chosen.

11C.9.8.6 Limiting the rate of HWMP sequence number increments

In order to improve path stability (and further reduce overhead), a mesh STA may use the same originator HWMP sequence number for a certain time interval. In this case, the originator HWMP SN shall be incremented only after at least dot11MeshHWMPnetDiameterTraversalTime has elapsed since the previous increment. This mechanism prevents mesh STAs from changing the path frequently to the originator mesh STA every time the originator mesh STA sends a burst of PREQs within a very short time. This element of the protocol allows an originator mesh STA to immediately initiate on-demand path discovery to a new target without affecting recently refreshed paths to the originator in other mesh STAs.

11C.9.9 Path request (PREQ)

11C.9.9.1 General

This subclause describes the function, generation, and processing of the Path Request (PREQ) element.

11C.9.9.2 Function

The PREQ element, described in 7.3.2.113, is used for the following three purposes:

- Discovering a path to one or more targets
- Building a proactive (reverse) path selection tree to the root mesh STA
- Path maintenance (optional)

11C.9.9.3 Conditions for generating and sending a PREQ

A mesh STA shall send a PREQ element in an HWMP Mesh Path Selection frame, as defined in 7.4.15.3, in the following cases:

Case A: Path Discovery (Original Transmission)

All of the following applies:

- The mesh STA needs to establish an on-demand path to one or more targets for which there is no ongoing path discovery initiated by this mesh STA.
- The mesh STA has not sent a PREQ element for the target mesh STAs less than dot11MeshHWMPpreqMinInterval TUs ago. If this is the case, the transmission of the PREQ has to be postponed until this condition becomes true.
- The mesh STA has not made more than (dot11MeshHWMPmaxPREQretries – 1) repeated attempts at path discovery towards the target of the PREQ.

The content of a PREQ element in Case A shall be as shown in Table 11C-10.

Table 11C-10—Contents of a PREQ element in Case A

Field		Value
Element ID		Value given in Table 7-26 for the PREQ element
Length		26 + N × 11 (if Bit 6 (AE subfield) in the Flags field = 0) 32 + N × 11 (if Bit 6 (AE subfield) in the Flags field = 1)
Flags		Bit 0: 0 (gate announcement not applicable) Bit 1: 0 (group addressed) Bit 2: 0 (no proactive PREP applicable) Bit 3–5: Reserved Bit 6: (1 – if external address present, 0 – otherwise) Bit 7: Reserved
Hop Count		0
Element TTL		Maximum number of hops allowed for this element, e.g., dot11MeshHWMPnetDiameter.
Path Discovery ID		New unique Path Discovery ID, for instance, previous Path Discovery ID + 1
Originator Mesh STA Address		MAC address of the path originator
Originator HWMP Sequence Number		Previous Originator HWMP SN + 1. See 11C.9.8.6
Originator External Address		Present only if Bit 6 in Flags field = 1. This value is set to the external address, which is the source address of the MSDU (from outside the mesh BSS) that triggered the path discovery at the originator.
Lifetime		The time for which mesh STAs receiving the PREQ consider the forwarding information to be valid, e.g., dot11MeshHWMPactivePathTimeout.
Metric		Initial value of active path selection metric
Target Count		N (N ≥ 1)
Per Target	Per Target Flags	Bit 0 (TO): dot11MeshHWMPtargetOnly Bit 1: Reserved Bit 2 (USN): 0 if forwarding information for Target Address with valid HWMP sequence number exists, 1 otherwise Bit 3–7: Reserved
	Target Address	MAC address of requested target
	Target HWMP Sequence Number	If Per Target Flags Bit 2 (USN) is 0, the latest HWMP sequence number stored by the originator mesh STA for the target mesh STA from the forwarding information (see 11C.9.8.4). Otherwise, reserved.

Case B: Path Maintenance (Original Transmission) (optional)

All of the following applies:

- The mesh STA has a path to a given target mesh STA that is not a root mesh STA
- The last PREQ to this target was sent dot11MeshHWMPmaintenanceInterval TUs (or more) ago

The content of a PREQ in Case B shall be as shown in Table 11C-11.

Table 11C-11—Contents of a PREQ element in Case B

Field		Value
Element ID		Value given in Table 7-26 for the PREQ element
Length		26 + N × 11
Flags		Bit 0: 0 (gate announcement not applicable) Bit 1: 0 (group addressed) Bit 2: 0 (no proactive PREP applicable) Bit 3–5: Reserved Bit 6: 0 (no address extension) Bit 7: Reserved
Hop Count		0
Element TTL		Maximum number of hops allowed for this element, e.g., dot11MeshHWMPnetDiameter
Path Discovery ID		New unique Path Discovery ID, for instance, previous Path Discovery ID + 1
Originator Mesh STA Address		MAC address of the originator of the PREQ
Originator HWMP Sequence Number		Originator HWMP SN + 1. See 11C.9.8.6
Originator External Address		Field not present
Lifetime		The time for which mesh STAs receiving the PREQ consider the forwarding information to be valid, e.g., dot11MeshHWMPactivePathTimeout.
Metric		Initial value of active path selection metric
Target Count		N (N ≥ 1)
Per Target	Per Target Flags	Bit 0 (TO): 1 (target only) Bit 1: Reserved Bit 2 (USN): 0 Bit 3–7: Reserved
	Target Address	MAC Address of target mesh STA
	Target HWMP Sequence Number	The latest HWMP sequence number for this target known to the originator mesh STA.

Case C: Proactive PREQ (Original Transmission)

All of the following applies:

— The root mesh STA is configured as root mesh STA using proactive PREQs ([dot11MeshHWMProotMode = proactivePREQnoPREP (2)] OR [dot11MeshHWMProotMode = proactivePREQwithPREP (3)]).

— The root mesh STA sent its previous proactive PREQ dot11MeshHWMProotInterval TUs ago.

The contents of a PREQ in Case C shall be as shown in Table 11C-12.

Table 11C-12—Contents of a PREQ element in Case C

Field		Value
Element ID		Value given in Table 7-26 for the PREQ element
Length		37
Flags		Bit 0: 1 if dot11MeshGateAnnouncementProtocol is true (gate announcement), 0 otherwise Bit 1: 0 (group addressed) Bit 2: 0 if dot11MeshHWMProotMode = proactivePREQnoPREP(2), 1 if dot11MeshHWMProotMode = proactivePREQwithPREP(3) (proactive PREP) Bit 3–5: Reserved Bit 6: 0 (no address extension) Bit 7: Reserved
Hop Count		0
Element TTL		Maximum number of hops allowed for this element, e.g., dot11MeshHWMPnetDiameter.
Path Discovery ID		New unique Path Discovery ID, for instance, previous Path Discovery ID + 1
Originator Mesh STA Address		MAC address of the root mesh STA
Originator HWMP Sequence Number		Originator HWMP SN + 1. See 11C.9.8.6
Originator External Address		Field not present
Lifetime		dot11MeshHWMPactivePathToRootTimeout
Metric		Initial value of active path selection metric
Target Count		1
Per Target	Per Target Flags	Bit 0 (TO): 1 Bit 1: Reserved Bit 2 (USN): 1 Bit 3–7: Reserved
	Target Address	Broadcast address
	Target HWMP Sequence Number	0

Case D: Root Path Confirmation (Original Transmission)

One of the following applies:

— The mesh STA has received a RANN and the metric (RANN metric ⊕ metric to the transmitter of the RANN) is better than the metric to the root in the current forwarding information.

— The mesh STA has a path to a root mesh STA and the last PREQ to the root mesh STA was sent dot11MeshHWMPconfirmationInterval TUs (or more) ago.

The content of a PREQ element in Case D shall be as shown in Table 11C-13.

Table 11C-13—Contents of a PREQ element in Case D

Field		Value
Element ID		Value given in Table 7-26 for the PREQ element
Length		As required
Flags		Bit 0: 0 (gate announcement not applicable) Bit 1: 1 (individually addressed) Bit 2: 0 (no proactive PREP applicable) Bit 3–5: Reserved Bit 6: 0 (no address extension) Bit 7: Reserved
Hop Count		0
Element TTL		Maximum number of hops allowed for this element, e.g., dot11MeshHWMPnetDiameter
Path Discovery ID		Not used
Originator Mesh STA Address		MAC address of the originator mesh STA
Originator HWMP Sequence Number		Originator HWMP SN + 1. See 11C.9.8.6
Originator External Address		Field not present
Lifetime		The time for which mesh STAs receiving the PREQ consider the forwarding information to be valid, e.g., dot11MeshHWMPactivePathToRootTimeout.
Metric		Initial value of active path selection metric
Target Count		1
Per Target	Per Target Flags	Bit 0 (TO): 1 Bit 1: Reserved Bit 2 (USN): 0 Bit 3–7: Reserved
	Target Address	Root mesh STA MAC Address
	Target HWMP Sequence Number	The latest HWMP sequence number for this target known to the originator mesh STA

Case E: PREQ Propagation

Case E1 (target count = 1, no PREP generation as intermediate mesh STA):

All of the following applies:

- The mesh STA has received and accepted a PREQ—see 11C.9.9.4.2
- dot11MeshForwarding is true
- [The active forwarding information for the Originator Mesh STA was created or updated according to the rules defined in 11C.9.8.4] OR [{the Originator HWMP Sequence Number of the accepted PREQ = HWMP sequence number in the forwarding information for this

originator mesh STA} AND {the mesh STA has not previously received a PREQ with the same Originator Mesh STA Address and the same Path Discovery ID}]

- The Element TTL field is greater than 1—see 11C.9.8.2
- Target Count = 1
- [The mesh STA is not the target of the PREQ)]
  OR
  [the target of the PREQ is the MAC broadcast address (all ones)]
- the mesh STA is not the proxy of the target address
- [The TO (Target Only) subfield of the target in the PREQ is set (TO = 1)]
  OR
  [{the TO (Target Only) subfield of the target in the PREQ is not set (TO = 0)} AND {mesh STA has no active forwarding information for the requested target}]

The content of a PREQ element in Case E1 shall be as shown in Table 11C-14.

Table 11C-14—Contents of a PREQ element in Case E1

Field		Value
Element ID		Value given in Table 7-26 for the PREQ element
Length		As received
Flags		As received
Hop Count		As received + 1
Element TTL		As received – 1
Path Discovery ID		As received
Originator Mesh STA Address		As received
Originator HWMP Sequence Number		As received
Originator External Address		As received. This field is only present if Bit 6 of the Flags field (AE subfield) is 1.
Lifetime		As received
Metric		As received ⊕ own metric toward transmitter of received PREQ
Target Count		1
Per Target	Per Target Flags	As received
	Target MAC Address	As received
	Target HWMP Sequence Number	As received

Case E2 (target count = 1, PREP generation as intermediate mesh STA):

All of the following applies:

- The mesh STA has received and accepted a PREQ—see 11C.9.9.4.2
- dot11MeshForwarding is true

- [The active forwarding information for the Originator Mesh STA was created or updated according to the rules defined in 11C.9.8.4]
  OR
  [{the Originator HWMP Sequence Number of the accepted PREQ = HWMP sequence number in the forwarding information for this originator mesh STA} AND {the mesh STA has not previously received a PREQ with the same Originator Mesh STA Address and the same Path Discovery ID}]
- The Element TTL field is greater than 1—see 11C.9.8.2
- Target Count = 1
- The mesh STA is not the target of the PREQ
- The mesh STA is not the proxy of the target address
- The mesh STA has active forwarding information for the requested target
- [The TO (Target Only) subfield of the target in the PREQ is not set (TO = 0)]
  AND
  [the mesh STA has active forwarding information for the requested target]

The contents of a PREQ element in Case E2 shall be as shown in Table 11C-15.

Table 11C-15—Contents of a PREQ element in Case E2

Field		Value
Element ID		Value given in Table 7-26 for the PREQ element
Length		As received
Flags		As received
Hop Count		As received + 1
Element TTL		As received – 1
Path Discovery ID		As received
Originator Mesh STA Address		As received
Originator HWMP Sequence Number		As received
Originator External Address		As received. This field is only present if Bit 6 of the Flags field (AE subfield) is 1.
Lifetime		As received
Metric		As received ⊕ own metric toward transmitter of received PREQ
Target Count		1
Per Target	Per Target Flags	Bit 0 (TO): 1 (target only because mesh STA sent a PREP) Bit 1: Reserved Bit 2 (USN): As received Bit 3–7: Reserved
	Target MAC Address	As received
	Target HWMP Sequence Number	As received

Case E3 (target count > 1):

All of the following applies:

- The mesh STA has received and accepted a PREQ—see 11C.9.9.4.2
- dot11MeshForwarding is true
- [The active forwarding information for the Originator Mesh STA was created or updated according to the rules defined in 11C.9.8.4 (Forwarding information)]
  OR
  [{the Originator HWMP Sequence Number of the accepted PREQ = HWMP sequence number in the forwarding information for this originator mesh STA} AND {the mesh STA has not previously received a PREQ with the same Originator Mesh STA Address and the same Path Discovery ID}]
- The Element TTL field is greater than 1—see 11C.9.8.2
- Target Count > 1
- There is at least one requested target that is neither the recipient MAC address nor an external MAC address proxied by the recipient

The contents of a PREQ element in Case E3 shall be as shown in Table 11C-16.

Table 11C-16—Contents of a PREQ element in Case E3

Field	Value
Element ID	Value given in Table 7-26 for the PREQ element
Length	26 + N × 11
Flags	As received
Hop Count	As received + 1
Element TTL	As received – 1
Path Discovery ID	As received
Originator Mesh STA Address	As received
Originator HWMP Sequence Number	As received
Originator External Address	As received. This field is only present if Bit 6 of the Flags field (AE subfield) is set to 1.
Lifetime	As received
Metric	As received ⊕ own metric toward the transmitter of the received PREQ
Target Count	1 ≤ target count ≤ received target count received target count less the number of requested destinations, for which the processing mesh STA — is the target mesh STA or — is the target proxy mesh gate

Table 11C-16—Contents of a PREQ element in Case E3 *(continued)*

Field		Value
Per Target #A	Per Target Flags #A	As received
	Target MAC Address #A	As received
	Target HWMP Sequence Number #A	As received
Per Target #B	Per Target Flags #B	Bit 0 (TO): 1 (target only because mesh STA sent PREP) Bit 1: As received Bit 2 (USN): As received Bit 3–7: As received
	Target MAC Address #B	As received
	Target HWMP Sequence Number #B	As received

For the per target fields (Per Target Flags, Target Address, Target HWMP Sequence Number) assume the following:

— Target #A: If target A would have been the only requested target, it would generate a PREQ for propagation according to case E1.

— Target #B: If target B would have been the only requested target, it would generate a PREQ for propagation according to case E2.

11C.9.9.4 PREQ processing

11C.9.9.4.1 General

Received PREQ elements are subject to certain acceptance criteria. Processing and actions taken depend on the contents of the PREQ and the information available to the receiving mesh STA. See also 11C.9.8.

11C.9.9.4.2 Acceptance criteria

The PREQ element shall not be accepted (and shall not be processed as described in 11C.9.9.4.3) if any of the following is true:

— (The target address of the PREQ is neither the recipient MAC address, broadcast address, nor an external MAC address proxied by the recipient) AND (dot11MeshForwarding is false)

— (Bit 1 (Addressing Mode subfield) of the Flags field in the PREQ element is equal to 1) AND (there is no valid forwarding information with the destination mesh STA address equal to the Target Address of the PREQ element)

Otherwise, the PREQ element is accepted. See also 11C.9.8.

11C.9.9.4.3 Effect of receipt

A mesh STA receiving a PREQ according to the acceptance criteria in 11C.9.9.4.2 shall record the Path Discovery ID and the Originator Mesh STA Address. The receiving mesh STA shall create or update the active forwarding information it maintains for the originator mesh STA of the PREQ according to the rules defined in 11C.9.8.4.

If the active forwarding information for the Originator Mesh STA was created or updated according to the rules defined in 11C.9.8.4 or if the Target HWMP sequence number of the PREQ is the same as the HWMP sequence number in the forwarding information for the Target Mesh STA and there has been no record of the Originator Mesh STA and Path Discovery ID, the following applies:

a) If the mesh STA is the target of the PREQ or is the proxy of the target MAC address it shall initiate the transmission of a PREP to the originator mesh STA (11C.9.10.3 Case A). If the PREQ carries an external address (indicated by the AE subfield in the Flags field), the mesh STA shall update its proxy information with the Originator External Address as external address, the PREQ Originator Mesh STA Address as the corresponding proxy, the HWMP Sequence Number as proxy information sequence number, and for the proxy lifetime the longer one of the value of the PREQ Lifetime field and the proxy lifetime if the proxy information already exists (see also 11C.10.4.3).

b) If step a) was not applicable for the mesh STA and the AE subfield in the Flags field in the PREQ is 1, the mesh STA may update its proxy information with the Originator External Address as external address, the PREQ Originator Mesh STA Address as the corresponding proxy, the HWMP Sequence Number as proxy information sequence number, and for the proxy lifetime the longer one of the value of the PREQ Lifetime field and the proxy lifetime if the proxy information already exists (see also 11C.10.4.3).

c) If the mesh STA has valid forwarding information to any of the requested targets and the TO (Target Only) subfield for such a target is not set (TO = 0), it initiates the transmission of a PREP for each of these targets (see 11C.9.10.3 Case C).

d) If the mesh STA is initiating a PREP transmission on behalf of another target according to step c) (intermediate reply), it shall process all of the following:

   - Update the precursor list in its forwarding information for the target mesh STA with the next hop from the forwarding information of the originator mesh STA.
   - Update the lifetime for this precursor that is the longer one of the lifetime of the forwarding information of the target mesh STA.
   - Update the lifetime of the precursor list entry in case it already exists.
   - Update the precursor list in its forwarding information for the originator mesh STA with the next hop toward the target mesh STA.
   - Update the lifetime for this precursor that is the longer one of the lifetime of the forwarding information of the originator mesh STA.
   - Update the lifetime of the precursor list entry in case it already exists.

e) If the received PREQ is a proactive PREQ [target address is set to all ones, TO subfield is set (TO = 1)], the mesh STA generates a proactive PREP to the root mesh STA (see 11C.9.10.3 Case D) depending on the setting of the Proactive PREP subfield. If the Proactive PREP subfield is 1, a proactive PREP is generated, if it is 0, a proactive PREP is generated only if a bidirectional path to the root mesh STA is required (see Y.6).

f) If there are individually addressed targets in the PREQ that have not been processed in step a) or that have been processed in step c) or in step e), the receiving mesh STA shall propagate the PREQ as defined in 11C.9.9.3 Case E.

11C.9.10 Path reply (PREP)

11C.9.10.1 General

This subclause describes the function, generation, and processing of the Path Reply (PREP) element.

11C.9.10.2 Function

The PREP element is transmitted in individually addressed frames and is described in 7.3.2.114. The purpose of the PREP is as follows:

— To establish the forward path to a target mesh STA or target proxy mesh gate.

— To confirm the reverse path to the originator.

11C.9.10.3 Conditions for generating and sending a PREP

A mesh STA sends out a PREP element in an HWMP Mesh Path Selection frame, as defined in 7.4.15.3, in the following cases:

Case A: Path Discovery (Original Transmission)

A PREP is transmitted if the mesh STA has received and accepted a PREQ (see 11C.9.9.4.2) fulfilling any one of the following conditions:

— The Target Address of the PREQ is the same as MAC address of the receiving mesh STA

— The Target Address of the PREQ is an external address currently proxied by the mesh STA

The content of the generated PREP in Case A shall be as shown in Table 11C-17.

Table 11C-17—Contents of a PREP element in Case A

Field	Value
Element ID	Value given in Table 7-26 for the PREP element
Length	As required
Flags	Bit 0–5: Reserved Bit 6 (AE): (1 = external address present, 0 = otherwise) Bit 7: Reserved
Hop Count	0
Element TTL	Maximum number of hops allowed for this element
Target Mesh STA Address	MAC address of the target mesh STA or target proxy mesh gate
Target HWMP Sequence Number	HWMP sequence number of the target mesh STA or target proxy mesh gate after it has been updated according to 11C.9.8.3
Target External Address	External target address on behalf of which the PREP is sent. Present only if Bit 6 (AE subfield) in the Flags field is 1
Lifetime	As per the PREQ that triggered the transmission of this PREP
Metric	Initial value of active path selection metric
Originator Mesh STA Address	MAC address of the originator mesh STA
Originator HWMP Sequence Number	HWMP sequence number of the originator mesh STA

Case B: PREP Propagation

A PREP is propagated if all of the following applies:

- The mesh STA has received and accepted the PREP—see 11C.9.10.4.2
- The mesh STA is not the path originator

The contents of a PREP element in Case B shall be as shown in Table 11C-18.

Table 11C-18—Contents of a PREP element in Case B

Field	Value
Element ID	Value given in Table 7-26 for the PREP element
Length	As received
Flags	As received
Hop Count	As received + 1
Element TTL	As received – 1
Target Mesh STA Address	As received
Target HWMP Sequence Number	As received
Target External Address	As received
Lifetime	As received
Metric	As received ⊕ own metric toward the transmitting mesh STA
Originator Mesh STA Address	As received
Originator HWMP Sequence Number	As received

Case C: Intermediate reply (Original Transmission)

A PREP is transmitted if the mesh STA has received a PREQ fulfilling all of the following conditions:

- The TO (Target Only) subfield in the corresponding Per Target Flags field in the PREQ is not set (TO = 0)
- The receiving mesh STA has active forwarding information with
  a) A destination that is the same as the Target Address of the PREQ
  b) An HWMP sequence number that is greater than or equal to the Target HWMP sequence number of the PREQ
  c) A non-zero lifetime

The content of the generated PREP in Case C shall be as shown in Table 11C-19.

Table 11C-19—Contents of a PREP element in Case C

Field	Value
Element ID	Value given in Table 7-26 for the PREP element
Length	31

Table 11C-19—Contents of a PREP element in Case C *(continued)*

Field	Value
Flags	Bit 0–5: Reserved Bit 6 (AE): 0 Bit 7: Reserved
Hop Count	0
Element TTL	Maximum number of hops allowed for this element
Target Mesh STA Address	Target MAC address from the PREQ
Target HWMP Sequence Number	HWMP sequence number of the stored forwarding information of the Target of the PREQ
Target External Address	Not present
Lifetime	As per the PREQ that triggered the transmission of this PREP
Metric	Value of path metric taken from the active forwarding information for the target address of the PREQ
Originator Mesh STA Address	MAC address of the originator mesh STA
Originator HWMP Sequence Number	HWMP Sequence number of the originator mesh STA

Case D: Proactive PREP in Proactive PREQ mode (Original Transmission)

One of the following applies:

— The mesh STA has received a proactive PREQ with the Proactive PREP subfield set to 0 AND the mesh STA needs to establish or update a bidirectional path to the root mesh STA.

— The mesh STA has received a proactive PREQ with the Proactive PREP subfield set to 1.

Note, a proactive PREQ is a PREQ with a Target Address set to all ones and its TO subfield set (TO=1).

The content of the generated PREP in Case D shall be as shown in Table 11C-20.

Table 11C-20—Contents of a PREP element in Case D

Field	Value
Element ID	Value given in Table 7-26 for the PREP element
Length	31
Flags	Bit 0–5: Reserved Bit 6 (AE): 0 Bit 7: Reserved
Hop Count	0
Element TTL	Maximum number of hops allowed for this element
Target Mesh STA Address	MAC address of the mesh STA
Target HWMP Sequence Number	HWMP sequence number of the mesh STA

Table 11C-20—Contents of a PREP element in Case D *(continued)*

Field	Value
Target External Address	Not present
Lifetime	Lifetime of the PREQ that triggered the transmission of this PREP
Metric	Initial value of active path selection metric
Originator Mesh STA Address	MAC address of the root mesh STA (originator mesh STA of the PREQ)
Originator HWMP Sequence Number	HWMP sequence number of the root mesh STA (originator HWMP sequence number of the PREQ)

11C.9.10.4 PREP processing

11C.9.10.4.1 General

Received PREP elements are subject to certain acceptance criteria. Processing and actions taken depend on the contents of the PREP and the information available to the receiving mesh STA.

11C.9.10.4.2 Acceptance criteria

The PREP element shall not be accepted (and shall not be processed as described in 11C.9.10.4.3) if any of the following is true:

- (The Originator Mesh STA Address of the PREP is neither the recipient MAC address nor an external MAC address proxied by the recipient) AND (dot11MeshForwarding is false)

Otherwise, the PREP element shall be accepted.

11C.9.10.4.3 Effect of receipt

A mesh STA receiving a PREP according to the acceptance criteria in 11C.9.10.4.2 shall create or update the active forwarding information it maintains for the target mesh STA of the PREP (according to the rules defined in 11C.9.8.4). If the conditions for creating or updating the forwarding information have not been met in those rules, no further steps are applied to the PREP.

If the active forwarding information was created or updated according to the rules defined in 11C.9.8.4, the following applies:

a) If the receiving mesh STA is not the final destination of the PREP (originator mesh STA) and the field Element TTL > 1, the PREP is propagated as defined in 11C.9.10.3 Case B.

b) If the receiving mesh STA is the final destination of the PREP (originator mesh STA) and its AE subfield in the Flags field is 1, the mesh STA shall store the Target External Address, the Target Mesh STA Address, and the HWMP Sequence Number as proxy information sequence number in its proxy information. The proxy lifetime is the longer one of the value of the PREP Lifetime field and the proxy lifetime if the proxy information already exists (see also 11C.10.4.3).

c) If the receiving mesh STA is not the final destination of the PREP (originator mesh STA) and its AE subfield in the Flags field is 1, the mesh STA may store the Target External Address, the Target Mesh STA Address, and the HWMP Sequence Number as proxy information sequence number in its proxy information. The proxy lifetime is the longer one of the value of the PREP Lifetime field and the proxy lifetime if the proxy information already exists (see also 11C.10.4.3).

d) If the mesh STA propagates the PREP, the precursor list for the Target Mesh STA Address is updated by adding the next-hop mesh STA to which the PREP is propagated. In addition, at the mesh STA the precursor list for the originator mesh STA address is updated by adding the next-hop mesh STA towards the Target Address. The lifetimes of these entries in the precursor lists are the values of the lifetimes of the corresponding forwarding information.

11C.9.11 Path error (PERR)

11C.9.11.1 General

This subclause describes the function, generation, and processing of the PERR element.

11C.9.11.2 Function

The PERR element is used for announcing one or more unreachable destination(s). The announcement is sent to all traffic sources that have a known active path to the destination(s). The active forwarding information associated with the unreachable destination(s) should no longer be used for forwarding.

A PERR element may be either group addressed (if there are many precursors), individually addressed (if there is only one precursor), or individually addressed iteratively to all precursors (see 11C.9.7, item “PERR individually addressed”). The PERR element is processed as a single element when iteratively individually addressed to several precursors. The PERR element contains the destinations that are unreachable.

A PERR element is propagated by mesh STAs receiving a PERR if certain conditions are met.

A mesh STA generating or receiving a PERR may decide to establish paths to unreachable destinations using any of the available HWMP mechanisms.

11C.9.11.3 Conditions for generating and sending a PERR

A mesh STA shall send out a PERR element in an HWMP Mesh Path Selection frame, as defined in 7.4.15.3, in the following cases:

Case A: Original transmission (next hop is unusable)

The mesh STA has not sent a PERR element less than dot11MeshHWMPperrMinInterval TUs ago and the following applies:

— The mesh STA determines that the link to the next hop of an active path in its forwarding information is no longer usable.

NOTE—The detection might be triggered by the fact that a mesh STA is unable to forward an MSDU/MMPDU to a next-hop mesh STA.

The HWMP sequence number in the forwarding information of all unreachable destinations announced in this PERR is incremented by 1. The forwarding information for each unreachable destination announced in this PERR is invalidated.

The contents of a PERR element in Case A shall be as shown in Table 11C-21.

Table 11C-21—Contents of a PERR element in Case A

Field	Value
Element ID	Value given in Table 7-26 for the PERR element
Length	2 + N × 13
Element TTL	The maximum number of hops the element is propagated before being discarded.
Number of Destinations	Number of announced unreachable destinations in the PERR.
Flags #1	Bit 0–5: Reserved Bit 6 (AE): 0 Bit 7: Reserved
Destination Address #1	MAC address of unreachable destination #1.
HWMP Sequence Number #1	HWMP sequence number for Destination Address #1 from the forwarding information after above increment.
Reason Code #1	"MESH-PATH-ERROR-DESTINATION-UNREACHABLE" (see 7.3.1.7).
...	...

Case B: Original transmission (missing forwarding information)

The mesh STA has not sent a PERR element less than dot11MeshHWMPperrMinInterval TUs ago and one of the following applies:

— The mesh STA receives an individually addressed frame with a destination address not matching its own MAC address for which it has no forwarding information.

— The mesh STA receives an individually addressed frame with a destination address not matching its own MAC address and dot11MeshForwarding is false.

The contents of a PERR element in Case B shall be as shown in Table 11C-22.

Table 11C-22—Contents of a PERR element in Case B

Field	Value
Element ID	Value given in Table 7-26 for the PERR element.
Length	2 + N × 13
Element TTL	The maximum number of hops the element is propagated before being discarded.
Number of Destinations	Number of announced destinations with missing forwarding information in the PERR.
Flags #1	Bit 0–5: Reserved Bit 6 (AE): 0 Bit 7: Reserved

Table 11C-22—Contents of a PERR element in Case B *(continued)*

Field	Value
Destination Address #1	MAC address of destination with missing forwarding information #1. This is Address 3 of the received individually addressed frame.
HWMP Sequence Number #1	Reserved (0)
Reason Code #1	“MESH-PATH-ERROR-NO-FORWARDING-INFORMATION” (see 7.3.1.7).
...	...

Case C: Original transmission (proxy information is unusable)

The mesh STA has not sent a PERR element less than dot11MeshHWMPperrMinInterval TUs ago and the following applies:

— The mesh STA is a proxy mesh gate and determines that an active proxy information where the mesh STA is the proxy mesh gate is no longer usable.

The contents of a PERR element in Case C shall be as shown in Table 11C-23.

Table 11C-23—Contents of a PERR element in Case C

Field	Value
Element ID	Value given in Table 7-26 for the PERR element.
Length	2 + N × 19
Element TTL	The maximum number of hops the element is propagated before being discarded.
Number of Destinations	Number of announced unreachable external destinations in the PERR.
Flags #1	Bit 0–5: Reserved Bit 6 (AE): 1 Bit 7: Reserved
Destination Address #1	MAC address of proxy mesh gate #1 with unusable active proxy information.
HWMP Sequence Number #1	Last used HWMP sequence number for Destination Address #1.
Destination External Address #1	External MAC address of the active proxy information that is not longer usable and for which the mesh STA is the proxy mesh gate.
Reason Code #1	“MESH-PATH-ERROR-NO-PROXY-INFORMATION” (see 7.3.1.7.
...	...

Case D: PERR propagation

The mesh STA has not sent a PERR element less than dot11MeshHWMPperrMinInterval TUs ago and all of the following applies:

- The mesh STA received a PERR from a neighbor peer mesh STA.
- A destination in the PERR is the same as one of the destinations in the active forwarding information of the mesh STA where the next hop is the transmitter of the received PERR, and the forwarding information or the proxy information has been invalidated according to conditions in 11C.9.11.4.3 case b), case c), or case d).
- dot11MeshForwarding is true.
- The Element TTL field in the received PERR element is greater than 1.

The contents of a PERR element in Case D shall be as shown in Table 11C-24.

Table 11C-24—Contents of a PERR element in Case D

Field	Value
Element ID	Value given in Table 7-26 for the PERR element

Length	if AE subfield = 0: 2 + N × 13
	if AE subfield = 1: 2 + N × 19

Element TTL	Element TTL in received PERR element – 1
Number of Destinations	1 ≤ number of destinations in the PERR ≤ received value Received number of destinations less the number of received destinations for which the transmitter of the PERR is not the next hop
Flags #1	As received
Destination Address #1	MAC address of unreachable destination #1, as received
HWMP Sequence Number #1	If Reason Code #1 = “MESH-PATH-ERROR-NO-FORWARDING-INFORMATION” and received value = 0, then HWMP sequence number for Destination Address #1 from the forwarding information after the increment of 11C.9.10.4.3 step b). Otherwise, as received.
Destination External Address #1	As received This field is only present if Bit 6 (AE subfield) of the Flags field #1 is 1.
Reason Code #1	As received
...	...

11C.9.11.4 PERR processing

11C.9.11.4.1 General

Received PERR elements are subject to certain acceptance criteria. Processing and actions taken depend on the contents of the PERR and the information available to the receiving mesh STA. See also 11C.9.8.

11C.9.11.4.2 Acceptance criteria

The PERR shall be accepted (and shall be processed as described in 11C.9.11.4.3) if the following applies:

- The mesh STA that receives the PERR has forwarding information stored where
  - The destination is contained in the list of unreachable destinations of the PERR and
  - The next hop is the transmitter of the received PERR

Otherwise, the PERR element shall be discarded.

11C.9.11.4.3 Effect of receipt

The following applies only to a PERR element that was accepted according to the acceptance criteria in 11C.9.11.4.2:

a) The mesh STA creates a list of unreachable destinations consisting of those destinations from the received PERR for which the next hop in the local active forwarding information is the transmitter of the PERR. Step b) through step e) are applied to the destinations in this list.

b) If the Reason Code is "MESH-PATH-ERROR-NO-FORWARDING-INFORMATION" and the HWMP Sequence Number is 0, the receiving mesh STA increments the HWMP sequence number in the forwarding information of the listed unreachable destination by 1 and invalidates the forwarding information.

c) If the Reason Code is "MESH-PATH-ERROR-NO-FORWARDING-INFORMATION" and the HWMP Sequence Number is not 0 or the Reason Code is "MESH-PATH-ERROR-DESTINATION-UNREACHABLE" and the received HWMP sequence number for a listed unreachable destination is higher than the current HWMP sequence number in the forwarding information for that destination, the receiving mesh STA shall consider that destination unreachable and shall set the HWMP sequence number in the forwarding information to the HWMP sequence number received in the PERR and shall invalidate the forwarding information associated with this unreachable destination.

d) If the Reason Code is "MESH-PATH-ERROR-NO-PROXY-INFORMATION", the receiving mesh STA shall consider the corresponding Destination External Address unreachable and shall invalidate the proxy information associated with this unreachable external destination (proxy mesh gate is the Destination Address of the PERR, external MAC address is the Destination External Address of the PERR, proxy information sequence number is the HWMP Sequence Number).

e) A PERR element is propagated according to the conditions defined in 11C.9.11.3 Case D "PERR propagation."

11C.9.12 Root announcement (RANN)

11C.9.12.1 General

This subclause describes the function, generation, and processing of the Root Announcement (RANN) element.

11C.9.12.2 Function

The RANN element, described in 7.3.2.112, is used for announcing the presence of a mesh STA configured as root mesh STA using the proactive RANN mechanism. RANN elements are sent out periodically by the root mesh STA.

The RANN element propagates path metric information across the network so that each mesh STA can select a best metric path to the announced root mesh STA. This mechanism allows bidirectional trees to be built, using a robust procedure based on individually addressed frames initiated by the mesh STAs. This procedure makes the root mesh STA aware of all mesh STAs.

Receiving mesh STAs shall propagate the RANN as described in 11C.9.12.3 Case B.

11C.9.12.3 Conditions for generating and sending a RANN

A mesh STA sends out a RANN element in an HWMP Mesh Path Selection frame, as defined in 7.4.15.3, in the following cases:

Case A: Original transmission

All of the following conditions apply:

— The mesh STA is configured as a root mesh STA using the proactive RANN mechanism [dot11MeshHWMProotMode = rann (4)].

— The root mesh STA sent its previous RANN dot11MeshHWMPrannInterval TUs ago.

The contents of a RANN element in Case A shall be as shown in Table 11C-25.

Table 11C-25—Contents of a RANN element in Case A

Field	Value
Element ID	Value given in Table 7-26 for the RANN element
Length	21
Flags	Bit 0: set to 1 if dot11MeshGateAnnouncementProtocol is true, set to 0 otherwise. Bit 1–7: Reserved
Hop Count	0
Element TTL	Maximum number of hops allowed for this element
Root Mesh STA Address	MAC address of the root mesh STA
HWMP Sequence Number	Last used HWMP sequence number of the root mesh STA + 1
Interval	dot11MeshHWMPrannInterval
Metric	Initial value of active path selection metric

Case B: Propagation

All of the following conditions apply:

— The mesh STA has valid forwarding information to a root mesh STA using the proactive RANN mechanism [dot11MeshHWMProotMode = rann (4)]

— The mesh STA sent its previous RANN dot11MeshHWMPrannInterval TUs ago

— dot11MeshForwarding is true

— The Element TTL field is greater than 1—see 11C.9.8.2

The contents of a RANN element in Case B shall be as shown in Table 11C-26.

Table 11C-26—Contents of a RANN element in Case B

Field	Value
Element ID	Value given in Table 7-26 for the RANN element
Length	As received
Flags	As received
Hop Count	As received + 1
Element TTL	As received – 1
Root Mesh STA Address	As received
HWMP Sequence Number	As received
Interval	As received
Metric	As received ⊕ own link metric toward the transmitting mesh STA

11C.9.12.4 RANN reception

11C.9.12.4.1 General

Received RANN elements are subject to certain acceptance criteria. Processing and actions taken depend on the content of the RANN and the forwarding information maintained by the receiving mesh STA. See also 11C.9.8.

11C.9.12.4.2 Acceptance criteria

The RANN element shall not be accepted (and shall not be processed as described in 11C.9.12.4.3) if any of the following is true:

- The HWMP Sequence Number < previous HWMP sequence number from this originating root mesh STA
- (The HWMP Sequence Number = previous HWMP sequence number) AND (updated path metric is *worse than* previous path metric)

Otherwise, the RANN element shall be accepted.

11C.9.12.4.3 Effect of receipt

The following applies only to a RANN element that was accepted according to the acceptance criteria in 11C.9.12.4.2.

a) The receiving mesh STA shall set dot11MeshHWMPrannInterval to the value of the Interval field of the received RANN.

b) The receiving mesh STA may initiate a PREQ/PREP exchange with the root mesh STA to set up or update a path to the root mesh STA. See 11C.9.9.3 Case D.

c) The receiving mesh STA may record the Root Mesh STA Address, together with the HWMP Sequence Number, Hop Count, and Metric in order to assist in executing 11C.9.9.3 Case D.

The receiving mesh STA shall transmit a RANN if the conditions defined in 11C.9.12.3 Case B are true.

11C.9.13 Considerations for support of STAs without mesh functionality

The verification, by the mesh STA collocated with the AP, of disjunct MAC addresses between a non-AP STA without mesh functionality and mesh STAs during authentication/association of the non-AP STA without mesh functionality (see 11.3.3) may be done by issuing a PREQ for the MAC address of the non-AP STA without mesh functionality by the mesh STA collocated with the AP. The TO (Target Only) subfield of the Per Target Flags field of the PREQ shall be set to 1.

The MAC address of the non-AP STA already exists in the MBSS if the AP with mesh functionality receives a PREP for the MAC address of the non-AP STA and it can be derived from the PREP that the requested MAC address is originated from a mesh STA. (The AE subfield of the Flags field of the PREP is set to 0, see 7.3.2.114.)

11C.10 Interworking with the DS

11C.10.1 Overview of interworking between a mesh BSS and a DS

A mesh STA that has access to a DS is called a mesh gate. Mesh STAs in an MBSS access the DS via the mesh gate. An MBSS functions like an IEEE 802 LAN segment that is compatible with IEEE 802.1D. The MBSS appears as a single access domain.

An MBSS may contain two or more mesh gates. When multiple mesh gates in an MBSS have access to the same DS, the MBSS has more than one “port” (in the sense of IEEE Std 802.1D-2004, for example) through which it accesses the DS. Accordingly, broadcast loops may occur. Therefore, mesh gates should implement a loop preventing protocol in the DS.

NOTE 1—In the DS a typical implementation uses the Rapid Spanning Tree Protocol (RSTP) as specified in IEEE Std 802.1D-2004. With RSTP the resulting active DS topology forms a tree. Then, even if multiple mesh gates connect with the same DS, the MBSS only accesses the DS through a single mesh gate.

When dot11MeshGateAnnouncementProtocol is true, the mesh gate announces its presence to other mesh STAs in the MBSS. The mesh gate uses the gate announcement protocol (see 11C.10.2) or alternatively one of the HWMP proactive path selection methods with the Gate Announcement field equal to 1:

- The proactive PREQ mechanism (see 11C.9.4.2), with the Gate Announcement field equal to 1 (see 11C.9.9.3)
- The proactive RANN mechanism (see 11C.9.4.3), with the Gate Announcement field equal to 1 (see 11C.9.12.3)

When the mesh gate uses one of the HWMP proactive path selection methods, the gate announcement protocol is not used.

A mesh STA discovers the presence of a mesh gate with access to the external network by receiving Gate Announcement elements (or PREQ and RANN with the Gate Announcement field equal to 1 if using such mechanisms). Mesh STAs propagate these elements to neighbor mesh STAs in order to propagate the information throughout the MBSS.

NOTE 2—The decision to set dot11MeshGateAnnouncementProtocol to true is beyond the scope of the standard. In general, the mesh gate announces that it has access to a broader network beyond the MBSS, using gate announcement protocol or HWMP proactive path selection methods with the Gate Announcement field equal to 1. One example of this configuration is that the mesh gate has access to a portal through the DS.

When a mesh gate has access to IEEE 802 STAs outside the mesh BSS (a mesh STA collocated with an AP, another mesh STA that belongs to another MBSS, etc.), the mesh gate acts as an intermediary for the IEEE 802 STAs outside the MBSS so that the forwarding information inside the MBSS only contains addresses that belong to the MBSS. The mesh gate acting as an intermediary for external STAs is termed proxy mesh gate. When the end station of an IEEE 802 communication is an external STA, mesh STAs handle addresses of the end-to-end IEEE 802 communication as depicted in Figure 9-38. Proxy mesh gate operation is described in 11C.10.4.

11C.10.2 Gate announcement protocol

11C.10.2.1 General

This subclause describes the function, generation, and processing of the Gate Announcement (GANN) element.

11C.10.2.2 Function

The Gate Announcement (GANN) element, described in 7.3.2.111, is used to announce the presence of a mesh gate with dot11MeshGateAnnouncementProtocol equal to true in the mesh BSS. Gate announcements allow mesh STAs to discover such a mesh gate and, if necessary, to build a path towards it.

11C.10.2.3 Conditions for generating and sending a GANN

A mesh STA shall send a GANN element in a Gate Announcement frame, as defined in 7.4.15.4, in the following cases:

Case A: Original transmission

The mesh STA is a mesh gate not sending PREQ or RANN with the Gate Announcement field equal to 1 and dot11MeshGateAnnouncementProtocol is true. The mesh STA shall transmit the Gate Announcement frame at every dot11MeshGateAnnouncementInterval.

The content of a GANN element in Case A shall be as shown in Table 11C-27.

Table 11C-27—Contents of a GANN element in Case A

Field	Value
Element ID	Value given in Table 7-26 for the GANN element
Length	15
Flags	Reserved
Hop Count	0
Element TTL	Maximum number of hops allowed for the gate announcement
Mesh Gate Address	Mesh STA MAC address
GANN Sequence Number	Previous GANN sequence number + 1
Interval	dot11MeshGateAnnouncementInterval

The mesh gate shall assign the GANN Sequence Number from a single modulo-2^{32} counter, starting at 0 and incrementing by 1 for each GANN element transmission.

Case B: Propagation

All of the following conditions are met:

- The mesh STA has received and accepted a gate announcement.
- The decremented Element TTL of the gate announcement is equal to or greater than 1.
- dot11MeshForwarding is true.

The content of a GANN element in Case B shall be as shown in Table 11C-28.

Table 11C-28—Contents of a GANN element in Case B

Field	Value
Element ID	Value given in Table 7-26 for the GANN element
Length	15
Flags	As received
Hop Count	As received + 1
Element TTL	As received – 1
Mesh Gate Address	As received
GANN Sequence Number	As received
Interval	As received

11C.10.2.4 GANN processing

11C.10.2.4.1 General

A received gate announcement is subject to certain acceptance criteria. Processing depends on the contents of the gate announcement and the information available at the receiving mesh STA.

11C.10.2.4.2 Acceptance criteria

The Gate Announcement element shall not be accepted (and shall not be processed as described in 11C.10.2.4.3) if the GANN Sequence Number of the gate announcement is equal or lower than the GANN Sequence Number of the most recently accepted gate announcement with the same Mesh Gate Address.

11C.10.2.4.3 Effect of receipt

The following applies only to a Gate Announcement element that was accepted according to the acceptance criteria in 11C.10.2.4.2. The receiving mesh STA shall transmit a gate announcement as described in 11C.10.2.3, Case B.

The Mesh Gate Address field of the GANN contains the address of the mesh gate, and may be stored for the purpose of determining paths to the mesh gates. Paths to mesh gates allow mesh STAs to forward MSDUs to addresses for which no path could be determined (see 9.22.9).

11C.10.3 Data forwarding at proxy mesh gates

11C.10.3.1 General

Forwarding of MSDUs from the DS into the MBSS by a proxy mesh gate follows the procedures given in 9.22.3.

Forwarding of MSDUs from the MBSS into the DS by a proxy mesh gate follows the procedures that apply for the specific collocated network.

A proxy mesh gate learns the addresses of the other proxy mesh gates in the MBSS and of external addresses proxied by them through the receipt of path selection messages and messages carrying proxy information (for example, see 7.3.2.116).

11C.10.3.2 Forwarding of MSDUs from the MBSS to the DS

On receipt of an individually addressed Mesh Data frame from the MBSS with Address Extension Mode equal to 10 (binary), a proxy mesh gate shall perform the following:

- If Address 5 is a known destination MAC address in the proxy information (external address) and proxied by the proxy mesh gate, the proxy mesh gate forwards the MSDU to the external address through the DS.
- If Address 5 is a known destination MAC address in the proxy information (external address) and proxied by a different proxy mesh gate, the MSDU is forwarded through the MBSS to the proxy mesh gate that proxies the external address. The MSDU is sent into the MBSS according to the procedures in 9.22.4.1 as an individually addressed Mesh Data frame with Address 3 set to the MAC address of the proxy mesh gate of the proxy information proxying Address 5, Address 4 set to the MAC address of this proxy mesh gate, and Address 5 and Address 6 kept unchanged.
- If Address 5 is unknown to the proxy mesh gate, the mesh gate forwards the MSDU to the DS. The mesh gate may send an error notification to the mesh source of the MSDU. In HWMP, this is done by sending a PERR as described in 11C.9.11.3 Case C.

On receipt of group addressed Mesh Data frame from the MBSS with Address Extension Mode equal to 01 (binary), a proxy mesh gate shall forward the MSDU to the DS using a group addressed frame.

11C.10.3.3 Forwarding of MSDUs from the DS to the MBSS

On receipt of an individually addressed MSDU from the DS, a proxy mesh gate shall perform the following depending on the possible destination:

a) If the destination of the MSDU is a mesh STA address that the mesh gate knows to be inside the MBSS, the mesh gate forwards the MSDU according to the procedures for frame addressing and data forwarding at source mesh STAs in an MBSS (9.22.4.1). The MSDU shall be transmitted using a frame with the four-address MAC header format [with the Address Extension Mode subfield in the Mesh Control field set to 10 (binary)], where the Mesh Address Extension subfield in the Mesh Control field carries the addresses of the end stations, as specified in row “Mesh Data (proxied, individually addressed)” of Table 9-13. The address fields are set as follows:
    - Address 1: The address of the next-hop mesh STA (toward the destination mesh STA according to the forwarding information—see 9.22.2)
    - Address 2: The address of the proxy mesh gate
    - Address 3: The address of the destination mesh STA
    - Address 4: The address of the proxy mesh gate
    - Address 5: The address of the destination end mesh STA that is the same as Address 3

— Address 6: The address of the source end mesh STA that is the source address of the MSDU received from the DS

b) If the destination of the MSDU is an external address that is proxied by another proxy mesh gate in the MBSS, the mesh gate forwards the MSDU according to the procedures for frame addressing and data forwarding at source mesh STAs in an MBSS (9.22.4.1). The MSDU shall be transmitted using a frame with the four-address MAC header format [with the Address Extension Mode subfield in the Mesh Control field set to 10 (binary)], where the Mesh Address Extension subfield in the Mesh Control field carries the addresses of the end stations, as specified in row "Mesh Data (proxied, individually addressed)" of Table 9-13. The address fields are set as follows:

— Address 1: The address of the next-hop mesh STA [toward the proxy mesh gate of the destination of the MSDU as derived from the proxy information (see 11C.10.4.2) and according to the forwarding information—9.22.2]
— Address 2: The address of this proxy mesh gate
— Address 3: The address of the proxy mesh gate of the destination of the MSDU as derived from the proxy information (see 11C.10.4.2)
— Address 4: The address of this proxy mesh gate
— Address 5: The address of the destination end mesh STA that is the destination address of the MSDU received from the DS
— Address 6: The address of the source end mesh STA that is the source address of the MSDU received from the DS

c) If the MSDU has a destination address that is unknown to the mesh gate, the mesh gate forwards the MSDU to other known mesh gates in the MBSS as an individually addressed frame according to the procedures for frame addressing and data forwarding of individually addressed frames at source mesh STAs in an MBSS (9.22.4.1). The MSDU shall be transmitted using a frame with the four-address MAC header format [with the Address Extension Mode subfield in the Mesh Control field set to 10 (binary)], where the Mesh Address Extension subfield in the Mesh Control field carries the addresses of the end stations, as specified in row "Mesh Data (proxied, individually addressed)" of Table 9-13. The address fields are set as follows:

— Address 1: The address of the next-hop mesh STA (toward the other known mesh gate in the MBSS according to the forwarding information—see 9.22.2)
— Address 2: The address of this proxy mesh gate
— Address 3: The address of the other known mesh gate in the MBSS
— Address 4: The address of this proxy mesh gate
— Address 5: The address of the destination end mesh STA that is the unknown destination address of the MSDU received from the DS
— Address 6: The address of the source end mesh STA that is the source address of the MSDU received from the DS

Note that the procedure to determine that an address is unknown depends on the active path selection protocol. It may require an attempt to establish a path to the destination (see 11C.7).

On receipt of a group addressed MSDU from the DS, the mesh gate forwards the MSDU according to the procedures for frame addressing and data forwarding of group addressed frames at source mesh STAs in an MBSS (9.22.5.1). The MSDU shall be transmitted using a frame with the three-address MAC header format [with the Address Extension Mode subfield in the Mesh Control field set to 01 (binary)], where the Mesh Address Extension subfield in the Mesh Control field carries the address of the source end stations, as specified in row "Mesh Data (proxied, group addressed)" of Table 9-13. The address fields are set as follows:

— Address 1: The group address
— Address 2: The address of the proxy mesh gate
— Address 3: The address of the proxy mesh gate
— Address 4: The address of the source external STA

11C.10.4 Proxy information and proxy update

11C.10.4.1 General

Forwarding information of mesh STAs only contains addresses of mesh STAs that belong to the MBSS. However, the end station of the IEEE 802 communication may be an IEEE 802 station outside the MBSS, and such station is called external STA. Examples of external STAs are as follows:

- STAs that are associated with an AP that is collocated with a mesh STA
- STAs that are behind a mesh gate

Mesh STAs forward MSDUs to external STAs by treating MAC addresses of the external STAs as external addresses. The mesh STAs that are the destination mesh STAs of the messages destined to external STAs are called proxy mesh gates, and their MAC addresses are called proxy addresses.

NOTE—External STAs are reached using mesh services solely, i.e., they are not part of an MBSS. The mechanism by which the proxy mesh gate bridges the MBSS and the external STAs are beyond the scope of the standard. However, the standard describes the method by which mesh STAs use the external addresses that are discovered and bridged by the proxy mesh gate.

11C.10.4.2 Proxy information

Proxy mesh gates and source mesh STAs of MSDUs destined to external STAs maintain proxy information. Proxy information contains the external address, the corresponding proxy address, the sequence number of the proxy information, and the corresponding proxy information lifetime.

Mesh STAs can learn the addresses of proxy mesh gates and of the external stations proxied by these proxy mesh gates through the receipt of proxy update messages or path selection messages carrying proxy information. Particularly, proxy information is updated in the following circumstances:

- A mesh STA receives and processes a proxy update (see 11C.10.4.3)
- A mesh STA receives and processes an element of the active path selection protocol containing proxy information. In HWMP, these are PREQ elements (see 11C.9.9.4.3), PREP elements (see 11C.9.10.4.3), and PERR elements (see 11C.9.11.4.3)

Additionally, proxy mesh gates may also proactively maintain proxy information on external STAs.

When the proxy information lifetime is specified, a mesh STA shall maintain the proxy information as valid information until the lifetime expires. Details of the lifetime are described in 11C.10.4.3.4.

The sequence number of the proxy information and the proxy mesh gate address define a chronological order of the proxy information of an external STA at a specific proxy mesh gate.

When the proxy information is created at the proxy mesh gate, the proxy sequence number is initialized to an arbitrary value. The proxy information sequence number in the proxy information at the proxy mesh gate is incremented by 1 before the transmission of the proxy information to another mesh station. The proxy information sequence number shall be incremented if the proxy information is invalidated.

If proxy information is transmitted in HWMP elements (PREQ, PREP, and PERR), the proxy information sequence number is set to the HWMP sequence number of the HWMP element containing this proxy information.

Comparison of the proxy information sequence numbers is performed using a circular modulo 2^{32} comparison.

Valid proxy information is used to determine and set Address 5 and Address 6 in individually addressed Mesh Data frames, or Address 4 in group addressed Mesh Data frames.

11C.10.4.3 Proxy update (PXU)

11C.10.4.3.1 General

This clause describes the function, generation, and processing of the Proxy Update (PXU) element.

11C.10.4.3.2 Function

A mesh STA generates a PXU element to inform a destination mesh STA about proxy information of external addresses that are reachable through the proxy mesh gate specified in the PXU element.

NOTE—Typically, a proxy mesh gate generates and sends a PXU element to another proxy mesh gate in the MBSS or a mesh STA that originates traffic to the external stations proxied by the proxy mesh gate. However, the standard also allows other usage of the PXU element.

The PXU element is transmitted in a Proxy Update frame (an individually addressed frame). The Proxy Update frame may contain multiple PXU elements when needed (for instance, the proxy mesh gate has a large number of proxy information).

11C.10.4.3.3 Conditions for generating and sending a PXU

A proxy mesh gate may transmit a PXU when it adds, updates, or deletes an external address to (or from) its proxy information. A proxy mesh gate may also transmit a PXU at periodic intervals.

A mesh STA that holds proxy information of a proxy mesh gate in the MBSS may also transmit a PXU.

A mesh STA may retransmit the same PXU element repeatedly until the mesh STA receives a PXUC element from the destination mesh STA. See 11C.10.4.4.

The content of a PXU element shall be as shown in Table 11C-29.

The proxy mesh gate shall assign the PXU ID from a single modulo-256 counter, starting at 0 and incrementing by 1 for each PXU element.

11C.10.4.3.4 Effect of receipt of a PXU

A mesh STA that receives the PXU element shall update its proxy information with the list of proxy information reported in the PXU under the following conditions:

- Proxy information for the external MAC address and the proxy mesh gate reported in the Proxy Information field of the PXU does not exist at the mesh STA.
- Proxy information for the external MAC address and the proxy mesh gate reported in the Proxy Information field of the PXU does exist at the mesh STA and the value of the Proxy Information Sequence Number subfield in the received PXU is larger than the value of the proxy information sequence number in the proxy information at the mesh STA.

When multiple PXU elements are contained in the received Proxy Update frame, the recipient mesh STA shall process all of the PXU elements in the frame.

The MAC address of the proxy mesh gate is taken from the Proxy MAC Address subfield in the Proxy Information field when bit 1 in the Flags subfield is equal to 0, and from the PXU Originator MAC Address field in the PXU element when bit 1 in the Flags subfield is equal to 1.

Table 11C-29—Contents of a PXU element

Field		Value/description
Element ID		Value given in Table 7-26 for the PXU element
Length		8 + length of N Proxy Information fields
PXU ID		Previous PXU ID + 1
PXU Originator MAC Address		MAC address of the originator of the PXU
Number of Proxy Information (N)		Number of proxy information reported to the destination mesh STA (N ≥ 1).
Per Proxy Information	Flags	Bit 0: 0: add proxy information; 1: delete proxy information Bit 1: 0: Proxy MAC Address field present; 1: Proxy MAC Address = PXU Originator MAC Address, Proxy MAC Address field not present Bit 2: 0: Proxy Information Lifetime field not present; 1: Proxy Information Lifetime field present. If Bit 0 is 1, Bit 1 shall be set to 0. Bit 3–7: Reserved
	External MAC Address	MAC address of the STA proxied by the proxy mesh gate.
	Proxy Information Sequence Number	Proxy information sequence number of the proxy information after being incremented. See 11C.10.4.2.
	Proxy MAC Address	MAC address of the proxy mesh gate. This field is only present if Bit 1 of the Flags field is 0.
	Proxy Information Lifetime	The proxy information lifetime of this proxy information as taken from the proxy information of the originator of the PXU.

The MAC address of the external STA is taken from the External MAC Address subfield of the corresponding Proxy Information field in the received PXU element.

The sequence number of the proxy information is taken from the Proxy Information Sequence Number subfield of the corresponding Proxy Information field in the received PXU element.

If the Proxy Information Lifetime subfield is present (the Lifetime subfield in the Flags subfield is 1) and there is already proxy information stored for the proxy mesh gate and external address reported in the proxy information of the PXU element, the mesh STA shall set the proxy lifetime to the larger one of the proxy lifetime reported by the PXU and the stored proxy information.

If the Proxy Information Lifetime subfield is present (bit 2 of the Flags subfield is 1) and there is proxy information stored for the proxy mesh gate and external address reported in the proxy information of the PXU element, the mesh STA shall set the proxy information lifetime to the value in the Proxy Information Lifetime subfield.

If the Proxy Information Lifetime subfield is not present, the lifetime of the proxy information is the same as the lifetime of the path to the proxy address. Alternatively, the lifetime of the proxy information may be set to a value representing infinity.

The destination mesh STA that received the PXU shall send a Proxy Update Confirmation (PXUC) to the originator mesh STA of the PXU as described in 11C.10.4.4.3.

11C.10.4.4 Proxy Update Confirmation (PXUC)

11C.10.4.4.1 General

This clause describes the function, generation, and processing of the Proxy Update Confirmation (PXUC) element.

11C.10.4.4.2 Function

A PXUC element is generated by the destination mesh STA of a PXU to inform the sender of the PXU that the PXU has been properly received.

The PXUC element is transmitted in a Proxy Update Confirmation frame (an individually addressed frame). The Proxy Update Confirmation frame may contain multiple PXUC elements in order to confirm the reception of multiple PXU elements to the destination of the Proxy Update Confirmation frame.

11C.10.4.4.3 Conditions for generating and sending a PXUC

The destination mesh STA of a Proxy Update frame containing a PXU element shall send a PXUC element to the originator mesh STA of the PXU element.

The content of a PXUC element shall be as shown in Table 11C-30.

Table 11C-30—Contents of a PXUC element

Field	Value/description
Element ID	Value given in Table 7-26 for the PXUC element
Length	7
PXU ID	PXU ID of the PXU that is being confirmed
Destination mesh STA Address	MAC address of the originator of the PXUC

11C.10.4.4.4 Effect of receipt of PXUC

If a mesh STA receives a PXUC element in a PXUC frame in response to a PXU element it originated, the mesh STA shall no longer send any PXUs with the same PXU ID as given in the received PXUC element.

11C.10.5 Mesh STA collocation

A mesh STA collocated with another STA shall use a MAC address that is different from the one used by the collocated STA. This precludes ambiguities relating to the presence of the Mesh Control field in the Frame Body (see 7.1.3.6), GTK use (see 8.4.1.1.3b), and proxy information (see 11C.10.4.2).

Path selection with collocated mesh STAs using HWMP is described in 11C.9.5.

11C.11 Intra-mesh congestion control

11C.11.1 General

Intra-mesh congestion control is based on the following three main mechanisms:

a) Local congestion monitoring and congestion detection

b) Congestion control signaling

c) Local rate control

A mesh STA shall activate a congestion control protocol specified by dot11MeshActiveCongestionControlMode. At any given time, there is only one congestion control protocol active in a particular MBSS, signalled in the Congestion Control Mode Identifier field of the Mesh Configuration element. This standard specifies the congestion control signaling protocol that shall be available in any MBSS with an activated congestion control.

NOTE—This standard allows for inclusion of more advanced or alternative congestion control schemes through the Congestion Control Mode Identifier in the Mesh Configuration element.

11C.11.2 Congestion control signaling protocol

When dot11MeshActiveCongestionControlMode is congestionControlSignaling (1), the mesh STA activates the congestion control signaling protocol. The congestion control signaling protocol specifies the signaling messages used with intra-mesh congestion control. Specific algorithms for local congestion monitoring and congestion detection are beyond the scope of the congestion control signaling protocol.

The congestion control signaling protocol is triggered after congestion is detected at a mesh STA. A mesh STA that detects congestion, and the traffic destination causing this congestion, may transmit a Congestion Control Notification frame to the mesh STAs of the traffic source and other neighboring mesh STAs. The frame contains one or more Congestion Notification elements, each of which specifies the traffic destination causing the congestion and the expected duration of the congestion per AC per mesh destination as estimated by the congested mesh STA.

Upon receipt of a Congestion Notification frame a mesh STA may stop forwarding, or reduce the rate of forwarding, traffic to the destinations listed in the Congestion Notification elements via the mesh STA reporting congestion for the duration specified in the Congestion Notification element. It may also send its own Congestion Control Notification frame to mesh STAs that are the source of the reported congestion, and other neighboring mesh STAs. Any time difference between receipt of the original Congestion Control Notification frame and the transmission of this new Congestion Control Notification frame should be reflected in the duration indicated in the new congestion control notification in such a way that any timers set by mesh STAs in response to the first report of congestion for a given destination all expire at the same time.

If the Destination MAC Address field in a received Congestion Notification element is the group address, it should be interpreted to mean that communication with the transmitter of this frame should be stopped, or reduced, for the duration specified in the Congestion Notification element. This event should not result in the transmission of a Congestion Notification element with a Destination MAC Address field set to the group address to any neighbor mesh STAs.

When the duration of a traffic congestion report has expired, a mesh STA should resume forwarding traffic to the destinations that were listed in the traffic congestion report via the mesh STA that reported congestion.

NOTE 1—Local policies/mechanisms implemented in a mesh STA might be required to ensure timely transmission of the congestion control signaling messages and to avoid transmission of stale messages that might reduce network efficiency.

NOTE 2—A mesh STA that receives a Congestion Control Notification frame might choose to adjust its frame rate, defined by the number of transmitted frames per a unit of time, to the sender of the Congestion Control Notification frame in the identified congested AC(s) for the duration specified in the Congestion Notification element. The reduction of the frame rate to a congested mesh STA avoids waste of the mesh resources for transmission of packets that with high probability will not be handled/forwarded by the congested mesh STA.

11C.12 Synchronization and beaconing in MBSSs

11C.12.1 TSF for MBSSs

A mesh STA shall initialize and update its TSF timer depending on its active synchronization method. Each mesh STA shall maintain a TSF timer as described in 11.1.2, and conform to the TSF timer accuracy as described in 11.1.2.4.

11C.12.2 Extensible synchronization framework

11C.12.2.1 General

This standard introduces an extensible framework to enable the implementation of multiple synchronization methods for mesh STAs. Within the extensible synchronization framework, the neighbor offset synchronization method is defined as the default mandatory synchronization method in order to enable minimal synchronization capabilities and interoperability between mesh STAs that use MCCA, MBCA, or operate in light or deep sleep mode. The framework allows to integrate other synchronization methods for MBSSs. A vendor may implement any synchronization method using this framework to meet special application needs. Although a mesh STA may include multiple implementations of synchronisation methods, only one synchronization method shall be used by a mesh STA at a time and all the mesh STAs in an MBSS use the same synchronisation method. The active synchronization method is controlled by the SME and given to the MLME by dot11MeshActiveSynchronizationMethod.

Mesh STAs shall announce the active synchronization method using the Synchronization Method Identifier field in the Mesh Configuration element in their Beacon and Probe Response frames.

11C.12.2.2 Neighbor offset synchronization method

11C.12.2.2.1 General

When dot11MeshActiveSynchronizationMethod is neighborOffsetSynchronization (1), the mesh STA shall use the neighbor offset synchronization method as its active synchronization method, and maintain the timing offset value between its own TSF timer and the TSF timer of each neighbor STA with which it synchronizes. The mesh STA shall set the Synchronization Method Identifier field in the Mesh Configuration element to 1.

The mesh STA shall maintain synchronization with all of its neighbor peer mesh STAs. The mesh STA should maintain synchronization with up to dot11MeshNbrOffsetMaxNeighbor neighbor mesh STAs that are in the same MBSS. Additionally, the mesh STA should maintain synchronization with up to dot11MeshNbrOffsetMaxNeighbor neighbor STAs that are outside of the MBSS.

Upon receipt of an MLME-MESHNEIGHBOROFFSETSYNCSTART.request primitive, the MLME shall start synchronization using the neighbor offset synchronization method with the specified peer STA. Upon receipt of an MLME-MESHNEIGHBOROFFSETSYNCSTOP.request primitive, the MLME shall stop synchronization using the neighbor offset synchronization method with the specified peer STA.

A mesh STA that utilizes the neighbor offset synchronization method may start its TSF timer independently of other mesh STAs. The mesh STA shall calculate the timing offset value with respect to the neighbor STA, with which it maintains synchronization, as described in 11C.12.2.2.2. The mesh STA shall adjust its TSF timer based on time stamps received in Beacon or Probe Response frames from neighbor STAs, with which it maintains synchronization, as described in 11C.12.2.2.3.

When the mesh STA alternates Awake state and Doze state, it may not listen to the Beacon frames of a neighbor mesh STA, with which it maintains synchronization, at all times. However, it shall conform to the clock drift compensation procedures and TSF jitter allowance as described in 11C.12.2.2.3. See Y.3.6 for more guidelines.

11C.12.2.2.2 Timing offset calculation

When dot11MeshActiveSynchronizationMethod is neighborOffsetSynchronization (1), the mesh STA shall calculate the timing offset value with respect to the neighbor STA with which it maintains synchronization. The calculation of the timing offset value is based on time stamps from the received Beacon and Probe Response frames as follows:

$$T_{offset} = T_t - T_r$$

where

T_{offset} is the timing offset value

T_t is the value in the Timestamp field in the received frame

T_r is the frame reception time measured in the TSF timer of the mesh STA

The offset value is represented as a signed integer. The unit of the offset value is µs. The mesh STA shall keep the T_{offset} value calculated from the latest Beacon or Probe Response frame received from each neighbor STA with which it maintains synchronization.

A mesh STA may translate the time measured in the TSF of the neighbor STA into the time base of its own TSF as follows:

$$T_{self} = T_{neighbor} - T_{offset}$$

where

T_{self} is the translated time in its own TSF

$T_{neighbor}$ is the time measured in the TSF timer of the neighbor STA

Upon receipt of an MLME-MESHNEIGHBOROFFSETCALCULATE.request primitive, the MLME shall receive a Beacon or Probe Response frame from the specified neighbor STA, calculate the T_{offset} from the received frame, and report the calculated T_{offset} to the SME by responding with an MLME-MESHNEIGHBOROFFSETCALCULATE.confirm primitive. T_{offset} is used to provide the timing reference of neighbor STAs.

11C.12.2.2.3 Clock drift adjustment

When dot11MeshActiveSynchronizationMethod is neighborOffsetSynchronization (1), the mesh STA shall examine the reception time of the Beacon frames from neighbor STAs with which it maintains synchronization and adjust its TSF timer to compensate the relative timing error among neighbor mesh STAs caused by the clock drift. The mesh STA adjusts its TSF so that its TSF counting is aligned to the most delayed neighbor STA.

When the mesh STA receives a Beacon frame or a Probe Response frame from one of the neighbor STAs with which it maintains synchronization, the mesh STA shall perform the following measurement procedure:

a) The mesh STA checks if the transmitter of the Beacon frame or Probe Response frame is in the process of the TBTT adjustment (see 11C.12.4.4.3). If the received frame contains the Mesh Configuration element and the TBTT Adjusting subfield in the Mesh Configuration field is 1, the mesh STA shall invalidate the T_{offset} value for this neighbor STA and shall not perform the following steps.

b) The mesh STA checks if it has a valid T_{offset} value obtained from the previous Beacon or Probe Response frame reception from the transmitter of the received frame. If it does not have the valid T_{offset} value, it shall not perform the following steps.

c) The mesh STA calculates the clock drift amount $T_{ClockDrift}$ by comparing the $T_{offset,p}$,the offset value obtained previously for this neighbor STA, and the $T_{offset,c}$, the offset value obtained from the current frame reception.

$$T_{ClockDrift} = T_{offset,\,p} - T_{offset,\,c}$$

where $T_{ClockDrift}$ is the clock drift amount in µs represented as a signed integer.

d) The mesh STA shall compare the $T_{ClockDrift}$ value with the $T_{MaxClockDrift}$, the largest $T_{ClockDrift}$ value obtained from other neighbor STA within this beacon period. If the $T_{ClockDrift}$ value for this neighbor STA is greater than the $T_{MaxClockDrift}$ value, the mesh STA replaces the $T_{MaxClockDrift}$ value with the $T_{ClockDrift}$ value, in order to determine the largest $T_{ClockDrift}$ value among neighbor STAs.

When the previous $T_{ClockDrift}$ values have been stable for a neighbor mesh STA, the mesh STA may substitute the previous $T_{ClockDrift}$ value for the $T_{ClockDrift}$ value in the measurement procedure and process the step d), at the time of a TBTT of the neighbor STA, without receiving a Beacon frame.

Before the mesh STA transmits a Beacon frame, it shall perform the following adjustment procedure:

— The mesh STA checks if the current $T_{MaxClockDrift}$ value is greater than zero. If the $T_{MaxClockDrift}$ value is greater than zero, it shall continue the following steps. Otherwise, it shall initialize the $T_{MaxClockDrift}$ with zero and shall not perform the following step.

— If the $T_{MaxClockDrift}$ value is smaller than 0.04% of its beacon interval, the mesh STA shall adjust its TSF timer so that the next TBTT will be delayed for the duration of the $T_{MaxClockDrift}$ and initialize the $T_{MaxClockDrift}$ value with zero. Otherwise, it shall adjust its TSF timer so that the next TBTT will be delayed for the duration of 0.04% of its beacon interval and subtract the value of 0.04% of its beacon interval from the $T_{MaxClockDrift}$.

The mesh STA may adjust its TSF timer only to slow the counting. The mesh STA may adjust its TSF timer within the range of 0.04% in a beacon period.

When the delay amount at each beacon periods are not stable, the mesh STA should listen to neighbor STA's Beacon frames frequently. An implementation may circumvent the potential jitter of the TSF timer introduced by the adjustment procedure by adopting additional adjustment to the $T_{MaxClockDrift}$, which is beyond the scope of the standard, as long as the mesh STA's TSF counting is aligned to the most delaying neighbor STA without introducing unnecessary delay of its TSF counting.

NOTE—This clock drift compensation procedure does not intend to maintain a strict synchronization. It aims to stop TBTT drifting away among neighbor mesh STAs, allowing some jitter of TSF timer.

11C.12.3 Beaconing

11C.12.3.1 Beacon generation in MBSSs

A mesh STA transmits Beacon frames that are specific to an MBSS. Beacon frames for MBSS, infrastructure BSS, or IBSS are differentiated by the Capability Information field in the Beacon frame as specified in 7.3.1.4. A mesh STA that collocates with an AP generates Beacon frames for the MBSS independently of the AP.

The mesh STA shall define a series of TBTTs exactly dot11BeaconPeriod TUs apart. Time zero is defined to be a TBTT with the Beacon frame being a DTIM. At each TBTT, the mesh STA shall schedule a Beacon frame as the next frame for transmission according to the medium access rules specified in Clause 9. The beacon period is included in Beacon and Probe Response frames.

The mesh STA shall start beaconing upon the receipt of the MLME-START.request primitive.

11C.12.3.2 Beacon reception for mesh STA

A mesh STA shall use information from the Timestamp field without regard to the BSSID or Mesh ID in order to obtain information necessary for synchronization, if the mesh STA maintains synchronization with the transmitter of the Beacon frame. A mesh STA may use information from the Beacon interval field and the Beacon Timing element without regard for the Mesh ID in order to obtain information necessary for MBCA, if the mesh STA maintains synchronization with the transmitter of the Beacon frame and dot11MBCAActivated is true. A mesh STA may use information from the MCCAOP Advertisement Overview element and MCCAOP Advertisement element without regard for the Mesh ID in order to obtain information necessary for MCCA, if the mesh STA maintains synchronization with the transmitter of the Beacon frame and dot11MCCAActivated is true.

A mesh STA in a mesh BSS shall use information that is not in the CF Parameter Set element, the Timestamp field, the Beacon interval field, the Beacon Timing element, the MCCAOP Advertisement Overview element, or the MCCAOP Advertisement element in received Beacon frames only if the mesh STA maintains a mesh peering with the transmitter of the Beacon frame.

11C.12.4 Mesh beacon collision avoidance (MBCA)

11C.12.4.1 Overview

Mesh STAs use the mesh beacon collision avoidance (MBCA) protocol to detect and mitigate collisions among Beacon frames transmitted by other STAs (including mesh STAs, APs, and STAs in an IBSS) on the same channel within the range of 2 hops. MBCA mitigates hidden node problems with respect to Beacon frames.

NOTE—Beacon frames are transmitted without acknowledgement and might collide with other frames. In a mesh BSS, multiple STAs transmit Beacon frames periodically, and mesh STAs might be located out of range of each other. This implies that Beacon frames might suffer from the so-called hidden node problem and might not be received by neighbor STAs. Once Beacon frames from hidden STAs start to collide, Beacon frames keep on colliding if these hidden STAs transmit Beacon frames at the same beacon interval that is a typical operation. MBCA provides a set of rules to mitigate this problem.

When dot11MBCAActivated is true, the mesh STA shall set the MBCA Enabled subfield in the Mesh Capability field of the Mesh Configuration element to 1.

MBCA is composed of beacon timing advertisements, TBTT selection, and TBTT adjustment. When dot11MBCAActivated is true, the mesh STA advertises the TBTT and beacon interval of its neighbor STAs through the Beacon Timing element as described in 11C.12.4.2. Upon reception of the Beacon Timing element, the mesh STA obtains the beacon timing information of its neighbor mesh STAs and uses this information for its TBTT selection and TBTT adjustment as described in 11C.12.4.3 and 11C.12.4.4. The mesh STA may also perform additional procedures described in 11C.12.4.5 and 11C.12.4.6.

When dot11MBCAActivated is true, the mesh STA that alternates Awake state and Doze state should listen to Beacon frames from its neighbor STAs, with which it maintains synchronization, often, in order to advertise and obtain the recent TBTT information.

11C.12.4.2 Beacon timing advertisement

11C.12.4.2.1 General

When dot11MBCAActivated is true, the mesh STA shall contain Beacon Timing element in Beacon and Probe Response frames in order to advertise its beacon timing information. The mesh STA calculates the TBTT of its neighbor STAs with which it maintains synchronization as described in 11C.12.4.2.2, and composes beacon timing information as described in 11C.12.4.2.3. The mesh STA collects the beacon timing information from each neighbor STA with which it maintains synchronization. The collection of the beacon timing information is termed “beacon timing information set.” The mesh STA contains whole or part of the beacon timing information set in the Beacon Timing element as described in 11C.12.4.2.5. The mesh STA also maintains the status number of the beacon timing information set and contains the status number in the Beacon Timing element as described in 11C.12.4.2.4. The receiver of the Beacon Timing element uses the received beacon timing information as described in 11C.12.4.2.6.

11C.12.4.2.2 Calculation of neighbor STA’s TBTT

When a Beacon frame is received from one of its neighbor STAs with which the mesh STA maintains synchronization, the mesh STA shall calculate the TBTT of the received Beacon frame as follows:

$$T_{TBTT} = T_r - (T_t \text{ modulo } (T_{BeaconInterval} \times 1024))$$

where

T_{TBTT} is the calculated TBTT

T_r is the frame reception time measured in the TSF timer of the receiving mesh STA

T_t is the value in the Timestamp field in the received frame

$T_{BeaconInterval}$ is the value in the Beacon interval field in the received frame

The T_{TBTT} is used as described in 11C.12.4.2.3.

Further, the mesh STA shall calculate the time difference between the TBTT of the received Beacon frame and the time predicted from the past TBTT as follows:

$$T_{Delta} = | T_{TBTT,\,c} - (T_{TBTT,\,p} + (T_{BeaconInterval} \times N_{Count})) |$$

where

T_{Delta} is the time difference

$T_{TBTT,\,c}$ is the TBTT calculated from the received Beacon frame

$T_{TBTT,\,p}$ is the TBTT calculated for the first time after the latest status number update (see 11C.12.4.2.4)

$T_{BeaconInterval}$ is the value in the Beacon interval field in the received Beacon frame

N_{Count} is the number of TBTTs since $T_{TBTT,\,p}$ has been calculated

T_{Delta} is used to maintain the status number described in 11C.12.4.2.4.

11C.12.4.2.3 Beacon timing information

The mesh STA shall keep the latest T_{TBTT} together with the Beacon interval contained in the received frame and the identifier of the neighbor STA as the beacon timing information with respect to the neighbor STA. When the elapsed time since the latest Beacon frame reception is smaller than 524 288 TU, the beacon timing information is valid.

NOTE—The beacon timing information provides the time reference for a series of the TBTTs of the corresponding STA. Using the beacon timing information, a mesh STA is able to predict future TBTTs by adding the reported beacon interval to the reported TBTT.

The mesh STA shall collect the valid beacon timing information from each neighbor STA with which it maintains synchronization and keep the collection as the beacon timing information set. The beacon timing information set is advertised to its neighbor mesh STAs through the Beacon Timing element as described in 11C.12.4.2.5.

When the amount of neighbors, for which valid beacon timing information is kept, is large, the beacon timing information set may be divided into multiple tuples of beacon timing information. In such case, a tuple of beacon timing information is included in the Beacon Timing element (see 11C.12.4.2.5).

11C.12.4.2.4 Maintenance of the status number

The mesh STA shall maintain the status number of the beacon timing information set. The status number is set to a value from a single modulo-16 counter, starting at 0 and incrementing by 1 for each transmission of a frame containing the Beacon Timing element after the mesh STA encountered any of the following events:

a) It starts or stops maintaining synchronization with a neighbor STA.

b) It receives a Beacon frame from a neighbor STA with which it maintains synchronization and the calculated T_{Delta} (see 11C.12.4.2.2) is larger than 255 µs.

c) It completes the TBTT adjustment procedure described in 11C.12.4.4.3.

The mesh STA shall set the Status Number subfield in the Report Control field in the Beacon Timing element to the status number. The Status Number subfield in the Report Control field facilitates the detection of the changes in the beacon timing information set.

11C.12.4.2.5 Transmitter’s procedure

When dot11MBCAActivated is true, the mesh STA shall report the TBTT and beacon interval of its neighbor STAs through the Beacon Timing element as described in this subclause.

The Beacon Timing element reports on timing information of the Beacon frames that are received from the neighbor STAs with which the mesh STA maintains synchronization on the operating channel. The mesh STA shall include the Beacon Timing element in Probe Response frames and in TBTT Adjustment Request frames. The mesh STA shall also include the Beacon Timing element in Beacon frames as specified by dot11MeshBeaconTimingReportInterval and dot11MeshBeaconTimingReportMaxNum. The Beacon Timing element is present in a Beacon frame when the DTIM Count value in the Beacon frame is zero or equal to an integer multiple of dot11MeshBeaconTimingReportInterval.

The maximum number of Beacon Timing Information fields contained in a Beacon Timing element is limited to dot11MeshBeaconTimingReportMaxNum for Beacon frames, or is limited by the maximum information element size for other frames. When the number of neighbors, for which valid beacon timing information is kept, is equal or smaller than the limit, the mesh STA shall include all the beacon timing information in a single Beacon Timing element, setting both Beacon Timing Element Number and More Beacon Timing Elements subfield in the Report Control field to 0. When the number of neighbors, for which valid beacon timing information is kept, exceeds the limit, the mesh STA shall divide the beacon timing information set into multiple tuples and assign each tuple with an index number starting from 0. When the beacon timing information set is divided, the mesh STA shall include the successive tuples of beacon timing information in the Beacon Timing elements. In this case, the mesh STA shall set the Beacon Timing Element Number subfield in the Report Control field to the index number of the tuple. The mesh STA shall set the More Beacon Timing Elements subfield in the Report Control field to 1 when it has one or more beacon timing information tuples with a larger index number. The mesh STA shall divide the beacon timing information set into no more than N_Info tuples, where N_Info = ceil(number of valid beacon timing information / maximum number of Beacon Timing Information fields in the Beacon Timing element).

The mesh STA may update the combination of the tuples only when the status number described in 11C.12.4.2.4 is updated. The mesh STA shall include newly updated beacon timing information (i.e., beacon timing information that causes an update of the status number as described in 11C.12.4.2.4) in the tuple with a smaller index number. When the status number is updated, the mesh STA shall include the tuple of beacon timing information indexed as 0 in the Beacon Timing element in the subsequent Beacon frame. Successive tuples shall be transmitted in ascending order of the index number in the successive Beacon frames.

NOTE—The standard does not impose mesh STAs to advertise a fragmented beacon timing information set sequentially in its Beacon frames at all times. This implies that the mesh STA might advertise tuples with a smaller index number more frequently, which is useful to notify new beacon timing information efficiently.

When the mesh STA receives a Probe Request frame containing a Beacon Timing element ID in its Request element, it shall respond with a Probe Response frame containing the Beacon Timing element. If all beacon timing information cannot be contained in a Beacon Timing element, the mesh STA shall include multiple Beacon Timing elements containing successive tuples of beacon timing information in the order of the Request element (see Table 7-15) so that all tuples are transmitted.

11C.12.4.2.6 Receiver's procedure

A mesh STA with dot11MBCAActivated equal to true that receives a Beacon Timing element obtains the beacon timing information of its neighbor mesh STA and uses it for its TBTT selection and TBTT adjustment as described in 11C.12.4.3 and 11C.12.4.4.

When a mesh STA receives a Beacon frame with a Beacon Timing element that contains only a subset of the beacon timing information set, the mesh STA may transmit a Probe Request frame containing a Beacon Timing element ID in its Request Information element to the transmitter of the Beacon Timing element, in order to request the rest of the beacon timing information.

NOTE 1—The Report Control field in the Beacon Timing element facilitates the detection of the missing beacon timing information.

NOTE 2—Once the entire beacon timing information set with a particular Status Number is obtained, the mesh STA does not need to retrieve beacon timing information as long as the Status Number remains the same.

A mesh STA that receives the Beacon Timing element shall record the reported TBTT and its successive TBTTs as neighbor's essential beacon reception timing if the MSB of the Neighbor STA ID field in the corresponding Beacon Timing Information field is 0. The essential beacon reception timing is used to control the transmission of frames as described in 11C.12.4.5.

A mesh STA can also check if its neighbor mesh STAs received its Beacon frame successfully by checking whether the Beacon Timing elements received from its neighbor mesh STAs contain beacon timing information of the mesh STA. When the Beacon Timing element is received from one of the peer mesh STAs, the mesh STA checks if the MSB of the Neighbor STA ID subfield is set to 0 and the rest of the field matches with the 7 LSBs of the AID value assigned to the mesh STA through the mesh peering establishment. When the Beacon Timing element is received from a non-peer mesh STA, the mesh STA checks if the MSB of the Neighbor STA ID subfield is set to 1 and the rest of the field matches with the 7 LSBs of its own MAC address (taking the I/G bit as the MSB). If the matching is verified, the corresponding beacon timing information represents the correct beacon reception by the neighbor mesh STA.

If a Beacon frame is received from a neighbor peer mesh STA that is either in active mode or in light sleep mode, the Beacon Timing element is present in the frame, and all beacon timing information is contained in the Beacon Timing element, the mesh STA shall verify whether the neighbor peer mesh STA received its Beacon frame. If the Beacon Timing element does not contain beacon timing information of the mesh STA or the Neighbor TBTT subfield of the corresponding beacon timing information does not reflect the recent

TBTT of the mesh STA, the mesh STA considers the previous Beacon frame was not received by the neighbor peer mesh STA.

11C.12.4.3 TBTT selection

When dot11MBCAActivated is true, the mesh STA performs the TBTT selection described herein before it starts beaconing (see 11C.2.8). The mesh STA selects its TBTTs and its beacon interval so that its Beacon frames do not collide with Beacon frames transmitted by other STAs in its 2 hop range.

Before the mesh STA starts beaconing, it performs scanning and discovered neighbor STAs are reported through an MLME-SCAN.confirm primitive (see 11C.2.6). Using TimeStamp, Local Time, and Beacon Period in the BSSDescription parameter provided by the MLME-SCAN.confirm primitive, the mesh STA shall obtain the TBTT and beacon interval of its neighbor STAs operating on the same channel as the mesh STA starts to operate. The mesh STA shall also collect the beacon timing information contained in the Beacon Timing elements received on the channel through Beacon Timing in the BSSDescription parameter provided by the MLME-SCAN.confirm primitive, in order to obtain the TBTT and beacon interval of STAs in 2 hop range. After obtaining this information, the mesh STA shall look for a timing of its beacon transmissions so that its Beacon frames are likely not to collide with Beacon frames transmitted by other STAs in its 2 hop range. The mesh STA shall update its TSF timer and select its beacon interval to set its TBTTs to the appropriate timing, and then it shall start beaconing using the MLME-START.request primitive.

11C.12.4.4 TBTT adjustment

11C.12.4.4.1 Self-determined TBTT adjustment

When dot11MBCAActivated is true, the mesh STA checks if it does not transmit Beacon frames during the beacon transmissions of other STAs within its 2 hop range using the Beacon Timing element received from its neighbor peer mesh STA.

When the mesh STA discovers that its Beacon frames repeatedly collide with the Beacon frames of a neighbor or a neighbor’s neighbor and its TBTT comes later than the TBTT of the colliding STA at the time of collision, it shall perform the TBTT scanning procedure described in 11C.12.4.4.3. If the mesh STA finds an alternative TBTT, it shall start the TBTT adjustment procedure as described in 11C.12.4.4.3.

11C.12.4.4.2 Requested TBTT adjustment

When a mesh STA discovers that Beacon frames from two or more neighbor STAs are colliding repeatedly or a series of TBTTs are close enough to trigger frequent beacon collisions, the mesh STA may transmit a TBTT Adjustment Request frame to the neighbor mesh STA of which the TBTT comes last at a particular collision timing in order to request this neighbor mesh STA to adjust its TBTT. The TBTT Adjustment Request frame may be transmitted only if the following conditions hold:

- The recipient of the TBTT Adjustment Request frame is a peer mesh STA and has set the MBCA Enabled subfield in the Mesh Capability field of the Mesh Configuration element to 1.
- The other colliding STA does not include the Mesh Configuration element in its Beacon frames or the TBTT Adjusting field in the Mesh Configuration element is 0.

When dot11MBCAActivated is true, the mesh STA that receives a TBTT Adjustment Request frame shall perform the TBTT scanning procedure described in 11C.12.4.4.3, and determine if it can find an appropriate alternative timing for its TBTTs. After the completion of the TBTT scanning procedure, the mesh STA that receives the TBTT Adjustment Request frame shall respond with a TBTT Adjustment Response frame containing the result of the TBTT scanning in the Status Code field. If the mesh STA finds an alternative TBTT, it shall agree with the request. If it agrees with the request, the Status Code field is set to 0 in the TBTT Adjustment Response frame, and it shall complete the TBTT adjustment procedure described in

11C.12.4.4.3. If it does not agree with the request, it shall indicate the reason in the Status Code field in the TBTT Adjustment Response frame. A mesh STA may set the Status Code to either 0, 1, or 78 in the TBTT Adjustment Response frame.

11C.12.4.4.3 TBTT scanning and adjustment procedures

When a mesh STA is in need of TBTT adjustment, it tries to find an alternative TBTT first. The mesh STA shall perform the TBTT scanning procedure as follows:

a) The mesh STA checks if its beacon timing information and collected neighbor’s beacon timing information are sufficiently new. If the mesh STA did not receive a Beacon frame from a neighbor STA with which it maintains synchronization at the latest TBTT, it shall receive a Beacon or Probe Response frame from the neighbor STA and obtain the TBTT of the neighbor STA and the beacon timing information contained in the Beacon Timing element.

NOTE 1—This is particularly important if the mesh STA is in deep sleep mode for a neighbor peer mesh STA.

b) Using the latest TBTT of its neighbor STAs and the latest beacon timing information of neighbor mesh STAs, the mesh STA shall look for an alternative TBTT that does not cause beacon collision among the STAs in its 2 hop range.

If an alternative TBTT is not available, the mesh STA terminates the procedure. If an alternative TBTT is available, the mesh STA shall start the TBTT adjustment procedure as follows:

c) The mesh STA shall set the TBTT Adjusting field in the Mesh Configuration element to 1 in order to announce that the TBTT adjustment procedure is ongoing.

d) The mesh STA shall suspend its TSF timer for a period of time, no longer than half of the Group Delivery Idle Time (defined in 11C.13.5) within a single beacon period, to slow its TSF.

e) The mesh STA shall adjust TBTT information of the neighbor STAs (see 11C.12.4.2.3), that are to be contained in the Beacon Timing element, accordingly by subtracting the delay amount.

f) When dot11MCCAActivated is true, the mesh STA shall adjust the MCCAOP reservations accordingly by modifying the MCCAOP Offset of each MCCAOP reservation. See 9.9a.3.3.

g) The mesh STA shall repeat suspending its TSF timer over multiple beacon periods until its TBTT is set to the alternative TBTT.

h) Upon completion of the TBTT adjustment, the mesh STA shall update the status number as described in 11C.12.4.2.4 and shall set the TBTT Adjusting field in the Mesh Configuration element to 0.

NOTE 2—A mesh STA in deep sleep mode might interpret its neighbor mesh STA’s TBTT adjustment as a large TSF jitter. When a mesh STA in deep sleep mode observes a large TSF jitter and the Status Number in the Report Control field in the Beacon Timing element of the received Beacon frame (or Probe Response frame) has been updated, the mesh STA in deep sleep mode should not take this jitter as clock drift and listen to the next Beacon frame to verify if the clock drift is large.

11C.12.4.5 Frame transmission across reported TBTT

When dot11MBCAActivated is true, the mesh STA should not extend its transmissions across TBTT of its neighbor STAs with which it maintains synchronization. Further, the mesh STA should not extend its transmissions, other than Beacon frames, across all essential beacon reception timing (see 11C.12.4.2.6) reported from its neighbor mesh STAs with which it maintains synchronization. This operation helps in reducing the hidden STA interference with beacon reception at its neighbor mesh STAs. When both dot11MBCAActivated and dot11MCCAActivated are true, the mesh STA shall not extend its transmissions across TBTT of its neighbor STAs with which it maintains synchronization. Further, the mesh STA shall not extend its transmissions, other than Beacon frames, across all beacon reception timing reported from its neighbor mesh STAs with which it maintains synchronization.

After silencing for dot11MeshAverageBeaconFrameDuration µs from the reported neighbor's TBTT, the mesh STA may start transmitting frames again.

11C.12.4.6 Delayed beacon transmissions

A mesh STA may occasionally delay its Beacon frame transmission from its TBTT for a pseudo-random time. This attribute is specified by dot11MeshDelayedBeaconTxInterval, dot11MeshDelayedBeaconTxMinDelay, and dot11MeshDelayedBeaconTxMaxDelay. When dot11MeshDelayedBeaconTxInterval is set to non-zero value, the mesh STA shall delay its Beacon frame transmission from TBTT, once every dot11MeshDelayedBeaconTxInterval. When the mesh STA transmits a Beacon frame with delay from its TBTT, the delay time shall be randomly selected between dot11MeshDelayedBeaconTxMinDelay and dot11MeshDelayedBeaconTxMaxDelay µs.

NOTE—Delayed beacon transmission allows mesh STAs to discover Beacon frames that are transmitted from multiple mesh STAs with TBTTs close to each other. It is recommended to set dot11MeshDelayedBeaconTxMaxDelay to a time longer than the typical duration of Beacon frames.

11C.13 Power save in a mesh BSS

11C.13.1 General

A mesh STA may use mesh power modes to reduce its power consumption. A mesh STA manages each of its mesh peerings with a peer-specific mesh power mode as described in 11C.13.2.2. A mesh STA may set the mesh power mode for a mesh peering independently of the mesh power modes for its other mesh peerings. A mesh STA also manages a non-peer mesh power mode as described in 11C.13.2.3. When a mesh STA is in light sleep mode or in deep sleep mode for a mesh peering, the mesh STA shall maintain its mesh awake window as described in 11C.13.6.

A mesh STA shall have the capability to buffer frames and to perform mesh power mode tracking for the peer-specific mesh power modes of its peer mesh STAs, as described in 11C.13.7. A mesh STA shall use mesh peer service periods for individually addressed frame transmissions to neighbor peer mesh STAs that are either in light sleep mode or in deep sleep mode towards this mesh STA, as described in 11C.13.9. A mesh STA transmits group addressed frames after the Beacon frame containing DTIM when any of its peer mesh STAs is in light sleep mode or deep sleep mode for the mesh peering with the mesh STA (see 11C.13.4 and 11C.13.5). These capabilities are referred to as support for power save.

11C.13.2 Mesh power modes

11C.13.2.1 General

A mesh STA is in one of two different power states, Awake or Doze, as defined in 11.2.1.1.

The manner in which a mesh STA transitions between power states is determined by its peer-specific mesh power modes and its non-peer mesh power mode. A mesh STA shall be in Awake state if any of the conditions specified in 11C.13.8.6 is not fulfilled. A mesh STA maintains peer-specific mesh power modes for each of its mesh peerings as described in 11C.13.2.2. A mesh STA may have a different peer-specific mesh power mode for each mesh peering. A mesh STA maintains a non-peer mesh power mode for non-peer mesh STAs that is described in 11C.13.2.3. An example illustration of the use of peer specific and non-peer mesh power modes is shown in Figure 11C-5.

11C.13.2.2 Peer-specific mesh power modes

The peer-specific mesh power mode specifies the activity level of the mesh STA for the corresponding mesh peering. Three mesh power modes are defined: active mode, light sleep mode, and deep sleep mode. The peer-specific mesh power modes are defined as follows:

- *Active mode:* The mesh STA shall be in Awake state all the time.
- *Light sleep mode:* The mesh STA alternates between Awake and Doze states, as specified in 11C.13.8.4. The mesh STA shall listen to all the Beacon frames from the corresponding peer mesh STA.
- *Deep sleep mode:* The mesh STA alternates between Awake and Doze states, as specified in 11C.13.8.5. The mesh STA may choose not to listen to the Beacon frames from the corresponding peer mesh STA.

The combination of the Power Management field in the Frame Control field and the Mesh Power Save Level subfield in the QoS Control field contained in Mesh Data frames indicates the peer-specific mesh power mode as shown in the Table 11C-31.

Table 11C-31—Peer-specific mesh power mode definition

Activity level	Peer-specific mesh power mode	Power Management field	Mesh Power Save Level subfield
Highest ↑ Lowest	Active mode	0	Reserved
	Light sleep mode	1	0
	Deep sleep mode	1	1

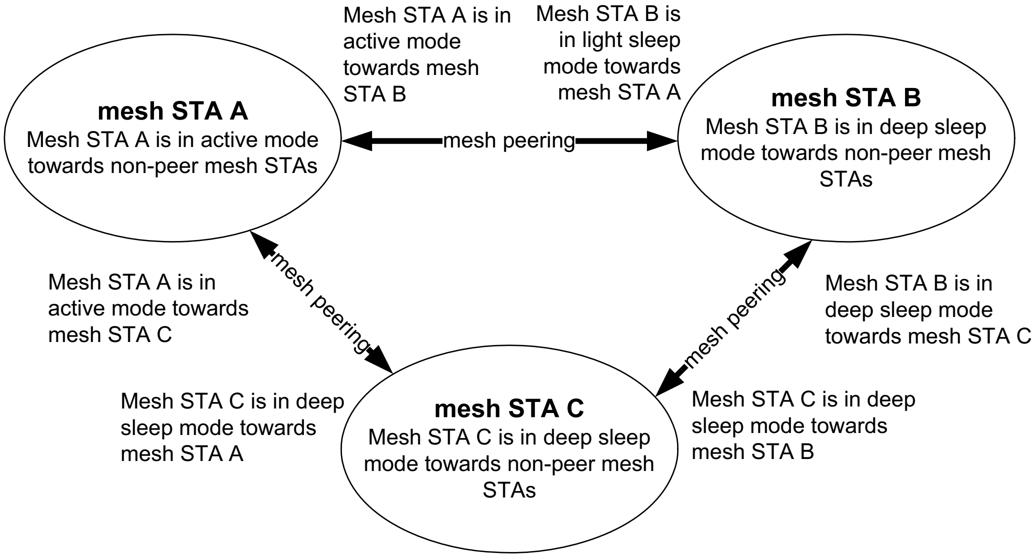

Mesh STA A is in active mode towards mesh STA B
Mesh STA B is in light sleep mode towards mesh STA A
mesh STA A
Mesh STA A is in active mode towards non-peer mesh STAs
mesh peering
mesh STA B
Mesh STA B is in deep sleep mode towards non-peer mesh STAs
mesh peering
Mesh STA A is in active mode towards mesh STA C
mesh peering
Mesh STA B is in deep sleep mode towards mesh STA C
Mesh STA C is in deep sleep mode towards mesh STA A
mesh STA C
Mesh STA C is in deep sleep mode towards non-peer mesh STAs
Mesh STA C is in deep sleep mode towards mesh STA B

Figure 11C-5—An example of mesh power mode usage

11C.13.2.3 Non-peer mesh power modes

The non-peer mesh power mode indicates the mesh power mode of the mesh STA toward the non-peer mesh STAs. Two non-peer mesh power modes are defined: active mode and deep sleep mode. The non-peer mesh power mode is indicated by the Power Management field in the Frame Control field in group addressed frames, management frames transmitted to non-peer neighbor STAs, and in Probe Response frames. When the Power Management field in the Frame Control field is set to 1, the non-peer mesh power mode is deep sleep mode. When the Power Management field in the Frame Control field is set to 0, the non-peer mesh power mode is active mode.

A mesh STA may send Probe Request and Mesh Peering Open Request frames to a non-peer mesh STA that sets its non-peer mesh power mode to deep sleep mode only during the mesh awake window of the mesh STA.

11C.13.3 Mesh power mode indications and transitions

11C.13.3.1 General

When a mesh STA is in active mode for a mesh peering, it shall set the Power Management field in the Frame Control field to 0 in all individually addressed Mesh Data or QoS Null frames transmitted to the corresponding peer mesh STA.

When a mesh STA is in light sleep mode for a mesh peering, it shall set the Power Management field in the Frame Control field to 1 and the Mesh Power Save Level subfield in the QoS Control field to 0 in all individually addressed Mesh Data or QoS Null frames transmitted to the corresponding peer mesh STA.

When a mesh STA is in deep sleep mode for a mesh peering, it shall set the Power Management field in the Frame Control field to 1 and the Mesh Power Save Level subfield in the QoS Control field to 1 in all individually addressed Mesh Data or QoS Null frames transmitted to the corresponding mesh STA.

When a mesh STA is in deep sleep mode for any of its mesh peerings, the Mesh Power Save Level subfield in the QoS Control field in group addressed Mesh Data frames and the Mesh Power Save Level subfield in the Mesh Capability field in the Mesh Configuration element shall be set to 1. When a mesh STA is not in deep sleep mode for any of its mesh peerings, these subfields shall be set to 1.

To change peer-specific mesh power modes, a mesh STA shall inform its peer mesh STAs through a successful frame exchange initiated by the mesh STA. The Power Management field in the Frame Control field and the Mesh Power Save Level subfield in the QoS Control field of the frame sent by the mesh STA in this exchange indicates the peer-specific mesh power mode that the STA shall adopt upon successful completion of the entire frame exchange.

The algorithm to trigger the change of a peer-specific mesh power mode is beyond the scope of this standard.

The non-peer mesh power mode is determined by the peer-specific mesh power modes of the mesh STA. When a mesh STA is in light sleep mode or deep sleep mode for at least one mesh peering, it shall set the non-peer mesh power mode to deep sleep mode.

When a mesh STA is in active mode for non-peer STAs, it shall set the Power Management field in the Frame Control field to 0 in group addressed frames, in management frames transmitted to non-peer mesh STAs, and in Probe Response frames.

When a mesh STA is in deep sleep mode for non-peer STAs, it shall set the Power Management field in the Frame Control field to 1 in group addressed frames, in management frames transmitted to non-peer mesh STAs, and in Probe Response frames.

11C.13.3.2 Transition to a higher activity level

A mesh STA may use group addressed or individually addressed Mesh Data or QoS Null frames to change its mesh power mode to a higher activity level, for example; from deep sleep to light sleep or to active mode; or from light sleep to active mode.

Individually addressed frames may be used to temporarily raise the activity level of the mesh STA for a mesh peering. This is useful in cases when a link temporarily requires efficient data transmission with the peer mesh STA and the mesh STA desires to be able to transit back to lower activity level without performing the mesh power mode transition signaling with all peer mesh STAs.

11C.13.3.3 Transition to a lower activity level

A mesh STA shall use acknowledged individually addressed Mesh Data or QoS Null frames to change its peer-specific mesh power mode to a lower activity level, for example; from active mode to light or deep sleep mode; or from light sleep to deep sleep mode.

11C.13.4 TIM transmissions in an MBSS

The TIM element identifies the peer mesh STAs for which traffic is pending and buffered in the reporting mesh STA. This information is coded in a partial virtual bitmap, as described in 7.3.2.6. In addition, the TIM contains an indication whether group addressed traffic is pending. Every neighbor peer mesh STA is assigned an AID by the reporting mesh STA as part of the mesh peering establishment process (see 11C.3.1). The mesh STA shall identify those peer mesh STAs for which it is prepared to deliver buffered MSDUs and MMPDUs by setting bits in the TIM's partial virtual bitmap that correspond to the appropriate AIDs.

11C.13.5 TIM types

There are two different TIM types: TIM and DTIM. A mesh STA shall transmit a TIM with every Beacon frame. Every DTIMPeriod, a TIM of type DTIM is transmitted with a Beacon frame. After transmitting a Beacon containing a DTIM, the mesh STA shall send the buffered group addressed MSDUs and MMPDUs, before transmitting any individually addressed frames. The More Data field of each group addressed frame shall be set to indicate the presence of further buffered group addressed MSDUs and MMPDUs. The mesh STA sets the More Data field to 0 in the last transmitted group addressed frame following the transmission of the DTIM Beacon.

When a mesh STA expects to receive a group addressed frame and CCA is IDLE for the duration of the PHY specific Group Delivery Idle Time, the receiving mesh STA may assume that no more frames destined to group addresses will be transmitted and may return to Doze state. The Group Delivery Idle Time is identical to the TXOP Limit for AC_VI specified by the default EDCA Parameter Set shown in Table 7-37.

11C.13.6 Mesh awake window

A mesh STA shall be in Awake state when its mesh awake window is active. A mesh awake window is active after the Beacon and Probe Response frames containing the Mesh Awake Window element. A mesh STA shall include the Mesh Awake Window element in its DTIM Beacon frames and may include the Mesh Awake Window element in its TIM Beacon and Probe Response frames. A mesh STA that operates in light sleep mode or deep sleep mode for any of its mesh peerings shall include the Mesh Awake Window element in its Beacon frame if the Beacon frame indicates buffered traffic for at least one peer mesh STA. The start of the mesh awake window is measured from the end of the Beacon or Probe Response transmission. The duration of the mesh awake window period is specified by dot11MeshAwakeWindowDuration. A mesh STA shall set the Mesh Awake Window field in the Mesh Awake Window element to dot11MeshAwakeWindowDuration. If the Mesh Awake Window element is not contained in the Beacon frame of a mesh STA, the duration of the mesh awake window period following this beacon is zero.

If the mesh STA that has its mesh awake window active transmits frames destined to group addresses, the duration of the mesh awake window is extended by an additional PostAwakeDuration. The PostAwakeDuration follows the group address frame, and the mesh STA that has its mesh awake window active shall stay in Awake state until it has transmitted all of its group addressed frames and the PostAwakeDuration has expired. The PostAwakeDuration is equal to duration of the mesh awake window.

A mesh STA may send a frame to a peer mesh STA that is in light sleep mode or deep sleep mode for the corresponding mesh peering during the mesh awake window of this peer mesh STA. When a peer trigger frame is successfully transmitted it initiates a mesh peer service period as described in 11C.13.9.

A mesh STA may send class 1 or class 2 frames, such as Probe Request or Mesh Peering Open frames, to a non-peer mesh STA that is in deep sleep mode for non-peer mesh STAs during the mesh awake window of this non-peer mesh STA.

11C.13.7 Power save support

As described in 11C.13.2, a mesh STA indicates its peer-specific mesh power modes and performs mesh power mode tracking of the peer-specific mesh power modes of its peer mesh STAs. A mesh STA shall not arbitrarily transmit frames to mesh STAs operating in a light or deep sleep mode, but shall buffer frames and only transmit them at designated times.

A mesh STA shall only transmit frames to a mesh STA operating in a light or deep sleep mode if the recipient mesh STA is in the Awake state as defined in 11C.13.8.4, 11C.13.8.5, and 11C.13.9; otherwise, the mesh STA shall buffer frames.

As described in 11C.13.4, a mesh STA indicates the presence of buffered traffic in TIM elements for all peer mesh STAs that operate in light or deep sleep mode towards the mesh STA. The mesh STA sets the bit for AID 0 (zero) in the bit map control field of the TIM element to 1 when group addressed traffic is buffered, according to 7.3.2.6. As described in 11C.13.5, a mesh STA transmits its group addressed frames after its DTIM Beacon if any of its peer mesh STA is in light or deep sleep mode towards the mesh STA.

As described in 11C.13.9, mesh peer service periods are used for frame transmissions towards a mesh STA that operates in light or deep sleep mode. Mesh peer service periods are not used in frame exchanges towards active mode mesh STAs.

A mesh STA may initiate a mesh peer service period with a peer mesh STA in deep or light sleep mode by transmitting a peer trigger frame when the mesh awake window of the peer mesh STA is active.

11C.13.8 Operation in peer-specific and non-peer mesh power modes

11C.13.8.1 General

Detailed operations of mesh STA in each mesh power mode are described in the following subclauses. Figure 11C-6 depicts example power state transitions of mesh STAs, when three mesh STAs are in the mesh power modes shown in the Figure 11C-5.

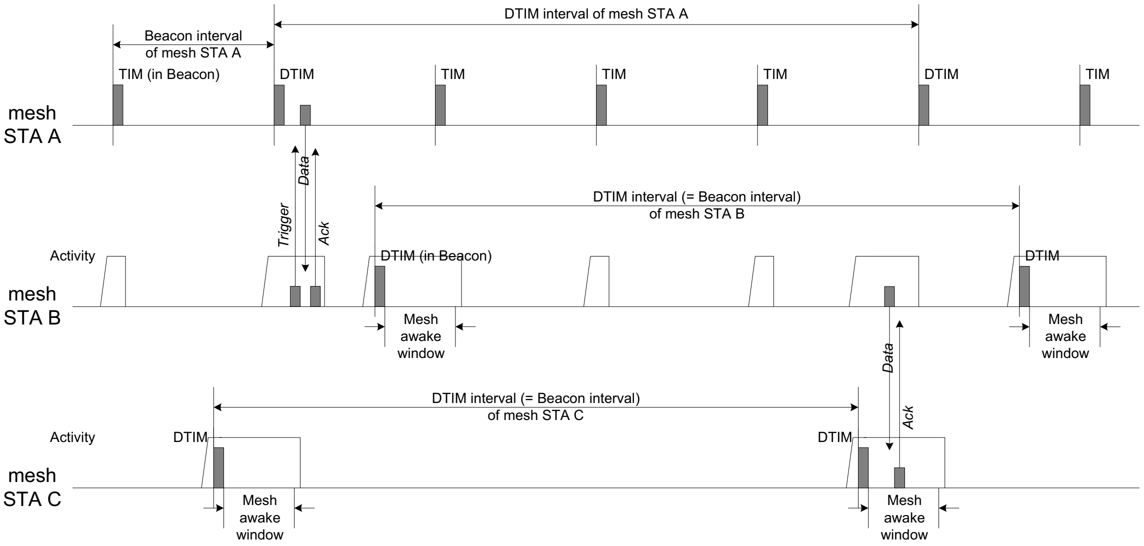


Figure 11C-6—Mesh power management operation

11C.13.8.2 Operation in active mode

When a mesh STA is in active mode for a mesh peering or for non-peer mesh STAs, it shall be in Awake state. Mesh peer service periods are not used in frame exchanges towards mesh STAs that are in active mode.

An active mode mesh STA may receive peer trigger frames from a peer mesh STA in light or deep sleep mode when there is no mesh peer service period ongoing between the peer mesh STAs.

11C.13.8.3 Operation in deep sleep mode for non-peer mesh STAs

If a mesh STA is in deep sleep mode for non-peer mesh STAs, it shall enter the Awake state prior to every TBTT of its own and shall remain in Awake state after the beacon transmission for the duration of the mesh awake window and the duration of its group addressed frame transmissions. The mesh STA may receive frames during its mesh awake window as described in 11C.13.6.

When receiving a frame initiating a mesh peering management procedure, an authentication procedure, or a passive scanning procedure, a mesh STA in deep sleep mode for non-peer mesh STAs shall operate in Awake state at least until the completion of the mesh peering management procedure (see 11C.3 and 11C.5), until the completion of the authentication procedure (see 11C.3.1 and 11C.3.3), or the transmission of the Probe Response frame.

If a mesh STA receives a peer trigger frame initiating a mesh peer service period from a peer mesh STA, the mesh STA shall remain in Awake state until the mesh peer service period is terminated as defined in 11C.13.9.4.

A mesh STA may return to Doze state after its mesh awake window if no frame initiating a response transaction or a mesh peer service period is received during the mesh awake window.

11C.13.8.4 Operation in light sleep mode for a mesh peering

If a mesh STA is in light sleep mode for a mesh peering, it shall enter the Awake state prior to every TBTT of the corresponding peer mesh STA to receive the Beacon frame from the peer mesh STA. The mesh STA may return to the Doze state after the beacon reception from this peer mesh STA, if the peer mesh STA did not indicate buffered individually addressed or group addressed frames. If an indication of buffered individually addressed frames is received, the light sleep mode mesh STA shall send a peer trigger frame with the RSPI field set to 1 to initiate a mesh peer service period with the mesh STA that transmitted the Beacon frame (see 11C.13.9.2). If an indication of buffered group addressed frames is received, the light sleep mode mesh STA shall remain in Awake state after the DTIM Beacon reception to receive group addressed frames The mesh STA shall remain Awake state until the More Data field of a received group addressed frame is set to 0 or if no group addressed frame is received within the PHY specific Group Delivery Idle Time. (See 11C.13.5.)

NOTE—When a mesh STA is in light sleep mode for a mesh peering, it sets its non-peer mesh power mode to deep sleep mode. This implies that a mesh STA operating in light sleep mode a mesh peering is required to conform to the rules described in 11C.13.8.3.

11C.13.8.5 Operation in deep sleep mode for a mesh peering

A mesh STA operating in deep sleep mode for a mesh peering might not receive Beacon frames from the corresponding peer mesh STA. The logic of how the mesh STA in deep sleep mode maintains synchronization among neighbors is beyond the scope of this standard. Guidance for the synchronization maintenance by the mesh STA in deep sleep mode is given in Y.3.6.

NOTE—When a mesh STA is in deep sleep mode for a mesh peering, it sets its non-peer mesh power mode to deep sleep mode. This implies that a mesh STA operating in deep sleep mode for a mesh peering is required to conform to the rules described in 11C.13.8.3.

11C.13.8.6 Conditions for Doze state

A mesh STA may enter Doze state if all of the following conditions are fulfilled:

- The mesh STA operates in light sleep mode or deep sleep mode for all of its mesh peerings, as described in 11C.13.8.4 or 11C.13.8.5
- The mesh STA has no mesh peer service period ongoing, as described in 11C.13.9
- The mesh STA has no pending transaction of mesh peering management, authentication, nor passive scanning (see 11C.13.8.3)
- The mesh awake window indicated by the mesh STA has expired, as described in 11C.13.6
- The mesh STA has terminated its group addressed frames delivery sequence after its DTIM Beacon, as described in 11C.13.5

Guidance for using the power save in mesh BSS and default parameter values are given in Y.3.

11C.13.9 Mesh peer service periods

11C.13.9.1 General

Mesh peer service periods are used for individually addressed frame exchanges between neighbor peer mesh STAs in which at least one of the mesh STAs is in light or deep sleep mode for the corresponding mesh peering. A mesh peer service period is a contiguous period of time during which one or more individually addressed frames are transmitted between two peer mesh STAs. Within a mesh peer service period, a mesh STA may obtain multiple TXOPs. A mesh peer service period is directional. One mesh STA is the owner of the mesh peer service period. It obtains TXOPs in order to transmit Data frames or Management frames to the recipient in the mesh peer service period. At the end of the frame transmissions, the owner of the mesh peer service period terminates the mesh peer service period. The other mesh STA operates as the recipient of the mesh peer service period and does not obtain TXOPs for transmitting Data frames or Management frames to the owner of the mesh peer service period. A mesh STA may have multiple mesh peer service periods concurrently toward multiple neighbor peer mesh STAs. At most, one mesh peer service period is set up in each direction with each peer mesh STA.

A mesh peer service period is initiated by a peer trigger frame. A peer trigger frame may initiate two mesh peer service periods. This enables both the transmitter and the receiver of the peer trigger frame to become the owner of a mesh peer service period. An example mesh peer service period between two mesh STAs in light or deep sleep mode is shown in Figure 11C-7. The numbering on the left-hand-side describes the phase of the operation: 1 indicates the Initiation phase, 2 indicates the data transmission phase, and 3 indicates the termination phase of the mesh peer service period.

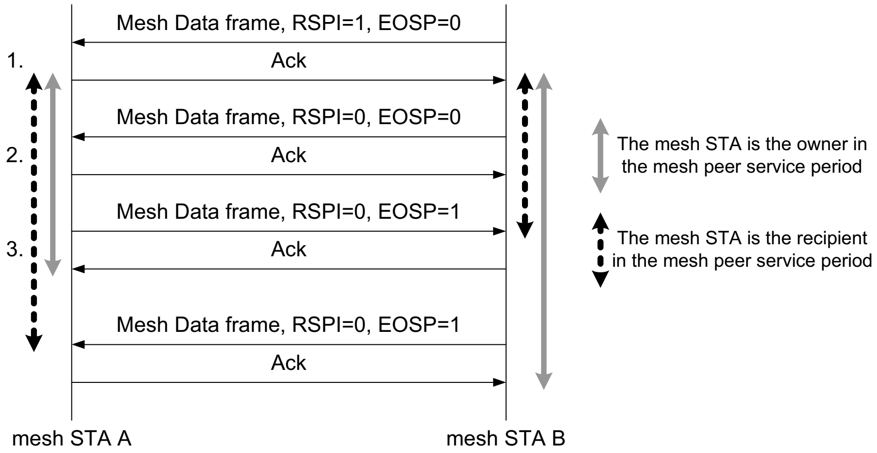

Mesh Data frame, RSPI=1, EOSP=0
1.
Ack
Mesh Data frame, RSPI=0, EOSP=0
2.
Ack
Mesh Data frame, RSPI=0, EOSP=1
3.
Ack
Mesh Data frame, RSPI=0, EOSP=1
Ack
mesh STA A
mesh STA B
The mesh STA is the owner in the mesh peer service period
The mesh STA is the recipient in the mesh peer service period

Figure 11C-7—Mesh peer service period

11C.13.9.2 Initiation of a mesh peer service period

A Mesh Data frame or a QoS Null frame that requires acknowledgement are used as a peer trigger frame. The RSPI and the EOSP subfields in the QoS Control field control the initiation of a mesh peer service period. Table 11C-32 lists how mesh peer service periods shall be initiated with different combinations of RSPI and EOSP field values.

Mesh peer service periods are not used in frame transmissions toward active mode mesh STAs.

Table 11C-32—Mesh peer service period triggering with RSPI and EOSP field combinations in peer trigger frame

RSPI	EOSP	Mesh peer service period triggering
0	0	One mesh peer service period is initiated. The transmitter of the trigger frame is the owner in the mesh peer service period.
0	1	No mesh peer service period is initiated.
1	0	Two mesh peer service periods are initiated. Both mesh STAs are owners in a mesh peer service period.
1	1	One mesh peer service period is initiated. The receiver of the trigger frame is the owner in the mesh peer service period.

The mesh peer service period may be initiated in the following cases:

- A mesh STA in light or deep sleep mode receives a peer trigger frame during its mesh awake window as described in 11C.13.6
- A mesh STA in active mode receives a peer trigger frame from the peer mesh STA in light or deep sleep mode as described in 11C.13.8.2
- A mesh STA receives a peer trigger frame from the peer mesh STA in light sleep mode as described in 11C.13.8.4

In addition, when a mesh STA uses MCCA with a neighbor peer mesh STA while in a light sleep mode for the corresponding mesh peering, a scheduled service period begins at the each MCCAOP start time as described in 11C.13.10. A mesh STA in a light or deep sleep mode shall enter the Awake state prior to the start time of scheduled service period.

11C.13.9.3 Operation during a mesh peer service period

During the mesh peer service period, the owner and the recipient of the mesh peer service period shall operate in Awake state. The mesh peer service period may contain one or more TXOPs.

Reverse Direction Grant (RDG) shall not be used when the receiver of the TXOP operates in light or deep sleep mode for the link and there is no mesh peer service period ongoing toward the TXOP holder.

11C.13.9.4 Termination of a mesh peer service period

The mesh peer service period is terminated after a successfully acknowledged QoS Null or Mesh Data frame with the EOSP subfield set to 1 from the owner of the mesh peer service period.

If the mesh STA does not receive an acknowledgement to a frame that requires an acknowledgement and that is sent with the EOSP subfield set to 1, the mesh STA shall retransmit that frame at least once within the same mesh peer service period—subject to applicable retry or lifetime limit. The maximum number of retransmissions within the same mesh peer service period is the lesser of the Max Retry Limit and the MIB attribute dot11MeshSTAMissingAckRetryLimit.

NOTE—If an Ack to the retransmission of this last frame in the same mesh peer service period is not received, the mesh STA might use the next mesh peer service period to further retransmit that frame subject to the applicable retry or lifetime limit.

11C.13.10 MCCA use by power saving mesh STA

When dot11MCCAActivated is true and the mesh STA establishes MCCAOPs, the mesh STA shall be in active mode or light sleep mode towards the neighbor peer mesh STAs with which it has established MCCAOPs.

A scheduled mesh peer service period begins at the MCCAOP start time, if the MCCAOP responder operates in light sleep mode for the MCCAOP owner. The MCCAOP owner is the owner of the scheduled mesh peer service period. The MCCAOP responder is the recipient of the scheduled mesh peer service period. Scheduled mesh peer service periods are not used if the MCCAOP responder is in active mode for the MCCAOP owner.

The scheduled mesh peer service period continues until it is successfully terminated by the acknowledged QoS Null or Mesh Data frame with the EOSP subfield set to 1 from the owner of the mesh peer service period to the recipient of the mesh peer service period as described in 11C.13.9.

Annex A

(normative)

Protocol Implementation Conformance Statement (PICS) proforma

A.2 Abbreviations and special symbols

A.2.2 General abbreviations for Item and Supported columns

Insert the following to the end of A.2.2:

MP	Mesh protocol capability
HWM	HWMP path selection protocol capability

A.4 PICS proforma—IEEE Std 802.11-2007

A.4.3 IUT configuration

Change the "*CF2" and "*CF12" row of the table in A.4.3 as follows:

Item	IUT configuration	References	Status	Support
*CF2	Independent station (~~not~~ neither an AP nor a mesh STA)	5.2	O.1	Yes □ No □
*CF12	Quality-of-service (QoS) supported	9.9, 9.10, 5.2.9, 5.2.14.3	O (CF16 or CF2a):M	Yes □ No □ N/A □

Insert the following rows after the "*CF2" row in the table in A.4.3:

Item	IUT configuration	References	Status	Support
*CF2a	Mesh station	5.2.14	O.1	Yes □ No □
CF2a.1	Operation in an MBSS	5.2.14	CF2a:M	Yes □ No □ N/A □

A.4.4 MAC protocol

A.4.4.1 MAC protocol capabilities

Insert the following row to end of table in A.4.4.1:

Item	Protocol capability	References	Status	Support
PC39	Simultaneous authentication of equals (SAE)	8.2a	CF2a:M	Yes □ No □

A.4.4.4 MAC addressing functions

Insert the following new rows to end of the table in A.4.4.4:

Item	Protocol capability	Reference	Status	Support
AD6	Group addressed Mesh Data frame addressing (3 address frame)	7.1.2, 7.1.3.1, 7.1.3.3, 9.22.3	CF2a:M	Yes □ No □ N/A □

Item	Protocol capability	Reference	Status	Support
AD7	Individually addressed Mesh Data frame addressing (4 address frame)	7.1.2, 7.1.3.1, 7.1.3.3, 9.22.3	CF2a:M	Yes □ No □ N/A □
AD8	Proxied group addressed Mesh Data frame addressing (4 address frame)	7.1.2, 7.1.3.1, 7.1.3.3, 7.1.3.6.3, 9.22.3	CF2a:M	Yes □ No □ N/A □
AD9	Proxied individually addressed Mesh Data frame addressing (6 address frame)	7.1.2, 7.1.3.1, 7.1.3.3, 7.1.3.6.3, 9.22.3	CF2a:M	Yes □ No □ N/A □
AD10	Multihop Action frame addressing (4 address frame)	7.1.2, 7.1.3.1, 7.1.3.3, 7.1.3.6.3, 7.4.16, 9.22.3	CF2a:M	Yes □ No □ N/A □
AD11	TA filtering for mesh STA	7.1.3.3, 7.2.2.1, 9.22.3	CF2a:M	Yes □ No □ N/A □

A.4.14 QoS base functionality

Change the "QB5" row of the table in A.4.14 as follows:

Item	Protocol capability	Reference	Status	Support
QB5	Automatic power-save delivery (APSD)	7.4.2, 11.2.1	(CF1 and CF12):O (CF2 and CF12):O	Yes □ No □ N/A □

A.4.15 QoS enhanced distributed channel access (EDCA)

Change the “QD7” row of the table in A.4.15 as follows:

Item	Protocol capability	Reference	Status	Support
QD7	Power management in an infrastructure BSS or in an IBSS	11.2	(CF1 and CF12):O (CF2 and CF12):O	Yes □ No □ N/A □

A.4.16 QoS hybrid coordination function (HCF) controlled channel access (HCCA)

Change the table in A.4.16 as follows:

Item	Protocol capability	Reference	Status	Support
QP1	Traffic specification (TSPEC) and associated frame formats	7.4.2	(CF1 and CF12):M (CF2 and CF12):M	Yes □ No □ N/A □
QP2	HCCA rules	9.1.3.2, 9.9.2, 9.9.2.1–9.9.2.3	(CF1 and CF12):M (CF2 and CF12):M	Yes □ No □ N/A □
QP3	HCCA schedule generation and management	9.9.3	(CF1 and CF12):M	Yes □ No □ N/A □
QP4	HCF frame exchange sequence	9.9.1, 9.3.2	(CF1 and CF12):M (CF2 and CF12):M	Yes □ No □ N/A □
QP5	Traffic stream (TS) management	11.4	(CF1 and CF12):M (CF2 and CF12):M	Yes □ No □ N/A □
QP6	Minimum TSPEC parameter set	9.9.3	(CF1 and CF12):M (CF2 and CF12):M	Yes □ No □ N/A □
QP7	Power management in an infrastructure BSS	11.2.1.4, 11.2.1.5, 11.2.1.6, 11.2.1.7, 11.2.1.8, 11.2.1.9, 11.2.1.10	(CF1 and CF12):M (CF2 and CF12):M	Yes □ No □ N/A □

Insert the following new subclause after the A.4.22:

A.4.23 Mesh protocol capabilities

A.4.23.1 General mesh support

Item	Protocol capability	Reference	Status	Support
*MP1	Support of mesh capability	5.2.14, 11C.1	CF2a:M	Yes □ No □ N/A □
MP1.1	Mesh BSS scanning	11C.2.2, 11C.2.6	MP1:M	Yes □ No □ N/A □

Item	Protocol capability	Reference	Status	Support
MP1.2	Candidate peer mesh STA determination	11C.2.7	MP1:M	Yes □ No □ N/A □
MP1.3	Active mesh profile determination	11C.2.3, 11C.2.4	MP1:M	Yes □ No □ N/A □
MP1.4	Establishing a mesh BSS	11C.2.8	MP1:M	Yes □ No □ N/A □
MP1.5	Becoming a member of a mesh BSS	11C.2.8	MP1:M	Yes □ No □ N/A □
MP1.6	Announcement of mesh profile and supplemental information for the mesh discovery	11C.2.3, 11C.2.5	MP1:M	Yes □ No □ N/A □
*MP2	Mesh peering management (MPM) framework	11C.3	CF2a:M	Yes □ No □ N/A □
*MP2.1	Mesh peering management (MPM) protocol	11C.3	MP2:M	Yes □ No □ N/A □
MP2.1.1	Processing of Mesh Peering Open frame	11C.3.6	MP2.1:M	Yes □ No □ N/A □
MP2.1.2	Processing of Mesh Peering Confirm frame	11C.3.7	MP2.1:M	Yes □ No □ N/A □
MP2.1.3	Processing of Mesh Peering Close frame	11C.3.8	MP2.1:M	Yes □ No □ N/A □
MP2.1.4	MPM finite state machine	11C.4	MP2.1:M	Yes □ No □ N/A □
*MP2.2	Authenticated mesh peering exchange (AMPE)	11C.5	MP2:O	Yes □ No □ N/A □
MP2.2.1	Mesh authentication using SAE	11C.3.3, 8.2a	MP2.2:M	Yes □ No □ N/A □
MP2.2.2	Mesh authentication using IEEE 802.1X	11C.3.3, 5.8	MP2.2:O	Yes □ No □ N/A □
MP2.2.3	Protected Mesh Peering Management frame processing	11C.5.3, 11C.5.5	MP2.2:M	Yes □ No □ N/A □
MP2.2.4	AMPE finite state machine	11C.5.6	MP2.2:M	Yes □ No □ N/A □
MP2.2.5	MGTK distribution	11C.5.4	MP2.2:M	Yes □ No □ N/A □
MP2.2.6	MGTK update	11C.6	MP2.2:O	Yes □ No □ N/A □
MP3	Mesh STA beaconing	11C.12.3	CF2a:M	Yes □ No □ N/A □
*MP4	Mesh STA synchronization	11C.12.2	CF2a:M	Yes □ No □ N/A □
*MP4.1	Neighbor offset synchronization method	11C.12.2.2	MP4:M	Yes □ No □ N/A □
MP4.1.1	Calculation of TSF offset	11C.12.2.2.2	MP4.1:M	Yes □ No □ N/A □
MP4.1.2	Clock drift adjustment	11C.12.2.2.3	MP4.1:M	Yes □ No □ N/A □
*MP4.2	Mesh beacon collision avoidance (MBCA)	11C.12.4	MP4:O	Yes □ No □ N/A □
MP4.2.1	Beacon timing advertisement	11C.12.4.2	MP4.2:M	Yes □ No □ N/A □

Item	Protocol capability	Reference	Status	Support
MP4.2.2	TBTT selection	11C.12.4.3	MP4.2:M	Yes ❑ No ❑ N/A ❑
MP4.2.3	TBTT adjustment	11C.12.4.4	MP4.2:M	Yes ❑ No ❑ N/A ❑
MP4.2.4	Frame transmission across reported TBTT	11C.12.4.5	MP4.2:O	Yes ❑ No ❑ N/A ❑
MP4.2.5	Delayed beacon transmission	11C.12.4.6	MP4.2:O	Yes ❑ No ❑ N/A ❑
*MP5	MCCA	9.9a.3	CF2a:O	Yes ❑ No ❑ N/A ❑
MP5.1	MCCAOP Advertisement	9.9a.3.7	MP5:M	Yes ❑ No ❑ N/A ❑
MP5.2	Neighbor MCCAOP Recognition	9.9a.3.4–9.9a.3.5	MP5:M	Yes ❑ No ❑ N/A ❑
MP5.3	MCCAOP Setup	9.9a.3.6	MP5:M	Yes ❑ No ❑ N/A ❑
MP5.4	Access during MCCAOPs	9.9a.3.9	MP5:M	Yes ❑ No ❑ N/A ❑
MP5.5	MCCAOP teardown	9.9a.3.8	MP5:M	Yes ❑ No ❑ N/A ❑
*MP6	Intra mesh congestion control	11C.11	CF2a:O	Yes ❑ No ❑ N/A ❑
MP6.1	Local congestion monitoring and detection	11C.11	MP6:M	Yes ❑ No ❑ N/A ❑
MP6.2	Congestion control signaling	11C.11	MP6:M	Yes ❑ No ❑ N/A ❑
MP6.3	Local rate control	11C.11	MP6:M	Yes ❑ No ❑ N/A ❑
*MP7	MBSS channel switch procedure	11.9.7, 11.9a.3	CF2a:M	Yes ❑ No ❑ N/A ❑
MP7.1	Transmission of channel switch advertisement	11.9.7, 11.9a.3	MP7:M	Yes ❑ No ❑ N/A ❑
MP7.2	Propagation of channel switch advertisement	11.9.7, 11.9a.3	MP7:M	Yes ❑ No ❑ N/A ❑
*MP8	Mesh power save operation (operation in light or deep sleep mode)	11C.13	CF2a:O	Yes ❑ No ❑ N/A ❑
MP8.1	Link-specific mesh power mode setting	11C.13.2.2, 11C.13.8	MP8:M	Yes ❑ No ❑ N/A ❑
MP8.2	Non-peer mesh power mode setting	11C.13.2.3	MP8:M	Yes ❑ No ❑ N/A ❑
MP8.3	Light sleep mode operation	11C.13.8.4	MP8:M	Yes ❑ No ❑ N/A ❑
MP8.4	Deep sleep mode operation	11C.13.8.5	MP8:M	Yes ❑ No ❑ N/A ❑
MP8.5	STA power state transitions	11C.13.3	MP8:M	Yes ❑ No ❑ N/A ❑
MP8.6	Mesh awake window operation	11C.13.6	MP8:M	Yes ❑ No ❑ N/A ❑
*MP9	Mesh power save support	11C.13	CF2a:M	Yes ❑ No ❑ N/A ❑
MP9.1	TIM transmission	11C.13.4	MP9:M	Yes ❑ No ❑ N/A ❑
MP9.2	Link-specific mesh power modes determination	11C.13.2	MP9:M	Yes ❑ No ❑ N/A ❑

Item	Protocol capability	Reference	Status	Support
MP9.3	Group addressed frame transmission	11C.13.7	MP9:M	Yes ❑ No ❑ N/A ❑
MP9.4	Frame transmission to a mesh STA in light sleep mode	11C.13.7, 11C.13.9	MP9:M	Yes ❑ No ❑ N/A ❑
MP9.5	Frame transmission to a mesh STA in deep sleep mode	11C.13.7, 11C.13.9	MP9:M	Yes ❑ No ❑ N/A ❑
MP10	Airtime link metric computation	11C.8	CF2a:M	Yes ❑ No ❑ N/A ❑
*MP11	Link metric reporting	11C.7.3	CF2a:M	Yes ❑ No ❑ N/A ❑
MP11.1	Autonomous link metric reporting	11C.7.3	MP11:O	Yes ❑ No ❑ N/A ❑
MP11.2	Link metric reporting upon request	11C.7.3	MP11:M	Yes ❑ No ❑ N/A ❑
*MP12	Proxy operation	11C.10.4	CF2a:O	Yes ❑ No ❑ N/A ❑
MP12.1	Data forwarding at proxy mesh gate	11C.10.3	MP12:M	Yes ❑ No ❑ N/A ❑
MP12.2	Maintenance of proxy information	11C.10.4.2	CF2a:M	Yes ❑ No ❑ N/A ❑
MP12.3	Proxy update using Proxy Update and Proxy Update Confirmation frames	11C.10.4	CF2a:M	Yes ❑ No ❑ N/A ❑
MP12.4	Proxy update using HWMP Mesh Path Selection frames	11C.9.9, 11C.9.10, 11C.9.11	HWM1:M	Yes ❑ No ❑ N/A ❑
*MP13	Gate announcement	11C.10.2	CF2a:O	Yes ❑ No ❑ N/A ❑
MP13.1	GANN transmission	11C.10.2	MP13:O	Yes ❑ No ❑ N/A ❑
MP13.2	GANN reception and propagation	11C.10.2	CF2a:M	Yes ❑ No ❑ N/A ❑
*MP14	Mesh Control field handling	7.1.3.6.3	CF2a:M	Yes ❑ No ❑ N/A ❑
MP14.1	Address Extension recognition	7.1.3.6.3, 9.22.3	MP14:M	Yes ❑ No ❑ N/A ❑
MP14.2	Mesh TTL handling	7.1.3.6.3, 9.22.4, 9.22.5, 9.22.6	MP14:M	Yes ❑ No ❑ N/A ❑
MP14.3	Mesh Sequence Number handling	7.1.3.6.3, 9.22.4, 9.22.5, 9.22.6, 9.22.7	MP14:M	Yes ❑ No ❑ N/A ❑
*MP15	MSDU/MMPDU forwarding	9.22	CF2a:O	Yes ❑ No ❑ N/A ❑
MP15.1	Individually addressed MSDU forwarding	9.22.4	MP15:M	Yes ❑ No ❑ N/A ❑
MP15.2	Group addressed MSDU forwarding	9.22.5	MP15:M	Yes ❑ No ❑ N/A ❑
MP15.3	MMPDU forwarding	9.22.6	MP15:M	Yes ❑ No ❑ N/A ❑
MP15.4	Detection of duplicate MSDUs/MMPDUs	9.22.7	CF2a:M	Yes ❑ No ❑ N/A ❑

Item	Protocol capability	Reference	Status	Support
MP15.5	Treatment of unknown destination	9.22.9	CF2a:M	Yes □ No □ N/A □

A.4.23.2 HWMP path selection protocol capabilities

Item	Protocol capability	Reference	Status	Support
*HWM1	Hybrid wireless mesh protocol (HWMP)	11C.9	CF2a:M	Yes □ No □ N/A □
*HWM1.1	On-demand path selection	11C.9.3	HWM1:M	Yes □ No □ N/A □
HWM1.1.1	PREQ processing for on-demand path selection	11C.9.9	HWM1.1:M	Yes □ No □ N/A □
HWM1.1.1	PREP processing for on-demand path selection	11C.9.10	HWM1.1:M	Yes □ No □ N/A □
HWM1.1.1	PERR processing for on-demand path selection	11C.9.11	HWM1.1:M	Yes □ No □ N/A □
*HWM1.2	Proactive tree building	11C.9.4	HWM1:M	Yes □ No □ N/A □
HWM1.2.1	PREQ processing for proactive tree building	11C.9.9	HWM1.2:M	Yes □ No □ N/A □
HWM1.2.2	PREP processing for proactive tree building	11C.9.10	HWM1.2:M	Yes □ No □ N/A □
HWM1.2.3	PERR processing for proactive tree building	11C.9.11	HWM1.2:M	Yes □ No □ N/A □
HWM1.2.4	RANN processing	11C.9.12	HWM1.2:M	Yes □ No □ N/A □
HWM2	Maintenance of forwarding information	9.22.2, 11C.9.8.4	MP15:M	Yes □ No □ N/A □

Annex D

(normative)

ASN.1 encoding of the MAC and PHY MIB

Change the "Station Management (SMT) Attributes" of "Major sections" in Annex D as follows:

```
-- **********************************************************************
-- * Major sections
-- **********************************************************************

-- Station ManagemenT (SMT) Attributes
   -- DEFINED AS "The SMT object class provides the necessary support
   -- at the station to manage the processes in the station such that
   -- the station may work cooperatively as a part of an IEEE 802.11
   -- network."

dot11smt OBJECT IDENTIFIER ::= { ieee802dot11 1 }

   -- dot11smt GROUPS
   -- dot11StationConfigTable                               ::= { dot11smt 1 }
   -- dot11AuthenticationAlgorithmsTable                    ::= { dot11smt 2 }
   -- dot11WEPDefaultKeysTable                              ::= { dot11smt 3 }
   -- dot11WEPKeyMappingsTable                              ::= { dot11smt 4 }
   -- dot11PrivacyTable                                     ::= { dot11smt 5 }
   -- dot11SMTnotification                                  ::= { dot11smt 6 }
   -- dot11MultiDomainCapabilityTable                       ::= { dot11smt 7 }
   -- dot11SpectrumManagementTable                          ::= { dot11smt 8 }
   -- dot11RSNAConfigTable                                  ::= { dot11smt 9 }
   -- dot11RSNAConfigPairwiseCiphersTable                   ::= { dot11smt 10 }
   -- dot11RSNAConfigAuthenticationSuitesTable              ::= { dot11smt 11 }
   -- dot11RSNAStatsTable                                   ::= { dot11smt 12 }
   -- dot11RegulatoryClassesTable                           ::= { dot11smt 13 }
   -- dot11RadioResourceMeasurement                         ::= { dot11smt 14 }
   -- dot11FastBSSTransitionConfigTable                     ::= { dot11smt 15 }
   -- dot11LCIDSETable                                      ::= { dot11smt 16 }
   -- dot11HTStationConfigTable                             ::= { dot11smt 17 }
   -- dot11WirelessMgmtOptionsTable                         ::= { dot11smt 18 }
   -- dot11LocationServicesNextIndex                        ::= { dot11smt 19 }
   -- dot11LocationServicesTable                            ::= { dot11smt 20 }
   -- dot11WirelessMGTEventTable                            ::= { dot11smt 21 }
   -- dot11WirelessNetworkManagement                        ::= { dot11smt 22 }
   -- dot11MeshSTAConfigTable                               ::= { dot11smt 23 }
   -- dot11MeshHWMPConfigTable                              ::= { dot11smt 24 }
   -- dot11RSNAConfigPasswordValueTable                     ::= { dot11smt 25 }
   -- dot11RSNAConfigDLCGroupTable                          ::= { dot11smt 26 }


-- **********************************************************************
-- * dot11StationConfig TABLE
-- **********************************************************************
```

Change the end of the "Dot11StationConfigEntry" of the "dot11StationConfigTable" as follows:

```
            dot11MSGCFActivated                                 TruthValue,
            dot11MeshActivated                                  TruthValue
                                                                       }
```

Insert "dot11MeshActivated" to the end of "dot11StationConfigTable" as follows:

```
dot11MeshActivated OBJECT-TYPE
       SYNTAX TruthValue
       MAX-ACCESS read-write
       STATUS current
       DESCRIPTION
       "This is a control variable.
       It is written by an external management entity.
       Changes take effect as soon as practical in the implementation.

       When this object is true, this indicates that the STA is a mesh STA.
       Configuration variables for mesh operation are found in the
       dot11MeshSTAConfigTable."
       ::= { dot11StationConfigEntry 136 }


-- **********************************************************************
-- * dot11AuthenticationAlgorithms TABLE
-- **********************************************************************
```

Change the "dot11AuthenticationAlgorithmsTable" as follows:

```
dot11AuthenticationAlgorithmsTable OBJECT-TYPE
       SYNTAX SEQUENCE of Dot11AuthenticationAlgorithmsEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
              "This (conceptual) table of attributes is a set of all the
              authentication algorithms supported by stations. The fol-
              lowing are the default values and the associated algorithm:
                 Value = 1: Open System
                 Value = 2: Shared Key
                 Value = 3: Fast BSS Transition (FT)
                 Value = 4: Simultaneous authentication of equals (SAE)"
       REFERENCE "IEEE Std 802.11-<year>, 7.3.1.1 (Authentication Algorithm
          Number field)"
       ::= { dot11smt 2 }

dot11AuthenticationAlgorithm OBJECT-TYPE
       SYNTAX INTEGER  {
              openSystem(1),
              sharedKey(2),
              fastBSSTransition(3),
              simultaneousAuthEquals(4) }
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
```

```
             "This is a control variable.
             It is written by an external management entity.
             Changes take effect as soon as practical in the implementation.

             This attribute is the authentication algorithm described by this
             entry in the table. The following values can be used here
                Value = 1: Open System
                Value = 2: Shared Key
                Value = 3: Fast BSS Transition (FT)
                Value = 4: Simultaneous authentication of equals (SAE)"
       ::= { dot11AuthenticationAlgorithmsEntry 2 }


-- ********************************************************************
-- * dot11RSNAConfig TABLE (RSNA and TSN)
-- ********************************************************************
```

Change the "Dot11RSNAConfigEntry" as follows:

```
Dot11RSNAConfigEntry ::=
    SEQUENCE {
       dot11RSNAConfigVersion                                   Unsigned32,
       dot11RSNAConfigPairwiseKeysImplemented                   Unsigned32,
       dot11RSNAConfigGroupCipher                               OCTET STRING,
       dot11RSNAConfigGroupRekeyMethod                          INTEGER,
       dot11RSNAConfigGroupRekeyTime                            Unsigned32,
       dot11RSNAConfigGroupRekeyPackets                         Unsigned32,
       dot11RSNAConfigGroupRekeyStrict                          TruthValue,
       dot11RSNAConfigPSKValue                                  OCTET STRING,
       dot11RSNAConfigPSKPassPhrase                             DisplayString,
       dot11RSNAConfigGroupUpdateCount                          Unsigned32,
       dot11RSNAConfigPairwiseUpdateCount                       Unsigned32,
       dot11RSNAConfigGroupCipherSize                           Unsigned32,
       dot11RSNAConfigPMKLifetime                               Unsigned32,
       dot11RSNAConfigPMKReauthThreshold                        Unsigned32,
       dot11RSNAConfigNumberOfPTKSAReplayCountersImplemented    Unsigned32,
       dot11RSNAConfigSATimeout                                 Unsigned32,
       dot11RSNAAuthenticationSuiteSelected                     OCTET STRING,
       dot11RSNAPairwiseCipherSelected                          OCTET STRING,
       dot11RSNAGroupCipherSelected                             OCTET STRING,
       dot11RSNAPMKIDUsed                                       OCTET STRING,
       dot11RSNAAuthenticationSuiteRequested                    OCTET STRING,
       dot11RSNAPairwiseCipherRequested                         OCTET STRING,
       dot11RSNAGroupCipherRequested                            OCTET STRING,
       dot11RSNATKIPCounterMeasuresInvoked                      Unsigned32,
       dot11RSNA4WayHandshakeFailures                           Unsigned32,
       dot11RSNAConfigNumberOfGTKSAReplayCountersImplemented    Unsigned32,
       dot11RSNAConfigSTKKeysImplemented                        Unsigned32,
       dot11RSNAConfigSTKCipher                                 OCTET STRING,
       dot11RSNAConfigSTKRekeyTime                              Unsigned32,
       dot11RSNAConfigSMKUpdateCount                            Unsigned32,
       dot11RSNAConfigSTKCipherSize                             Unsigned32,
       dot11RSNAConfigSMKLifetime                               Unsigned32,
       dot11RSNAConfigSMKReauthThreshold                        Unsigned32,
       dot11RSNAConfigNumberOfSTKSAReplayCountersImplemented    Unsigned32,
       dot11RSNAPairwiseSTKSelected                             OCTET STRING,
       dot11RSNASMKHandshakeFailures                            Unsigned32,
       dot11RSNASAERetransPeriod                                Unsigned32,
       dot11RSNASAEAntiCloggingThreshold                        Unsigned32,
       dot11RSNASAESync                                         Unsigned32 }
```

Change the "dot11RSNAConfigNumberOfPTKSAReplayCountersImplemented" as follows:

```
dot11RSNAConfigNumberOfPTKSAReplayCountersImplemented OBJECT-TYPE
          SYNTAX Unsigned32
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
              "This is a capability variable.
              Its value is determined by device capabilities.

              Specifies the number of PTKSA replay counters per association:
                 0 -> 1 replay counters,
                 1 -> 2 replay counters,
                 2 -> 4 replay counters,
                 3 -> 16 replay counters"
          DEFVAL { 3 }
      ::= { dot11RSNAConfigEntry 18 }
```

Change the "dot11RSNAConfigNumberOfGTKSAReplayCountersImplemented" as follows:

```
dot11RSNAConfigNumberOfGTKSAReplayCountersImplemented OBJECT-TYPE
          SYNTAX Unsigned32
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
              "This is a capability variable.
              Its value is determined by device capabilities.

              Specifies the number of GTKSA replay counters per association:
                 0 -> 1 replay counter,
                 1 -> 2 replay counters,
                 2 -> 4 replay counters,
                 3 -> 16 replay counters"
          DEFVAL { 3 }
      ::= { dot11RSNAConfigEntry 29 }
```

Change the "dot11RSNAConfigNumberOfSTKSAReplayCountersImplemented" as follows:

```
dot11RSNAConfigNumberOfSTKSAReplayCountersImplemented OBJECT-TYPE
          SYNTAX Unsigned32
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
              "This is a capability variable.
              Its value is determined by device capabilities.

              Specifies the number of STKSA replay counters per association:
                 0 -> 1 replay counter,
                 1 -> 2 replay counters,
                 2 -> 4 replay counters,
                 3 -> 16 replay counters"
          DEFVAL { 3 }
      ::= { dot11RSNAConfigEntry 37 }
```

Insert the following three components to the end of the "dot11RSNAConfig TABLE":

```
dot11RSNASAERetransPeriod OBJECT-TYPE
          SYNTAX Unsigned32
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
             "This is a control variable.
             It is written by the SME when establishing or becoming a member of
             a BSS.
             Changes take effect for the next MLME-START.request.

             This object specifies the initial retry timeout, in millisecond
             units, used by the SAE authentication and key establishment
             protocol."
          DEFVAL { 40 }
      ::= { dot11RSNAConfigEntry 40 }

dot11RSNASAEAntiCloggingThreshold OBJECT-TYPE
          SYNTAX Unsigned32
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
             "This is a capability variable.
             Its value is determined by device capabilities.

             This object specifies the maximum number of SAE protocol instances
             allowed to simultaneously be in either Commit or Confirmed state."
          DEFVAL { 5 }
      ::= { dot11RSNAConfigEntry 41 }

dot11RSNASAESync OBJECT-TYPE
          SYNTAX Unsigned32
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
             "This is a capability variable.
             Its value is determined by device capabilities.

             This object specifies the maximum number of synchronization errors
             that are allowed to happen prior to disassociation of the offending
             SAE peer."
          DEFVAL { 5 }
      ::= { dot11RSNAConfigEntry 42 }
```

Insert the following "dot11MeshSTAConfig TABLE," "dot11MeshHWMPConfig TABLE," "dot11RSNAConfigPasswordValue TABLE," and "dot11RSNAConfigDLCGroup TABLE" after "dot11HTStationConfig TABLE" and prior to "MAC Attribute Templates":

```
-- **********************************************************************
-- * dot11MeshSTAConfig TABLE
-- **********************************************************************
```

```
dot11MeshSTAConfigTable OBJECT-TYPE
          SYNTAX SEQUENCE OF Dot11MeshSTAConfigEntry
          MAX-ACCESS not-accessible
          STATUS current
          DESCRIPTION
              "Mesh Station Configuration attributes. In tabular form to allow
              for multiple instances on an agent."
      ::= { dot11smt 23 }

dot11MeshSTAConfigEntry  OBJECT-TYPE
          SYNTAX Dot11MeshSTAConfigEntry
          MAX-ACCESS not-accessible
          STATUS current
          DESCRIPTION
              "An entry in the dot11MeshStationConfigTable. It is possible for
              there to be multiple IEEE 802.11 interfaces on one agent, each with
              its unique MAC address. The relationship between an IEEE 802.11
              interface and an interface in the context of the Internet-standard
              MIB is one-to-one. As such, the value of an ifIndex object instance
              can be directly used to identify corresponding instances of the
              objects defined herein.
              ifIndex - Each IEEE 802.11 interface is represented by an ifEntry.
              Interface tables in this MIB module are indexed by ifIndex."
          INDEX { ifIndex }
      ::= { dot11MeshSTAConfigTable 1 }

Dot11MeshSTAConfigEntry ::=
       SEQUENCE {
             dot11MeshID                                    OCTET STRING,
             dot11MeshNumberOfPeerings                      Unsigned32,
             dot11MeshAcceptingAdditionalPeerings           TruthValue,
             dot11MeshConnectedToMeshGate                   TruthValue,
             dot11MeshSecurityActivated                     TruthValue,
             dot11MeshActiveAuthenticationProtocol          INTEGER,
             dot11MeshMaxRetries                            Unsigned32,
             dot11MeshRetryTimeout                          Unsigned32,
             dot11MeshConfirmTimeout                        Unsigned32,
             dot11MeshHoldingTimeout                        Unsigned32,
             dot11MeshConfigGroupUpdateCount                Unsigned32,
             dot11MeshActivePathSelectionProtocol           INTEGER,
             dot11MeshActivePathSelectionMetric             INTEGER,
             dot11MeshForwarding                            TruthValue,
             dot11MeshTTL                                   Unsigned32,
             dot11MeshGateAnnouncementProtocol              TruthValue,
             dot11MeshGateAnnouncementInterval              Unsigned32,
             dot11MeshActiveCongestionControlMode           INTEGER,
             dot11MeshActiveSynchronizationMethod           INTEGER,
             dot11MeshNbrOffsetMaxNeighbor                  Unsigned32,
             dot11MBCAActivated                             TruthValue,
             dot11MeshBeaconTimingReportInterval            Unsigned32,
             dot11MeshBeaconTimingReportMaxNum              Unsigned32,
             dot11MeshDelayedBeaconTxInterval               Unsigned32,
             dot11MeshDelayedBeaconTxMaxDelay               Unsigned32,
             dot11MeshDelayedBeaconTxMinDelay               Unsigned32,
             dot11MeshAverageBeaconFrameDuration            Unsigned32,
             dot11MeshSTAMissingAckRetryLimit               Unsigned32,
             dot11MeshAwakeWindowDuration                   Unsigned32,
             dot11MCCAImplemented                           TruthValue,
             dot11MCCAActivated                             TruthValue,
             dot11MAFlimit                                  Unsigned32,
             dot11MCCAScanDuration                          Unsigned32,
             dot11MCCAAdvertPeriodMax                       Unsigned32,
             dot11MCCAMinTrackStates                        Unsigned32,
             dot11MCCAMaxTrackStates                        Unsigned32,
             dot11MCCAOPtimeout                             Unsigned32,
             dot11MCCACWmin                                 Unsigned32,
             dot11MCCACWmax                                 Unsigned32,
             dot11MCCAAIFSN                                 Unsigned32
                                                                   }


dot11MeshID OBJECT-TYPE
          SYNTAX OCTET STRING (SIZE(0..32))
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
             "This is a control variable.
             It is written by an external management entity.
             Changes take effect for the next MLME-START.request.

             This attribute reflects the Mesh ID configured in this entity."
        ::= { dot11MeshSTAConfigEntry 1 }

dot11MeshNumberOfPeerings OBJECT-TYPE
          SYNTAX Unsigned32 (0..255)
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
             "This is a control variable.
             It is written by an external management entity.
             Changes take effect as soon as practical in the implementation.

             This attribute indicates the number of mesh peering currently
             maintained by the STA. This value is reflected in the Number of
             Peerings subfield in the Mesh Formation Info field in the Mesh
             Configuration element."
        ::= { dot11MeshSTAConfigEntry 2 }

dot11MeshAcceptingAdditionalPeerings OBJECT-TYPE
          SYNTAX TruthValue
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
             "This is a control variable.
             It is written by an external management entity.
             Changes take effect as soon as practical in the implementation.

             This attribute indicates whether or not the station is willing to
             accept additional peerings. This value is reflected in the
             Accepting Additional Mesh Peerings subfield in the Mesh Capability
             field in the Mesh Configuration element."
        ::= { dot11MeshSTAConfigEntry 3 }

dot11MeshConnectedToMeshGate OBJECT-TYPE
          SYNTAX TruthValue
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
             "This is a control variable.
             It is written by an external management entity.
             Changes take effect as soon as practical in the implementation.

             This attribute indicates whether or not the station has a mesh path
             to a mesh gate. This value is reflected in the Connected to Mesh
             Gate subfield in the Mesh Formation Info field in the Mesh
             Configuration element."
        ::= { dot11MeshSTAConfigEntry 4 }

dot11MeshSecurityActivated OBJECT-TYPE
          SYNTAX TruthValue
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
             "This is a control variable.
             It is written by an external management entity.
             Changes take effect for the next MLME-START.request.

             This attribute specifies whether or not the station is security
             enabled."
        ::= { dot11MeshSTAConfigEntry 5 }

dot11MeshActiveAuthenticationProtocol OBJECT-TYPE
          SYNTAX INTEGER {
              null (0),
              sae (1),
              ieee8021x (2),
              vendorSpecific (255) }
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
             "This is a control variable.
             It is written by an external management entity.
             Changes take effect for the next MLME-START.request.

             This attribute specifies the active authentication protocol."
          DEFVAL { null }
        ::= { dot11MeshSTAConfigEntry 6 }

dot11MeshMaxRetries OBJECT-TYPE
          SYNTAX Unsigned32 (0..16)
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
             "This is a control variable.
             It is written by an external management entity.
             Changes take effect as soon as practical in the implementation.

             This attribute specifies the maximum number of Mesh Peering Open
             retries that can be sent to establish a new mesh peering instance
             in a mesh BSS."
          DEFVAL { 2 }
        ::= { dot11MeshSTAConfigEntry 7 }

dot11MeshRetryTimeout OBJECT-TYPE
          SYNTAX Unsigned32 (1..255)
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
             "This is a control variable.
             It is written by an external management entity.
             Changes take effect as soon as practical in the implementation.

             This attribute specifies the initial retry timeout, in millisecond
             units, used by the Mesh Peering Open message."
          DEFVAL { 40 }
        ::= { dot11MeshSTAConfigEntry 8 }

dot11MeshConfirmTimeout OBJECT-TYPE
          SYNTAX Unsigned32 (1..255)
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
             "This is a control variable.
             It is written by an external management entity.
             Changes take effect as soon as practical in the implementation.

             This attribute specifies the initial retry timeout, in millisecond
             units, used by the Mesh Peering Open message."
          DEFVAL { 40 }
        ::= { dot11MeshSTAConfigEntry 9 }

dot11MeshHoldingTimeout OBJECT-TYPE
          SYNTAX Unsigned32 (1..255)
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
             "This is a control variable.
             It is written by an external management entity.
             Changes take effect as soon as practical in the implementation.

             This attribute specifies the confirm timeout, in millisecond units,
             used by the mesh peering management to close a mesh peering."
          DEFVAL { 40 }
        ::= { dot11MeshSTAConfigEntry 10 }

dot11MeshConfigGroupUpdateCount OBJECT-TYPE
          SYNTAX Unsigned32 (1..4294967295)
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
             "This is a control variable.
             It is written by an external management entity.
             Changes take effect as soon as practical in the implementation.

             This attribute specifies how many times the Mesh Group Key Inform
             frame will be retried per mesh group key handshake attempt."
          DEFVAL { 3 }
        ::= { dot11MeshSTAConfigEntry 11 }
```

```
dot11MeshActivePathSelectionProtocol OBJECT-TYPE
          SYNTAX INTEGER { hwmp (1), vendorSpecific (255) }
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
             "This is a control variable.
             It is written by an external management entity.
             Changes take effect for the next MLME-START.request.

             This attribute specifies the active path selection protocol."
          DEFVAL { hwmp }
        ::= { dot11MeshSTAConfigEntry 12 }

dot11MeshActivePathSelectionMetric OBJECT-TYPE
          SYNTAX INTEGER { airtimeLinkMetric (1), vendorSpecific (255) }
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
             "This is a control variable.
             It is written by an external management entity.
             Changes take effect for the next MLME-START.request.

             This attribute specifies the active path selection metric."
          DEFVAL { airtimeLinkMetric }
        ::= { dot11MeshSTAConfigEntry 13 }

dot11MeshForwarding OBJECT-TYPE
          SYNTAX TruthValue
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
             "This is a control variable.
             It is written by an external management entity.
             Changes take effect as soon as practical in the implementation.

             This attribute specifies the ability of a mesh STA to forward
             MSDUs."
          DEFVAL { true }
        ::= { dot11MeshSTAConfigEntry 14 }

dot11MeshTTL OBJECT-TYPE
          SYNTAX Unsigned32 (0..255)
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
             "This is a control variable.
             It is written by an external management entity.
             Changes take effect as soon as practical in the implementation.

             This attribute specifies the value of Mesh TTL subfield set at a
             source mesh STA."
          DEFVAL { 31 }
        ::= { dot11MeshSTAConfigEntry 15 }

dot11MeshGateAnnouncementProtocol OBJECT-TYPE
          SYNTAX TruthValue
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
```

```
            "This is a control variable.
            It is written by an external management entity.
            Changes take effect as soon as practical in the implementation.

            This attribute specifies whether or not the mesh STA, which is
            collocated with a mesh gate, is using the gate announcement
            protocol."
         DEFVAL { false }
       ::= { dot11MeshSTAConfigEntry 16 }

dot11MeshGateAnnouncementInterval OBJECT-TYPE
         SYNTAX Unsigned32 (1..65535)
         MAX-ACCESS read-write
         STATUS current
         DESCRIPTION
            "This is a control variable.
            It is written by an external management entity.
            Changes take effect as soon as practical in the implementation.

            This attribute specifies the gate announcement interval. The gate
            announcement interval is the number of seconds between the
            transmission of two gate announcements."
         DEFVAL { 10 }
       ::= { dot11MeshSTAConfigEntry 17 }

dot11MeshActiveCongestionControlMode OBJECT-TYPE
         SYNTAX INTEGER {
             null (0),
             congestionControlSignaling (1),
             vendorSpecific (255) }
         MAX-ACCESS read-write
         STATUS current
         DESCRIPTION
            "This is a control variable.
            It is written by an external management entity.
            Changes take effect for the next MLME-START.request.

            This attribute specifies the active congestion control protocol."
         DEFVAL { null }
       ::= { dot11MeshSTAConfigEntry 18 }

dot11MeshActiveSynchronizationMethod OBJECT-TYPE
         SYNTAX INTEGER {
             neighborOffsetSynchronization (1),
             vendorSpecific (255) }
         MAX-ACCESS read-write
         STATUS current
         DESCRIPTION
            "This is a control variable.
            It is written by an external management entity.
            Changes take effect for the next MLME-START.request.

            This attribute specifies the active synchronization method."
         DEFVAL { neighborOffsetSynchronization }
       ::= { dot11MeshSTAConfigEntry 19 }

dot11MeshNbrOffsetMaxNeighbor OBJECT-TYPE
         SYNTAX Unsigned32 (1..255)
         MAX-ACCESS read-write
```

```
          STATUS current
          DESCRIPTION
             "This is a capability variable.
             Its value is determined by device capabilities.

             This attribute specifies the maximum number of neighbor STAs with
             which the mesh STA maintains synchronization using the neighbor
             offset synchronization method."
          DEFVAL { 16 }
        ::= { dot11MeshSTAConfigEntry 20 }

dot11MBCAActivated OBJECT-TYPE
          SYNTAX TruthValue
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
             "This is a control variable.
             It is written by an external management entity.
             Changes take effect as soon as practical in the implementation.

             This attribute specifies whether or not the station activates mesh
             beacon collision avoidance mechanisms."
          DEFVAL { false }
        ::= { dot11MeshSTAConfigEntry 21 }

dot11MeshBeaconTimingReportInterval OBJECT-TYPE
          SYNTAX Unsigned32 (1..255)
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
             "This is a control variable.
             It is written by an external management entity.

             This attribute specifies when the Beacon Timing element is present
             in Beacon frames. The Beacon Timing element is present when the
             DTIM Count value in the Beacon frame is zero or equal to an integer
             multiple of the set value."
          DEFVAL { 4 }
        ::= { dot11MeshSTAConfigEntry 22 }

dot11MeshBeaconTimingReportMaxNum OBJECT-TYPE
          SYNTAX Unsigned32 (0..50)
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
             "This is a control variable.
             It is written by an external management entity.
             Changes take effect as soon as practical in the implementation.

             This attribute specifies the maximum number of the Beacon Timing
             Information field contained in a Beacon Timing element in the
             transmitting Beacon frames."
          DEFVAL { 16 }
        ::= { dot11MeshSTAConfigEntry 23 }

dot11MeshDelayedBeaconTxInterval OBJECT-TYPE
          SYNTAX Unsigned32 (0..255)
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
             "This is a control variable.
             It is written by an external management entity.
             Changes take effect as soon as practical in the implementation.

             This attribute specifies the interval of the delayed beacon
             transmission for the purpose of MBCA. The value is expressed in
             units of Beacon Interval. The value 0 indicates that the delayed
             beacon transmission is disabled."
          DEFVAL { 0 }
        ::= { dot11MeshSTAConfigEntry 24 }

dot11MeshDelayedBeaconTxMaxDelay OBJECT-TYPE
          SYNTAX Unsigned32 (0..65535)
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
             "This is a control variable.
             It is written by an external management entity.
             Changes take effect as soon as practical in the implementation.

             This attribute specifies the maximum delay time from a TBTT of
             delayed beacon transmissions for the purpose of MBCA. The value is
             expressed in units of microseconds."
          DEFVAL { 2048 }
        ::= { dot11MeshSTAConfigEntry 25 }

dot11MeshDelayedBeaconTxMinDelay OBJECT-TYPE
          SYNTAX Unsigned32 (0..4023)
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
             "This is a control variable.
             It is written by an external management entity.
             Changes take effect as soon as practical in the implementation.

             This attribute specifies the minimum delay time from a TBTT of
             delayed beacon transmissions for the purpose of MBCA. The value is
             expressed in units of microseconds."
          DEFVAL { 0 }
        ::= { dot11MeshSTAConfigEntry 26 }

dot11MeshAverageBeaconFrameDuration OBJECT-TYPE
          SYNTAX Unsigned32 (0..16383)
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
             "This is a control variable.
             It is written by an external management entity.
             Changes take effect as soon as practical in the implementation.

             This attribute specifies the average duration of the last 16 Beacon
             frames of other mesh STAs received by this mesh STA. The value is
             expressed in units of microseconds."
        ::= { dot11MeshSTAConfigEntry 27 }

dot11MeshSTAMissingAckRetryLimit OBJECT-TYPE
          SYNTAX Unsigned32 (0..100)
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
             "This is a control variable.
             It is written by an external management entity.
             Changes take effect as soon as practical in the implementation.

             This attribute specifies the number of times the mesh STA may retry
             a frame for which it does not receive an ACK for a STA in power-save
             mode after the mesh STA does not receive an ACK to a directed MPDU
             sent with the EOSP set to 1."
        ::= { dot11MeshSTAConfigEntry 28 }

dot11MeshAwakeWindowDuration OBJECT-TYPE
          SYNTAX Unsigned32 (0..65535)
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
             "This is a control variable.
             It is written by an external management entity.
             Changes take effect as soon as practical in the implementation.

             This attribute specifies the duration of the mesh Awake Window in
             TUs. This value is reflected in the value of the Mesh Awake Window
             element."
        ::= { dot11MeshSTAConfigEntry 29 }

dot11MCCAImplemented OBJECT-TYPE
          SYNTAX TruthValue
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
             "This is a capability variable.
             Its value is determined by device capabilities.

             This attribute specifies whether or not the MCCA is implemented in
             this station."
        ::= { dot11MeshSTAConfigEntry 30 }

dot11MCCAActivated OBJECT-TYPE
          SYNTAX TruthValue
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
             "This is a control variable.
             It is written by an external management entity.
             Changes take effect as soon as practical in the implementation.

             This attribute specifies whether or not the station is MCCA
             enabled."
          DEFVAL { false }
        ::= { dot11MeshSTAConfigEntry 31 }

dot11MAFlimit OBJECT-TYPE
          SYNTAX Unsigned32 (0..255)
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
             "This is a control variable.
             It is written by an external management entity.
```

```
              Changes take effect as soon as practical in the implementation.

              This attribute specifies the maximum MCCA access fraction allowed
              at the mesh STA. This number expresses a multiple of (1/255)."
           DEFVAL { 128 }
        ::= { dot11MeshSTAConfigEntry 32 }

dot11MCCAScanDuration OBJECT-TYPE
           SYNTAX Unsigned32 (1..65535)
           MAX-ACCESS read-write
           STATUS current
           DESCRIPTION
              "This is a control variable.
              It is written by an external management entity.
              Changes take effect as soon as practical in the implementation.

              This attribute specifies the duration in TUs after the activation
              of MCCA that the mesh STA shall not initiate or accept MCCAOP Setup
              Requests."
           DEFVAL { 3200 } -- 2^5 * 100
        ::= { dot11MeshSTAConfigEntry 33 }

dot11MCCAAdvertPeriodMax OBJECT-TYPE
           SYNTAX Unsigned32 (0..255)
           MAX-ACCESS read-write
           STATUS current
           DESCRIPTION
              "This is a control variable.
              It is written by an external management entity.
              Changes take effect as soon as practical in the implementation.

              This attribute specifies the maximum interval that a mesh STA with
              dot11MCCAActivated equal to true waits for an MCCAOP advertisement.
              It is expressed in number of DTIM intervals."
           DEFVAL { 1 }
        ::= { dot11MeshSTAConfigEntry 34 }

dot11MCCAMinTrackStates OBJECT-TYPE
           SYNTAX Unsigned32 (83..65535)
           MAX-ACCESS read-write
           STATUS current
           DESCRIPTION
              "This is a capability variable.
              It is written by an external management entity.
              Changes take effect as soon as practical in the implementation.

              This attribute specifies the smallest number of MCCAOP reservations
              that the MAC entity is able to track."
           DEFVAL { 83 }
        ::= { dot11MeshSTAConfigEntry 35 }

dot11MCCAMaxTrackStates OBJECT-TYPE
           SYNTAX Unsigned32 (83..65535)
           MAX-ACCESS read-write
           STATUS current
           DESCRIPTION
              "This is a control variable.
              It is written by an external management entity.
              Changes take effect as soon as practical in the implementation.
```

```
             The lower bound is given by the current value of
             dot11MCCAMinTrackStates.

             This attribute specifies the maximum number of MCCAOP reservations
             that the MAC entity is able to track."
          DEFVAL { 83 }
        ::= { dot11MeshSTAConfigEntry 36 }

dot11MCCAOPtimeout OBJECT-TYPE
          SYNTAX Unsigned32
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
             "This is a control variable.
             It is written by an external management entity.
             Changes take effect as soon as practical in the implementation.

             This attribute specifies the timeout value for an MCCAOP teardown.
             It is expressed in TU."
          DEFVAL { 10000 }
        ::= { dot11MeshSTAConfigEntry 37 }

dot11MCCACWmin OBJECT-TYPE
          SYNTAX Unsigned32 (0..65535)
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
             "This is a control variable.
             It is written by an external management entity.
             Changes take effect as soon as practical in the implementation.

             This attribute specifies the value of the minimum size of the
             window that shall be used by the mesh STA during the MCCAOP for
             which it is the MCCAOP owner for generating a random number for the
             backoff."
          DEFVAL { 0 }
        ::= { dot11MeshSTAConfigEntry 38 }

dot11MCCACWmax OBJECT-TYPE
          SYNTAX Unsigned32 (0..65535)
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
             "This is a control variable.
             It is written by an external management entity.
             Changes take effect as soon as practical in the implementation.

             This attribute specifies the value of the maximum size of the
             window that shall be used by the mesh STA during the MCCAOP for
             which it is the MCCAOP owner for generating a random number for the
             backoff. The value of this attribute shall be such that it could
             always be expressed in the form of 2X - 1, where X is an integer."
          DEFVAL { 31 }
        ::= { dot11MeshSTAConfigEntry 39 }

dot11MCCAAIFSN OBJECT-TYPE
          SYNTAX Unsigned32 (0..15)
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
             "This is a control variable.
             It is written by an external management entity.
             Changes take effect as soon as practical in the implementation.

             This attribute specifies the number of slots, after a SIFS
             duration, that the mesh STA shall sense the medium idle either
             before transmitting or executing a backoff during an MCCAOP for
             which it is the MCCAOP owner."
          DEFVAL { 1 }
        ::= { dot11MeshSTAConfigEntry 40 }

-- **********************************************************************
-- * End of dot11MeshSTAConfig TABLE
-- **********************************************************************

-- **********************************************************************
-- * dot11MeshHWMPConfig TABLE
-- **********************************************************************

dot11MeshHWMPConfigTable OBJECT-TYPE
          SYNTAX SEQUENCE OF Dot11MeshHWMPConfigEntry
          MAX-ACCESS not-accessible
          STATUS current
          DESCRIPTION
             "Mesh Station HWMP Configuration attributes. In tabular form to
             allow for multiple instances on an agent."
      ::= { dot11smt 24 }

dot11MeshHWMPConfigEntry  OBJECT-TYPE
          SYNTAX Dot11MeshHWMPConfigEntry
          MAX-ACCESS not-accessible
          STATUS current
          DESCRIPTION
             "An entry in the dot11MeshHWMPConfigTable. It is possible for there
             to be multiple IEEE 802.11 interfaces on one agent, each with its
             unique MAC address. The relationship between an IEEE 802.11
             interface and an interface in the context of the Internet-standard
             MIB is one-to-one. As such, the value of an ifIndex object instance
             can be directly used to identify corresponding instances of the
             objects defined herein.
             ifIndex - Each IEEE 802.11 interface is represented by an ifEntry.
             Interface tables in this MIB module are indexed by ifIndex."
          INDEX { ifIndex }
      ::= { dot11MeshHWMPConfigTable 1 }

Dot11MeshHWMPConfigEntry ::=
       SEQUENCE {
             dot11MeshHWMPmaxPREQretries                      Unsigned32,
             dot11MeshHWMPnetDiameter                         Unsigned32,
             dot11MeshHWMPnetDiameterTraversalTime            Unsigned32,
             dot11MeshHWMPpreqMinInterval                     Unsigned32,
             dot11MeshHWMPperrMinInterval                     Unsigned32,
             dot11MeshHWMPactivePathToRootTimeout             Unsigned32,
             dot11MeshHWMPactivePathTimeout                   Unsigned32,
             dot11MeshHWMProotMode                            INTEGER,
             dot11MeshHWMProotInterval                      Unsigned32,
             dot11MeshHWMPrannInterval                      Unsigned32,
             dot11MeshHWMPtargetOnly                        INTEGER,
             dot11MeshHWMPmaintenanceInterval               Unsigned32,
             dot11MeshHWMPconfirmationInterval              Unsigned32
                                                                          }

dot11MeshHWMPmaxPREQretries OBJECT-TYPE
          SYNTAX Unsigned32 (0..255)
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
             "This is a control variable.
             It is written by an external management entity.
             Changes take effect as soon as practical in the implementation.

             This attribute specifies the number of Action frames containing a
             PREQ that an originator mesh STA can send to a particular path
             target for a specific path discovery."
          DEFVAL { 3 }
       ::= { dot11MeshHWMPConfigEntry 1}

dot11MeshHWMPnetDiameter OBJECT-TYPE
          SYNTAX Unsigned32 (1..255)
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
             "This is a control variable.
             It is written by an external management entity.
             Changes take effect as soon as practical in the implementation.

             This attribute specifies the estimate of the maximum number of hops
             that it takes for an HWMP element to propagate across the mesh
             BSS."
          DEFVAL { 31 }
       ::= { dot11MeshHWMPConfigEntry 2}

dot11MeshHWMPnetDiameterTraversalTime OBJECT-TYPE
          SYNTAX Unsigned32 (1..65535)
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
             "This is a control variable.
             It is written by an external management entity.
             Changes take effect as soon as practical in the implementation.

             This attribute specifies the estimate of the interval of time (in
             TUs) that it takes for an HWMP element to propagate across the mesh
             BSS."
          DEFVAL { 500 }
       ::= { dot11MeshHWMPConfigEntry 3}

dot11MeshHWMPpreqMinInterval OBJECT-TYPE
          SYNTAX Unsigned32 (1..65535)
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
             "This is a control variable.
```

```
            It is written by an external management entity.
            Changes take effect as soon as practical in the implementation.

            This attribute specifies the minimum interval of time (in TUs)
            during which a mesh STA can send only one Action frame containing a
            PREQ element."
         DEFVAL { 100 }
      ::= { dot11MeshHWMPConfigEntry 4}

dot11MeshHWMPperrMinInterval OBJECT-TYPE
         SYNTAX Unsigned32 (1..65535)
         MAX-ACCESS read-write
         STATUS current
         DESCRIPTION
            "This is a control variable.
            It is written by an external management entity.
            Changes take effect as soon as practical in the implementation.

            This attribute specifies the minimum interval of time (in TUs)
            during which a mesh STA can send only one Action frame containing a
            PERR element."
         DEFVAL { 100 }
      ::= { dot11MeshHWMPConfigEntry 5}

dot11MeshHWMPactivePathToRootTimeout OBJECT-TYPE
         SYNTAX Unsigned32 (1..65535)
         MAX-ACCESS read-write
         STATUS current
         DESCRIPTION
            "This is a control variable.
            It is written by an external management entity.
            Changes take effect as soon as practical in the implementation.

            This object shall specify the time (in TUs) for which mesh STAs
            receiving a proactive PREQ shall consider the forwarding
            information to the root mesh STA to be valid; it needs to be greater
            than dot11MeshHWMProotInterval."
         DEFVAL { 5000 }
      ::= { dot11MeshHWMPConfigEntry 6}

dot11MeshHWMPactivePathTimeout OBJECT-TYPE
         SYNTAX Unsigned32 (1..65535)
         MAX-ACCESS read-write
         STATUS current
         DESCRIPTION
            "This is a control variable.
            It is written by an external management entity.
            Changes take effect as soon as practical in the implementation.

            This attribute specifies the time (in TUs) for which mesh STAs
            receiving a PREQ to individual target(s) shall consider the
            forwarding information to be valid."
         DEFVAL { 5000 }
      ::= { dot11MeshHWMPConfigEntry 7}

dot11MeshHWMProotMode OBJECT-TYPE
         SYNTAX INTEGER {
             noRoot(0),
             proactivePREQnoPREP(2),
                 proactivePREQwithPREP(3),
                 rann(4) }
            MAX-ACCESS read-write
            STATUS current
            DESCRIPTION
               "This is a control variable.
               It is written by an external management entity.
               Changes take effect as soon as practical in the implementation.

               This attribute controls the configuration of a mesh STA as root
               mesh STA. A mesh STA is configured as a root mesh STA if
               dot11MeshHWMProotMode is set to 2, 3 or 4. Different values
               correspond to different modes of the root mesh STA. The mesh STA is
               not a root mesh STA when the attribute is set to 0."
            DEFVAL { noRoot }
        ::= { dot11MeshHWMPConfigEntry 8}

dot11MeshHWMProotInterval OBJECT-TYPE
            SYNTAX Unsigned32 (1..65535)
            MAX-ACCESS read-write
            STATUS current
            DESCRIPTION
               "This is a control variable.
               It is written by an external management entity.
               Changes take effect as soon as practical in the implementation.

               This attribute specifies the minimum interval of time (in TUs)
               during which a root mesh STA can send only one Action frame
               containing a proactive PREQ element."
            DEFVAL { 2000 }
        ::= { dot11MeshHWMPConfigEntry 9}

dot11MeshHWMPrannInterval OBJECT-TYPE
            SYNTAX Unsigned32 (1..65535)
            MAX-ACCESS read-write
            STATUS current
            DESCRIPTION
               "This is a control variable.
               It is written by an external management entity.
               Changes take effect as soon as practical in the implementation.

               This attribute specifies the minimum interval of time (in TUs)
               during which a mesh STA can send only one Action frame containing a
               RANN element."
            DEFVAL { 2000 }
        ::= { dot11MeshHWMPConfigEntry 10}

dot11MeshHWMPtargetOnly OBJECT-TYPE
            SYNTAX INTEGER { intermediateMSTA(0), targetOnly(1) }
            MAX-ACCESS read-write
            STATUS current
            DESCRIPTION
               "This is a control variable.
               It is written by an external management entity.
               Changes take effect as soon as practical in the implementation.

               This attribute, when set to intermediateMSTA (0), allows
               intermediate mesh STAs to respond with a PREP to a PREQ if they have
               valid forwarding information to the requested target. When set to
```

```
             targetOnly (1), only the target mesh STA is allowed to respond with
             a PREP to a PREQ."
          DEFVAL { targetOnly }
      ::= { dot11MeshHWMPConfigEntry 11}

dot11MeshHWMPmaintenanceInterval OBJECT-TYPE
          SYNTAX Unsigned32 (1..65535)
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
             "This is a control variable.
             It is written by an external management entity.
             Changes take effect as soon as practical in the implementation.

             This attribute specifies the minimum interval of time (in TUs)
             during which a mesh STA can send only one Action frame containing a
             PREQ element for path maintenance."
          DEFVAL { 2000 }
      ::= { dot11MeshHWMPConfigEntry 12}

dot11MeshHWMPconfirmationInterval OBJECT-TYPE
          SYNTAX Unsigned32 (1..65535)
          MAX-ACCESS read-write
          STATUS current
          DESCRIPTION
             "This is a control variable.
             It is written by an external management entity.
             Changes take effect as soon as practical in the implementation.

             This attribute specifies the minimum interval of time (in TUs)
             during which a mesh STA can send only one Action frame containing a
             PREQ element for root path confirmation."
          DEFVAL { 2000 }
      ::= { dot11MeshHWMPConfigEntry 13}

-- ********************************************************************
-- * End of dot11MeshHWMPConfig TABLE
-- ********************************************************************

-- *******************************************************************
-- * dot11RSNAConfigPasswordValue TABLE
-- *******************************************************************

dot11RSNAConfigPasswordValueTable OBJECT-TYPE
      SYNTAX SEQUENCE OF Dot11RSNAConfigPasswordValueEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
          "When SAE authentication is the selected AKM suite,
          this table is used to locate the binary representation
          of a shared, secret, and potentially low-entropy word,
          phrase, code, or key that will be used as the
          authentication credential between a TA/RA pair.

          This table is logically write-only. Reading this table
          returns unsuccessful status or null or zero."
      ::= { dot11smt 25 }

dot11RSNAConfigPasswordValueEntry OBJECT-TYPE
```

```
      SYNTAX Dot11RSNAConfigPasswordValueEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
          "An entry (conceptual row) in the Password Value Table"
      INDEX { dot11RSNAConfigPasswordValueIndex }
    ::= { dot11RSNAConfigPasswordValueTable 1 }

Dot11RSNAConfigPasswordValueEntry ::=
      SEQUENCE {
          dot11RSNAConfigPasswordValueIndex           Unsigned32,
          dot11RSNAConfigPasswordCredential           OCTET STRING,
          dot11RSNAConfigPasswordPeerMac              MacAddress }

dot11RSNAConfigPasswordValueIndex OBJECT-TYPE
      SYNTAX Unsigned32
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
          "The auxiliary variable used to identify instances of the columnar
          objects in the Password Value table."
    ::= { dot11RSNAConfigPasswordValueEntry 1 }

dot11RSNAConfigPasswordCredential OBJECT-TYPE
      SYNTAX OCTET STRING
      MAX-ACCESS read-write
      STATUS current
      DESCRIPTION
          "This is a control variable.
          It is written by an external management entity.
          Changes take effect as soon as practical in the implementation.

          This variable is a binary representation of a shared,
          secret, and potentially low-entropy word, phrase, code
          or key used as an authentication credential.

          Any character-based word or phrase shall be converted
          into a canonical binary representation according to
          8.2a.3 before populating the Password Credential."
    ::= { dot11RSNAConfigPasswordValueEntry 2 }

dot11RSNAConfigPasswordPeerMac OBJECT-TYPE
      SYNTAX MacAddress
      MAX-ACCESS read-write
      STATUS current
      DESCRIPTION
          "This is a control variable.
          It is written by an external management entity.
          Changes take effect as soon as practical in the implementation.

          This variable represents the MAC address of the peer
          that is to be authenticated. A wildcard BSSID is
          permitted when passwords are shared among peers."
    ::= { dot11RSNAConfigPasswordValueEntry 3 }

-- ******************************************************************
-- * End of dot11RSNAConfigPasswordValue TABLE
-- ******************************************************************
-- ********************************************************************
-- * dot11RSNAConfigDLCGroup TABLE
-- ********************************************************************

dot11RSNAConfigDLCGroupTable OBJECT-TYPE
      SYNTAX SEQUENCE OF Dot11RSNAConfigDLCGroupEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
           "This table gives a prioritized list of domain parameter set
             Identifiers for discrete logarithm cryptography (DLC) groups."
     ::= { dot11smt 26 }

dot11RSNAConfigDLCGroupEntry OBJECT-TYPE
      SYNTAX Dot11RSNAConfigDLCGroupEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
           "An entry (conceptual row) in the DLC Group Table."
      INDEX { dot11RSNAConfigDLCGroupIndex }
    ::= { dot11RSNAConfigDLCGroupTable 1 }

Dot11RSNAConfigDLCGroupEntry ::=
      SEQUENCE {
           dot11RSNAConfigDLCGroupIndex               Unsigned32,
           dot11RSNAConfigDLCGroupIdentifier          Unsigned32 }

dot11RSNAConfigDLCGroupIndex OBJECT-TYPE
      SYNTAX Unsigned32
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
           "The variable used to identify instances of the columnar
           objects in the DLC Group Table. Entries are sorted
           based on the Group Index according to the priority
           of the Group Identifier relative to other objects.

           More preferred Group Identifiers will have a lower
           index in the Group Entry."
   ::= { dot11RSNAConfigDLCGroupEntry 1 }

dot11RSNAConfigDLCGroupIdentifier OBJECT-TYPE
      SYNTAX Unsigned32
      MAX-ACCESS read-write
      STATUS current
      DESCRIPTION
           "This is a control variable.
           It is written by an external management entity.
           Changes take effect as soon as practical in the implementation.

           This variable uniquely identifies a domain parameter
           set for a group in the IANA registry `Group Description'
           attributes for RFC 2409 (IKE)."
   ::= { dot11RSNAConfigDLCGroupEntry 2 }

-- *********************************************************************
-- *    End of dot11RSNAConfigDLCGroup TABLE
-- *********************************************************************
-- **********************************************************************
-- * Compliance Statements
-- **********************************************************************
```

Change dot11Compliance MODULE-COMPLIANCE in the "Compliance Statements" as follows:

```
dot11Compliance MODULE-COMPLIANCE
      STATUS  current
      DESCRIPTION
      "The compliance statement for SNMPv2 entities that implement the
      IEEE 802.11 MIB."
      MODULE -- this module
      MANDATORY-GROUPS {
      dot11SMTbase~~11~~12,
      dot11MACbase3,
      dot11CountersGroup3,
      dot11SmtAuthenticationAlgorithms,
      dot11ResourceTypeID,
      dot11PhyOperationComplianceGroup2 }
```

Insert the following "dot11MeshCompliance" to the end of the "Compliance Statements":

```
-- ********************************************************************
-- * Compliance Statements - Mesh
-- ********************************************************************

dot11MeshCompliance MODULE-COMPLIANCE
      STATUS current
      DESCRIPTION
           "The compliance statement for SNMPv2 entities that implement the IEEE
           802.11 MIB for Mesh."
      MODULE -- this module
      MANDATORY-GROUPS {
           dot11MeshComplianceGroup,
           dot11MeshHWMPComplianceGroup,
           dot11PasswordAuthComplianceGroup }
--    OPTIONAL-GROUPS { dot11MeshOptionGroup }
      ::= { dot11Compliances 4 }


-- **********************************************************************
-- * Groups - units of conformance
-- **********************************************************************
```

Change the end of the "dot11SMTbase11" of the "Groups - units of conformance" as follows:

```
           dot11WirelessNetworkManagementImplemented }
      STATUS ~~current~~deprecated
      DESCRIPTION
           "Superseded by dot11SMTbase12.
           The SMTbase11 object class provides the necessary support at the STA to
           manage the processes in the STA so that the STA may work cooperatively
           as a part of an IEEE 802.11 network, when the STA is capable of
           multidomain operation. This object group should be implemented when
           the multidomain capability option is implemented."
           ::= { dot11Groups 53 }
```

Insert the following "dot11SMTbase12," "dot11MeshComplianceGroup," "dot11MeshOptionGroup," "dot11MeshHWMPComplianceGroup," and "dot11PasswordAuthComplianceGroup" to the end of the "Groups - units of conformance":

```
dot11SMTbase12 OBJECT-GROUP
      OBJECTS {
           dot11MediumOccupancyLimit,
           dot11CFPollable,
           dot11CFPPeriod,
           dot11CFPMaxDuration,
           dot11AuthenticationResponseTimeOut,
           dot11PrivacyOptionImplemented,
           dot11PowerManagementMode,
           dot11DesiredSSID,
           dot11DesiredBSSType,
           dot11OperationalRateSet,
           dot11BeaconPeriod,
           dot11DTIMPeriod,
           dot11AssociationResponseTimeOut,
           dot11DisassociateReason,
           dot11DisassociateStation,
           dot11DeauthenticateReason,
           dot11DeauthenticateStation,
           dot11AuthenticateFailStatus,
           dot11AuthenticateFailStation,
           dot11MultiDomainCapabilityImplemented,
           dot11MultiDomainCapabilityEnabled,
           dot11CountryString,
           dot11SpectrumManagementImplemented,
           dot11SpectrumManagementRequired ,
           dot11RSNAOptionImplemented,
           dot11RegulatoryClassesImplemented,
           dot11RegulatoryClassesRequired,
           dot11QosOptionImplemented,
           dot11ImmediateBlockAckOptionImplemented,
           dot11DelayedBlockAckOptionImplemented,
           dot11DirectOptionImplemented,
           dot11APSDOptionImplemented,
           dot11QAckOptionImplemented,
           dot11QBSSLoadOptionImplemented,
           dot11QueueRequestOptionImplemented,
           dot11TXOPRequestOptionImplemented,
           dot11MoreDataAckOptionImplemented,
           dot11AssociateinNQBSS,
           dot11DLSAllowedInQBSS,
           dot11DLSAllowed,
           dot11AssociateStation,
           dot11AssociateID,
           dot11AssociateFailStation,
           dot11AssociateFailStatus,
           dot11ReassociateStation,
           dot11ReassociateID,
           dot11ReassociateFailStation,
           dot11ReassociateFailStatus,
           dot11RadioMeasurementCapable,
           dot11RadioMeasurementEnabled,
           dot11RRMMeasurementProbeDelay,
           dot11RRMMeasurementPilotPeriod,
           dot11RRMLinkMeasurementEnabled,
           dot11RRMNeighborReportEnabled,
           dot11RRMParallelMeasurementsEnabled,
           dot11RRMRepeatedMeasurementsEnabled,
           dot11RRMBeaconPassiveMeasurementEnabled,
           dot11RRMBeaconActiveMeasurementEnabled,
           dot11RRMBeaconTableMeasurementEnabled,
           dot11RRMBeaconMeasurementReportingConditionsEnabled,
           dot11RRMFrameMeasurementEnabled,
           dot11RRMChannelLoadMeasurementEnabled,
           dot11RRMNoiseHistogramMeasurementEnabled,
           dot11RRMStatisticsMeasaurementEnabled,
           dot11RRMLCIMeasurementEnabled,
           dot11RRMLCIAzimuthEnabled,
           dot11RRMTransmitStreamCategoryMeasurementEnabled,
           dot11RRMTriggeredTransmitStreamCategoryMeasurementEnabled,
           dot11RRMAPChannelReportEnabled,
           dot11RRMMIBEnabled,
           dot11RRMMaxMeasurementDuration,
           dot11RRMNonOperatingChannelMaxMeasurementDuration,
           dot11RRMMeasurementPilotTransmissionInformationEnabled,
           dot11RRMMeasurementPilotCapability,
           dot11RRMNeighborReportTSFOffsetEnabled,
           dot11RRMRCPIMeasurementEnabled,
           dot11RRMRSNIMeasurementEnabled,
           dot11RRMBSSAverageAccessDelayEnabled,
           dot11RRMBSSAvailableAdmissionCapacityEnabled,
           dot11FastBSSTransitionImplemented,
           dot11LCIDSEImplemented,
           dot11LCIDSERequired,
           dot11DSERequired,
           dot11ExtendedChannelSwitchEnabled,
           dot11HighThroughputOptionImplemented,
           dot11WirelessManagementImplemented,
           dot11MeshActivated,
           dot11RSNAPBACRequired,
           dot11PSMPOptionImplemented }
      STATUS current
      DESCRIPTION
           "The SMTbase12 object class provides the necessary support at the STA
           to manage the processes in the STA such that the STA may work
           cooperatively as a part of an IEEE 802.11 network."
      ::= { dot11Groups 57 }


 dot11MeshComplianceGroup OBJECT-GROUP
      OBJECTS {
           -- dot11MeshSTAConfigTable
           dot11MeshID,
           dot11MeshNumberOfPeerings,
           dot11MeshAcceptingAdditionalPeerings,
           dot11MeshConnectedToMeshGate,
           dot11MeshSecurityActivated,
           dot11MeshActiveAuthenticationProtocol,
           dot11MeshMaxRetries,
           dot11MeshRetryTimeout,
           dot11MeshConfirmTimeout,
           dot11MeshHoldingTimeout,
           dot11MeshActivePathSelectionProtocol,
           dot11MeshActivePathSelectionMetric,
           dot11MeshForwarding,
           dot11MeshTTL,
           dot11MeshGateAnnouncementProtocol,
           dot11MeshActiveCongestionControlMode,
           dot11MeshActiveSynchronizationMethod,
           dot11MeshNbrOffsetMaxNeighbor,
           dot11MBCAActivated,
           dot11MCCAImplemented,
           dot11MCCAActivated }
     STATUS current
     DESCRIPTION
           "This object class provides the objects from the IEEE 802.11 MIB
           required to manage mandatory mesh functionality. Note that additional
           objects for managing mesh functionality are located in the
           dot11MeshOptionGroup, dot11MeshHWMPComplianceGroup, and
           dot11PasswordAuthComplianceGroup."
     ::= { dot11Groups 56}


 dot11MeshOptionGroup OBJECT-GROUP
     OBJECTS {
           -- dot11MeshSTAConfigTable
           dot11MeshConfigGroupUpdateCount,
           dot11MeshGateAnnouncementInterval,
           dot11MeshBeaconTimingReportInterval,
           dot11MeshBeaconTimingReportMaxNum,
           dot11MeshDelayedBeaconTxInterval,
           dot11MeshDelayedBeaconTxMaxDelay,
           dot11MeshDelayedBeaconTxMinDelay,
           dot11MeshAverageBeaconFrameDuration,
           dot11MeshSTAMissingAckRetryLimit,
           dot11MeshAwakeWindowDuration,
           dot11MAFlimit,
           dot11MCCAScanDuration,
           dot11MCCAAdvertPeriodMax,
           dot11MCCAMinTrackStates,
           dot11MCCAMaxTrackStates,
           dot11MCCAOPtimeout,
           dot11MCCACWmin,
           dot11MCCACWmax,
           dot11MCCAAIFSN
           }
     STATUS current
     DESCRIPTION
           "This object class provides the objects from the IEEE 802.11 MIB
           required to manage optional mesh functionality. Note that other objects
           for managing mesh functionality are located in the
           dot11MeshComplianceGroup, dot11MeshHWMPComplianceGroup, and
           dot11PasswordAuthComplianceGroup."
     ::= { dot11Groups 60 }


 dot11MeshHWMPComplianceGroup OBJECT-GROUP
     OBJECTS {
           -- dot11MeshHWMPConfigTable
           dot11MeshHWMPmaxPREQretries,
           dot11MeshHWMPnetDiameter,
           dot11MeshHWMPnetDiameterTraversalTime,
           dot11MeshHWMPpreqMinInterval,
           dot11MeshHWMPperrMinInterval,
           dot11MeshHWMPactivePathToRootTimeout,
           dot11MeshHWMPactivePathTimeout,
           dot11MeshHWMProotMode,
           dot11MeshHWMProotInterval,
           dot11MeshHWMPrannInterval,
           dot11MeshHWMPtargetOnly,
           dot11MeshHWMPmaintenanceInterval,
           dot11MeshHWMPconfirmationInterval }
     STATUS current
     DESCRIPTION
           "This object class provides the objects from the IEEE 802.11 MIB
           required to manage HWMP path selection functionality. Note that other
           objects for managing mesh functionality are located in the
           dot11MeshComplianceGroup, dot11MeshOptionGroup, and
           dot11PasswordAuthComplianceGroup."
     ::= { dot11Groups 61 }


 dot11PasswordAuthComplianceGroup OBJECT-GROUP
     OBJECTS {
           -- dot11RSNAConfigTable
           dot11RSNASAERetransPeriod,
           dot11RSNASAEAntiCloggingThreshold,
           dot11RSNASAESync,
           -- dot11RSNAConfigPasswordValueTable
 --        dot11RSNAConfigPasswordValueIndex,
           dot11RSNAConfigPasswordCredential,
           dot11RSNAConfigPasswordPeerMac,
           -- dot11RSNAConfigDLCGroupTable
 --        dot11RSNAConfigDLCGroupIndex,
           dot11RSNAConfigDLCGroupIdentifier }
     STATUS current
     DESCRIPTION
           "This object class provides the objects from the IEEE 802.11 MIB
           required to manage password authentication. Note that other objects for
           managing mesh functionality are located in the
           dot11MeshComplianceGroup, dot11MeshOptionGroup, and
           dot11MeshHWMPComplianceGroup."
     ::= { dot11Groups 62 }
```

Annex H

(informative)

RSNA reference implementation and test vectors

Insert the following new subclause after H.9 (Management Frame Protection test vectors).

H.10 SAE test vector

```
group: 19
Password:  'thisisreallysecret'
Local MAC address: 7b:88:56:20:2d:8d
Peer's MAC address: e2:47:1c:0a:5a:cb
-------------------------------------------------------------------

H(e2:47:1c:0a:5a:cb | 7b:88:56:20:2d:8d, thisisreallysecret | 1)
69f69099 83675392 d0a3a882 47ffef20 413ee972 15872942 4415e139 46ecc206

candidate x value:
a16729e0 339c38f8 b06e2b83 76d43066 85578354 ab09d848 a0f140ac 825e6a3d

no solution to the equation of the elliptic curve with counter = 1

H(e2:47:1c:0a:5a:cb | 7b:88:56:20:2d:8d, thisisreallysecret | 2)
ab4b22f1 0e7cdbb2 9d1fd2de 5823198c 3c66733f 40e3f94a da06ee05 c83dac37

candidate x value:
103a5b96 8873bab0 fafc6dd8 ff3476ff 56487e7f 072b38e4 c9705497 1ce72b7b

PWE (x,y):
103a5b96 8873bab0 fafc6dd8 ff3476ff 56487e7f 072b38e4 c9705497 1ce72b7b
31742d39 f380f247 624218e9 45543004 d39973a5 68c5b904 b0cf5d36 2d44f3bf

local private value:
c5d7019e 7612d5f4 3cf91fe5 62b40bb8 b2640c65 c577b9b1 9994bf50 6baf2859

local mask value:
19d030fe 5bb11ee4 c27c9dfc 3c06520f 8fbe9290 059b0cc5 50db0d2b 9d3ac452

local commit:
dfa7329c d1c3f4d8 ff75bde1 9eba5dc8 42229ef5 cb12c676 ea6fcc7c 08e9ecab
3008b40e 01912bc5 3b862cd9 43305e86 46ee3b3e 6f745c5b b3ae8dfc 2ebf654e
d0a4e2a2 8bb98b62 9a4b0084 9df93d22 2999d086 5c9cceed a8e90fcb 53af5ae6

peer's commit:
10c1e1f1 d008713b 41986cdd 441eb991 bc823b60 118a5fc9 f51b16aa 00342147
19475f6f 50dbc87f 1505c109 e421a7e3 6b3a2e3f 48bfe52e 01b75f2b e7e5f4bc
948fe44c 741bd97f 51654857 7c6f320d 0c349939 857e0c79 30916d6f 323739d6

k:
```

```
6761b6a9 a9421297 f54a97ee c0daf188 49e582b4 bea7f06a cc34686f bd7900b1

keyseed::
ea72c928 2e364654 5faaa2c3 fb5791d4 74885907 7f3c7b04 208aceab 2ca9dd45

KCK:
cb7636d9 9b0dad17 2bd6a3fd 40bb76f4 4ecbb874 750396bc 74fba0ea 3a11f86c

local confirm:
01004664 47ab0962 ae780bcc 7a0ac672 a39c62ec 3009cfb2 34dd1918 37c792b9
548e

peer's confirm:
01002df5 f62c4610 5b606d76 72b89c3e 615421d2 6d9991da a8183778 811d30ac
e3db

PMK:
f6ecb8ad e3ae30df e35d31ea ee047161 b3a00d94 45c5dfd2 2cd8d8fb af83d9c7
```

Annex P

(informative)

Bibliography

P.1 General

Insert the following entries in P.1, renumbering as necessary.

[B52] IANA Protocol Assignments for the Internet Key Exchange (IKE) Attributes, http://www.iana.org/assignments/ipsec-registry

Annex S

(informative)

Frame exchange sequences

S.2 Basic sequences

Change the second paragraph of S.2 as follows:

(* This rule defines all the allowable frame exchange sequences *)
frame-sequence =
([**CTS**] (**Management** +broadcast | **Data** +group)) |
([**CTS** | **RTS CTS** | **PS-Poll**] {frag-frame **Ack**} last-frame **Ack**) |
(**PS-Poll Ack**) |
([**Beacon** +DTIM] {cf-sequence} [**CF-End** [+CF-Ack]])|
hcf-sequence |
mcf-sequence;

S.3 EDCA and HCCA sequences

Insert the following to the end of S.3:

(* An mcf-sequence represents all the sequences that may be generated under MCF. The sequence may be initiated by a mesh STA using EDCA channel access or MCCA channel access. *)
mcf-sequence =
([**CTS**] |{(**Data** +group +QoS) | **Management** +broadcast) } | ([**CTS**] 1{txop-sequence}) |
group-mccaop-abandon;

(* A group-mccaop-abandon is the delivery of a single QoS Null frame by a mesh STA that has dot11MCCAActivated true. *)
group-mccaop-abandon =
Data + broadcast + null + QoS

Annex X

(informative)

Interworking with external networks

X.4 Interworking with external networks and emergency call support

X.4.4 Access to emergency services in an RSN

Change the X.4.4 as follows:

If an infrastructure BSS ~~a network~~ requires authentication and encryption with RSN, a non-AP STA placing an emergency call associates and authenticate to the network by using an emergency services association (see 7.3.2.92). If the non-AP STA has user credentials that allow it to use a particular network, the non-AP STA can use its credentials to authenticate to the SSPN through the IEEE 802.11 infrastructure.

When a mesh STA has an emergency services association, and it receives a Mesh Peering Open frame that includes the Interworking element, with ASRA bit equal to 1 and UESA bit equal to 0, and the Authenticated Mesh Peering Exchange element (see 7.3.2.118) it allows access to emergency services. If the mesh STA has user credentials that allow the accessing mesh STA to use the mesh network, the mesh STA can use its credentials to authenticate with the accessing mesh peer.

In order to ~~To~~ use an emergency services association in an infrastructure BSS, a STA lacking security credentials can associate to a BSS in which emergency services are accessible by including an Interworking Element with the UESA field ~~set~~ equal to 1 in a (Re)-association Request frame. An AP receiving this type of (re)assocation request recognizes this as a request for unauthenticated emergency access. The AP can look up the VLAN ID to use with a AAA Server, or it can have an emergency services VLAN configured. The STA can either have, policies configured locally for quality-of-service parameters and network access restrictions, or it can look them up through external policy servers. When a mesh STA has an emergency services association, and it receives a Mesh Peering Open frame, from a mesh STA lacking security credentials, that includes the Interworking element, with ASRA bit equal to 1 and UESA bit equal to 1, and the Mesh Peering Management element (see 7.3.2.102) it allows access to emergency services.

When an emergency services association is used, ~~the IEEE 802.11 infrastructure~~ an infrastructure BSS or an MBSS should be designed to restrict access to ~~emergency call users~~ emergency services only (or alternatively prioritize the emergency services to the highest level of access). Methods of such restriction are beyond the scope of this standard, but can include an isolated VLAN for emergency services, filtering rules in the AP or network entity (e.g., router) in an external network to limit network access to only network elements involved in emergency calls, and per-session bandwidth control to place an upper limit on resource utilization.

Insert the following new annex after Annex X:

Annex Y

(informative)

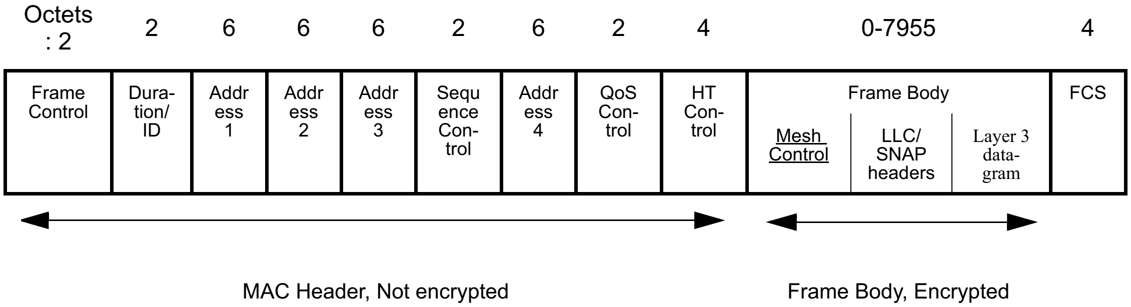
Mesh BSS operation

Y.1 Clarification of Mesh Data frame format

The Mesh Data frame consists of MAC Header, Frame Body, and the FCS. The fields in the MAC Header are described in 7.1.2.

In a Mesh Data frame, the Mesh Control field is placed as the first element in the encrypted Frame Body. The Frame Body contains also the LLC/SNAP headers and the higher layer data. The format of the Mesh Data frame is shown in Figure Y-1.

When the Mesh Data frame is fragmented, only the first fragment contains the Mesh Control field.



Octets : 2	2	6	6	6	2	6	2	4	0-7955			4
Frame Control	Dura-tion/ ID	Addr ess 1	Addr ess 2	Addr ess 3	Sequ ence Con-trol	Addr ess 4	QoS Con-trol	HT Con-trol	Frame Body			FCS
									Mesh Control	LLC/ SNAP headers	Layer 3 data-gram	

Figure Y-1—Format of the Mesh Data frame

Y.2 Operational considerations for interworking

Y.2.1 Formation and maintenance of the IEEE 802.1D spanning tree

No special action is required to support formation of the IEEE 802.1D spanning tree. Spanning tree control messages are typically delivered to bridges in group addressed frames. These messages are data frames from the point of view of the mesh BSS.

Y.3 Power save parameters selection

Y.3.1 General

Power save mechanisms enable mesh STA operation in Doze state. In Doze state a mesh STA may set the transceivers off. The power save mechanism may adjust the mesh power mode of a mesh STA for various reasons that are beyond the scope of the standard, including but not limited to traffic load or scanning and network maintenance needs.

This annex specifies recommendations on how a mesh STA selects mesh power modes based on traffic load, recommendations on how to do scanning in mesh BSSs that may contain mesh STAs in light or deep sleep mode, and recommended default values for power save related parameters.

Y.3.2 Selecting the mesh power mode based on traffic load

A mesh STA may adjust its mesh power mode based on the traffic load or the QoS requirements of the forwarded traffic. If a mesh STA has high traffic load or if a mesh STA is congested, the mesh STA should operate in active mode on the corresponding mesh peering to reduce delays or overhead that may be created by the operation in light or deep sleep mode.

A mesh STA may consider staying in active mode for all peer mesh STAs, even if only a single link is congested. When mesh STA operates in active mode, the mesh STA may receive frames at any time and mesh peer service periods may be triggered on demand, as in case of an AP that receives a trigger frame at times controlled by a non-AP STA.

If traffic load is moderate or low and best effort data is transmitted, a mesh STA may consider staying in light sleep mode for all or some mesh peerings. The mesh STA in light sleep mode for a mesh peering receives beacons from the peer mesh STA with periodic indication of buffered traffic.

If traffic load is low or no traffic is transmitted, a mesh STA may consider staying in deep sleep mode for a mesh peering. The mesh STA does not need to wake up to receive a beacon from the peer mesh STA to which it is in deep sleep mode.

The mesh STA may use deep sleep mode to control the number of times it enters the Awake state to receive Beacon frame from a peer mesh STA. If the mesh STA is in deep sleep mode for all of its mesh peering, the mesh STA needs only to remain in Awake state during its own beacon transmission and mesh awake window.

The mesh STA that forwards real time traffic (AC3 or AC2 with EDCA) should be in active mode for the corresponding link. A mesh STA forwarding real time traffic with EDCA may stay in light or deep sleep mode for the corresponding link, if the mesh STA is capable of initiating mesh peer service periods frequently and the other forwarding peer mesh STAs are in active mode. Poor handling of the mesh power modes may result to delays and inappropriate QoS of the forwarded MSDUs.

Y.3.3 Scanning of mesh BSSs

A mesh BSS may have mesh STAs that alternate Awake state and Doze state. With active scanning only devices in Awake state at the transmission time of Probe Request frame may be found. However, with passive scanning also mesh STAs in light or deep sleep mode for a mesh peering can be found.

Since mesh STAs in light or deep sleep mode may transmit beacons at long intervals, a mesh STA seeking for candidate mesh STAs for a new mesh peerings should perform passive scanning relatively for a longer time compared to passive scanning in BSS infrastructure mode. Mesh STAs in light or deep sleep mode with long DTIM interval might not be discovered with short scanning durations. Mesh STAs that operate in light or deep sleep mode for a mesh peering may use a short DTIM interval, if they intend to establish new mesh peerings.

Y.3.4 Default parameters

The following are the recommended default values for power save related parameters for mesh STAs:

A mesh STA that is eager to conserve power and likely to remain in deep sleep mode with most of the mesh peerings should utilize the value from aggressive power save parameters.

Table Y-1—Default parameters for mesh STAs that intend to operate in light or deep sleep mode for mesh peerings

	For moderate power save	For aggressive power save
Beacon period	200 TU	800 TU
DTIM Period	4	1
Mesh awake window	10 TU	10 TU

Although mesh STAs may utilize individual parameters regardless of the parameters used by neighbor mesh STAs or peer mesh STAs, each implementer should recognize balance between the power save efficiency and delay in the service initiation.

Y.3.5 MSDU forwarding in an MBSS containing mesh STAs in light or deep sleep mode

The battery powered mesh STAs should avoid forwarding MSDUs, to avoid power consumption and possible additional delays and inefficiencies that power save mechanisms may cause. If a light or deep sleep mode mesh STA forwards MSDUs, it should select its mesh power mode based on the traffic load and the traffic type as described in Y.3.2.

The mesh STAs that desire to save power may select the light or deep sleep mode for a mesh peering as follows:

— mesh STAs that are “mains powered” may apply light or deep sleep mode if they do not have MSDUs to forward

— mesh STAs that are “battery powered” and desire to minimize the power consumption may freely use all mesh power modes

Mesh STAs that are in light or deep sleep mode for a mesh peering may degrade the corresponding link metric value. The use of worse metric values reduces the probability of a link being used for MSDU forwarding. If the mesh STA will operate in active mode for the link in forwarding path, it should apply the link metric value without degradation.

Y.3.6 Synchronization maintenance of mesh STAs in deep sleep mode

A mesh STA in deep sleep mode for a mesh peering might not receive Beacon frames from the corresponding peer mesh STA, but the mesh STA is required to maintain synchronization with the neighbor peer mesh STAs. Neighbor offset synchronization method imposes the maintenance of TSF offset values between neighbor peer mesh STAs, which generally requires the reception of Beacon frame or Probe Response frame. The simplest way to maintain the synchronization is that the mesh STA in deep sleep mode for a mesh peering listens to the corresponding peer mesh STA’s Beacon frame for certain periods and check the TSF offset value.

Y.4 SIV key wrapping test vector

```
This test vector is from the appendix of IETF RFC 5297.

   Input:
   -----
   Key:
           fffefdfc fbfaf9f8 f7f6f5f4 f3f2f1f0
```

```
        f0f1f2f3 f4f5f6f7 f8f9fafb fcfdfeff

Associated Data (ad):
        10111213 14151617 18191a1b 1c1d1e1f
        20212223 24252627

Plaintext:
        11223344 55667788 99aabbcc ddee

S2V-CMAC-AES
------------
CMAC(zero):
        0e04dfaf c1efbf04 01405828 59bf073a

double():
        1c09bf5f 83df7e08 0280b050 b37e0e74

CMAC(ad):
        f1f922b7 f5193ce6 4ff80cb4 7d93f23b

xor:
        edf09de8 76c642ee 4d78bce4 ceedfc4f

double():
        dbe13bd0 ed8c85dc 9af179c9 9ddbf819

pad:
        11223344 55667788 99aabbcc ddee8000

xor:
        cac30894 b8eaf254 035bc205 40357819

CMAC(final):
        85632d07 c6e8f37f 950acd32 0a2ecc93

CTR-AES
-------
CTR:
        85632d07 c6e8f37f 150acd32 0a2ecc93

E(K,CTR):
        51e218d2 c5a2ab8c 4345c4a6 23b2f08f

ciphertext:
        40c02b96 90c4dc04 daef7f6a fe5c

output
------
IV || C:
        85632d07 c6e8f37f 950acd32 0a2ecc93
        40c02b96 90c4dc04 daef7f6a fe5c
```

Y.5 Airtime link metric usage example

The airtime cost constants (Table 11C-4) and estimates of the average data rate and frame error rate will vary from one implementation and configuration of the IEEE 802.11 PHY and MAC to the other. While no mechanism is defined to measure the average data rate and the frame error rate, it is expected that numeric values will not exhibit large non-monotonic variations in amplitude over the lifetime of a path. Unstable measurements may cause path selection instabilities.

An example of an airtime link metric is provided to illustrate how it may be calculated.

Assume a DSSS PHY with an average data rate of 1 Mb/s to a given neighbor and a frame size of 8192 bits.

The overhead O for the data frame is comprised of the PLCP preamble (144 µs) and the PLCP header (48 µs). The payload Bit is 8192 bits at an r of 1 Mb/s, or 8192 µs. The data transmission time is therefore 8416 µs. Other transmission times for different frame types are calculated in the same way (based on their size, rate, and overhead).

If RTS/CTS is used, the data transmission time (including PLCP preamble and header) is 8416 µs, the RTS transmission time (including PLCP preamble and header) is 352 µs, the CTS transmission time (including PLCP preamble and header) is 304 µs, the ACK transmission time (including PLCP preamble and header) is 304 µs and the interframe spacing overhead is 390 µs. The total time, including overhead, is 9766 µs.

This airtime and overhead value is converted to units of 0.01 TU (10.24 µs), i.e., 953.71 (rounded to 954).

If the frame error rate to the neighbor is 0%, the metric is 954. If the frame error rate is 80%, the metric is 4769 (i.e., 953.71/(1–0.8), rounded).

Y.6 Generation of proactive PREPs in proactive PREQ mechanism of HWMP

Y.6.1 General

In the proactive PREQ mechanism of HWMP, the generation of a proactive PREP in response to the receipt of a proactive PREQ depends on the value of the Proactive PREP subfield in the received proactive PREQ (see 11C.9.4.2). Furthermore, if the Proactive PREP subfield is 0, the mesh STA may respond with the proactive PREP in case it needs a bidirectional path between the root mesh STA and itself. This is usually the case if the mesh STA has data to be sent to the root mesh STA. This clause provides a unified mechanism that controls the generation of proactive PREPs in the proactive PREQ mechanism of HWMP.

A proactive PREQ is defined by all of the following:

— The Target Address is set to all ones; and

— The TO subfield in the Per Target Flags field is 1.

Y.6.2 Additions to forwarding information

The forwarding information to a root mesh STA contains two additional Boolean information fields. The Proactive PREP field indicates whether the mesh STA will generate a proactive PREP to the root mesh STA in response to a proactive PREQ (Proactive PREP = 1) or not (Proactive PREP = 0). The Proactive PREP Sent field indicates whether the mesh STA has sent a proactive PREP to the root mesh STA (Proactive PREP Sent = 1) or not (Proactive PREP Sent = 0). Both fields are initialized with 0.

Y.6.3 Actions when sending data frames as source mesh STA

If a mesh STA receives proactive PREQs with the Proactive PREP subfield set to 0, the recipient mesh STA sends a proactive PREP before sending a data frame as source mesh STA only if the mesh STA has data to send to the root mesh STA, which requires establishing a bidirectional path with the root mesh STA and the field Proactive PREP Sent of the forwarding information to the root mesh STA is not set (=0).

If the mesh STA sends a data frame as source mesh STA to the root mesh STA, the mesh STA sets the field Proactive PREP of the forwarding information to 1.

Y.6.4 Actions on receipt of proactive PREQ

If the mesh STA receives a proactive PREQ, the field Proactive PREP Sent of the forwarding information to the root mesh STA is set to 0. If the field Proactive PREP of the forwarding information to the root mesh STA is 1, the mesh STA generates a proactive PREP to the root mesh STA (see 11C.9.10.3 Case D), sets the field Proactive PREP of the forwarding information to 0, and sets the field Proactive PREP Sent of the forwarding information to 1.

If the Proactive PREP subfield of the Flags field of the received proactive PREQ is 1, the mesh STA sets the field Proactive PREP of the forwarding information to the root mesh STA to 1.

Y.6.5 Generation of proactive PREPs

A mesh STA will generate a proactive PREP according to 11C.9.10.3 Case D if one of the following applies:

- [The mesh STA has received a proactive PREQ] AND [in the forwarding information to the root mesh STA of the proactive PREQ, field Proactive PREP is set to 1 AND field Proactive PREP Sent is set to 0].
- [The mesh STA has data to send to the root mesh STA which requires establishing a bidirectional path with the root mesh STA] AND [the field Proactive PREP Sent of the forwarding information to the root mesh STA is not set (=0)].

If the mesh STA generates a proactive PREP to the root mesh STA, the field Proactive PREP Sent of the forwarding information is set to 1 and the field Proactive PREP of the forwarding information is set to 0.

Y.7 Generation of PREQs in proactive RANN mechanism of HWMP

Y.7.1 General

In the proactive RANN mechanism of HWMP, the generation of a PREQ for root path confirmation in response to the receipt of a RANN depends on the path metric from the mesh STA to the root mesh STA as computed by the RANN propagation. However, the RANN mechanism does not setup the necessary forwarding information. This is done with individually addressed PREQs. This clause provides further details to the generation of individually addressed PREQs in the proactive RANN mechanism of HWMP.

An individually addressed PREQ is defined by the following:

- The Addressing Mode subfield in the Flags field is 1 (individually addressed).

Y.7.2 Additions to forwarding information

The forwarding information to a root mesh STA contains an additional address field. The RANN Sender Address field contains the MAC address of the neighbor peer mesh STA that has sent the RANN with the best metric.

Y.7.3 Actions when sending data frames as source mesh STA

If the mesh STA sends a data frame as source mesh STA to the root mesh STA, the mesh STA uses the forwarding information to the root mesh STA. The RANN Sender Address is not used for forwarding Mesh Data frames to the root mesh STA.

Y.7.4 Actions on receipt of proactive RANN

If the mesh STA receives a proactive RANN, the field RANN Sender Address of the forwarding information to the root mesh STA is set to the sender of the RANN element if the metric to the root mesh STA is better than the path metric of the existing mesh path in the forwarding information.

In this case, the mesh STA generates an individually addressed PREQ to the root mesh STA—see 11C.9.9.3 Case D (Root Path Confirmation (Original Transmission)). This individually addressed PREQ is not sent according to the general forwarding information. Instead, it is sent to the neighbor peer mesh STA indicated in the RANN Sender Address field.

Since all mesh STAs between the mesh STA and the root mesh STA have already a value in the RANN Sender Address field, the individually address PREQ will eventually reach the root mesh STA. Since the TO subfield in the Per Target Flags field is 1, only the root mesh STA will generate a PREQ as reply to the individually addressed PREQ.

The individually addressed PREQs will setup forwarding information (that can be used for forwarding Mesh Data frames) from the root mesh STA towards the mesh STA that initiated the root path confirmation in all intermediate mesh STAs according to the normal HWMP procedures.

Note, since the PREQ is sent in an individually addressed frame, every sender of the individually addressed PREQ will be able to determine whether the next mesh STA towards the root mesh STA has received the PREQ.

The root mesh STA will generate a PREP according to the normal HWMP procedures—see 11C.9.10.3 Case A (Original Transmission).

Y.7.5 Actions on receipt of PREP

The PREP generated by the root mesh STA will be forwarded on the mesh path from the root mesh STA to the mesh STA as setup by the individually addressed PREQ. The PREP will be propagated and will eventually reach the mesh STA. The PREP will setup forwarding information (that can be used for forwarding Mesh Data frames) towards the root mesh STA in the mesh STA that initiated the root path confirmation and in all intermediate mesh STAs on this path according to the normal HWMP procedures.

After establishing the forwarding information (mesh path) towards the root mesh STA, the path to the root mesh STA is updated to the better one.

Note, since the PREP is sent in an individually addressed frame, every sender of the PREP will be able to determine whether the next mesh STA has received the PREP.

Y.8 Informative references[2]

IETF RFC 3561, “Ad hoc On-Demand Distance Vector (AODV) Routing,” C. Perkins, E. Belding-Royer, S. Das, July 2003. (status: experimental).

[2] Internet RFCs are available from the Internet Engineering Task Force at http://www.ietf.org/.