802.11e™

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 8: Medium Access Control (MAC) Quality of Service Enhancements

IEEE Computer Society

Sponsored by the
LAN/MAN Standards Committee

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

11 November 2005

IEEE Std 802.11e™-2005
(Amendment to IEEE Std 802.11™, 1999 Edition (Reaff 2003)
as amended by IEEE Std 802.11a™-1999, IEEE Std 802.11b™-1999,
IEEE Std 802.11b-1999/Cor 1-2001, IEEE Std 802.11d™-2001,
IEEE Std 802.11g™-2003, IEEE Std 802.11h™-2003,
IEEE Std 802.11i™-2004, and IEEE Std 802.11j™-2004)

IEEE Std 802.11e™-2005
(Amendment to IEEE Std 802.11™, 1999 Edition (Reaff 2003)
as amended by IEEE Std 802.11a™-1999, IEEE Std 802.11b™-1999,
IEEE Std 802.11b-1999/Cor 1-2001, IEEE Std 802.11d™-2001,
IEEE Std 802.11g™-2003, IEEE Std 802.11h™-2003,
IEEE Std 802.11i™-2004, and IEEE Std 802.11j™-2004)

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 8: Medium Access Control (MAC) Quality of Service Enhancements

Sponsor
LAN/MAN Committee
of the
IEEE Computer Society

Approved 22 September 2005

IEEE-SA Standards Board

Abstract: This amendment defines the medium access control (MAC) procedures to support local area network (LAN) applications with quality of service (QoS) requirements. The procedures include the transport of voice, audio, and video over IEEE 802.11 wireless LANs (WLANs).
Keywords: BA, block acknowledgement, carrier sense multiple access/collision avoidance, CSMA/CA, priority, polling, power save, QoS, quality of service, traffic stream

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

Print: ISBN 0-7381-4772-9 SH95357/0
PDF: ISBN 0-7381-4773-7 SS95357/0

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 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. When a document is more than five years old and has not been reaffirmed, 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. 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. Comments on standards and requests for interpretations should be addressed to:

Secretary, IEEE-SA Standards Board

445 Hoes Lane

Piscataway, NJ 08854

USA

NOTE—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 shall not be responsible for identifying patents for which a license may be required by an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Introduction

This introduction is not part of IEEE Std 802.11e-2005, 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 8: Medium Access Control (MAC) Quality of Service (QoS) Enhancements).

This amendment defines the medium access control (MAC) procedures to support local area network (LAN) applications with quality of service (QoS) requirements, including the transport of voice, audio, and video over IEEE 802.11 wireless LANs (WLANs).

Notice to users

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 shall not be responsible for identifying patents or patent applications for which a license may be required to implement an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Participants

When this amendment was sent to sponsor ballot, the IEEE 802.11 Working Group had the following officers:

Stuart J. Kerry, *Chair*
Al Petrick and **Harry Worstell,** *Vice-Chairs*
Tim Godfrey, *Secretary*
Brian Mathews, *Publicity Standing Committee*
Tan Teik-Kheong, *Wireless Next Generation Standing Committee*
Terry Cole, *Editor*

Dave Halasz, *Chair Task Group i*
Shueng Li, *Chair Task Group j*
Richard Paine, *Chair Task Group k*
Bob O'Hara, *Chair Task Group m*

Bruce P. Kramer, *Chair Task Group n*

When this amendment was sent to sponsor ballot, Task Group e had the following officers:

John Fakatselis, *Chair*
Duncan Kitchin, *Vice-Chair*
Srinivas Kandala, Michael Fischer*, Editors*
Robert Miller, David Hunter, Tim Godfrey, *Secretaries*

When this amendment was sent to sponsor ballot, the IEEE 802.11 Working Group had the following membership:

Osama Aboul-Magd
Tomoko Adachi
Jaemin Ahn
Thomas Alexander
Areg Alimian
Richard Allen
Keith Amann
Dov Andelman
Merwyn Andrade
Carl Andren
David Andrus
Butch Anton
Hidenori Aoki
Tsuguhide Aoki
Michimasa Aramaki
Takashi Aramaki
William Arbaugh
Lee Armstrong
Larry Arnett
Hiroshi Asai
Yusuke Asai
Arthur Astrin
Malik Audeh
Geert Awater
Shahrnaz Azizi
Floyd Backes
Jin-Seok Bae
David Bagby
Jay Bain
Dennis Baker
Ramanathan Balachander
Jaiganesh Balakrishnan
Boyd Bangerter
John Barr
Simon Barber
Farooq Bari
Michael Barkway
Kevin Barry
Anuj Batra
Burak Baysal
Tomer Bentzion
Mathilde Benveniste
Don Berry
Jan Biermann
Arnold Bilstad
Harry Bims
Bjorn Bjerke
Simon Black
Jan Boer
William Brasier
Jennifer Bray
Phillip Brownlee
Alex Bugeja
Alistair Buttar
Peter Cain
Richard Cam
Nancy Cam-Winget
Bill Carney
Pat Carson
Broady Cash
Jayant Chande
Kisoo Chang
Clint Chaplin
Ye Chen
Hong Cheng
Greg Chesson
Aik Chindapol
Sunghyun Choi
Won-Joon Choi
Woo-Yong Choi
Yang-Seok Choi
Per Christoffersson
Simon Chung
Ken Clements
Sean Coffey
Paul Congdon
W. Steven Conner
Charles Cook
Kenneth Cook
Todor Cooklev
Mary Cramer
Steven Crowley
Nora Dabbous
Rolf De Vegt
Javier Del Prado Pavon
Georg Dickmann
Wim Diepstraten
Yoshiharu Doi
Brett Douglas
Simon Duggins
Baris Dundar
Bryan Dunn
Roger Durand
Eryk Dutkiewicz
Mary DuVal
Yaron Dycian
Donald Eastlake
Dennis Eaton
Peter Ecclesine
Jonathan Edney
Bruce Edwards
Natarajan Ekambaram
Jason Ellis
Darwin Engwer
Jeff Erwin
Andrew Estrada
Christoph Euscher
Knut Evensen
Lars Falk
Steve Fantaske
Paul Feinberg
Alex Feldman
Weishi Feng
Nestor Fesas
Matthew Fischer
Wayne Fisher
Helena Flygare
Brian Ford
Ruben Formoso
Sheila Frankel
John Fuller
James Gardner
Atul Garg
Albert Garrett
Ramez Gerges
Noam Geri
Vafa Ghazi
Monisha Ghosh
James Gilb
Jeffrey Gilbert
Rabinder Gill
Wataru Gohda
Yuri Goldstein
Jim Goodman
Aviv Goren
Andrew Gowans
Rik Graulus
Gordon Gray
Evan Green
Larry Green
Patrick Green
Kerry Greer
Daqing Gu
Rajugopal Gubbi
Sam Guirguis
Srikanth Gummadi
Qiang Guo
Vivek Gupta

Herman Haisch
David Halasz
Steven Halford
Robert Hall
Neil Hamady
Mounir Hamdi
Christopher Hansen
Yasuo Harada
Daniel Harkins
Thomas Haslestad
Amer Hassan
Vann Hasty
James Hauser
Yutaka Hayakawa
Morihiko Hayashi
Haixiang He
Xiaoning He
Robert Heile
Eleanor Hepworth
Frans Hermodsson
Dave Hetherington
Guido Hiertz
Garth Hillman
Christopher Hinsz
Jun Hirano
Mikael Hjelm
Jin-Meng Ho
Michael Hoghooghi
Allen Hollister
Keith Holt
Satoru Hori
William Horne
Srinath Hosur
Russell Housley
Frank Howley
Yungping Hsu
Robert Huang
Dave Hudak
Syang-Myau Hwang
David Hytha
Muhammad Ikram
Daichi Imamura
Kimihiko Imamura
Yasuhiko Inoue
Katsumi Ishii
Stephen Jackson
Eric Jacobsen
Marc Jalfon
KyungHun Jang
Bruno Jechoux
Taehyun Jeon
Moo Ryong Jeong
Daniel Jiang
Kuniko Jimi
Walter Johnson
David Johnston
Jari Jokela
VK Jones
Bobby Jose
Tyan-Shu Jou
Carl Kain
You Sung Kang
Jeyhan Karaoguz
Kevin Karcz
Pankaj Karnik
Mika Kasslin
Dean Kawaguchi
Patrick Kelly
Richard Kennedy
John Ketchum
Vytas Kezys
Andrew Khieu
Ryoji Kido
Tomohiro Kikuma
Byoung-Jo Kim
Dooseok Kim
Joonsuk Kim
Yongbum Kim
Yongsuk Kim
Young Kim
Youngsoo Kim
Wayne King
John Klein
Guenter Kleindl
Toshiya Kobashi
Keiichiro Koga
Lalit Kotecha
John Kowalski
Bruce Kraemer
Gopal Krishnan
Shuji Kubota
Thomas Kuehnel
Tomoaki Kumagai
Takushi Kunihiro
Thomas Kurihara
Denis Kuwahara
Joe Kwak
Paul Lambert
David Landeta
Jim Lansford
Colin Lanzl
Jon LaRosa
Choi Law
David Leach
Dongjun Lee
Insun Lee
Jae Hwa Lee
Marty Lefkowitz
Onno Letanche
Joseph Levy
Mike Lewis
Pen Li
Quinn LI
Sheung Li
Jie Liang
Wei Lih Lim
Yong Je Lim
Huashih Lin
Sheng Lin
Victor Lin
Stanley Ling
Der-Zheng Liu
I-Ru Liu
Yonghe Liu
Titus Lo
Peter Loc
Patrick Lopez
Hui-Ling Lou
Xiaolin Lu
Luke Ludeman
Yi-Jen Lung
Akira Maeki
Ravishankar Mahadevappa
Doug Makishima
Majid Malek
Rahul Malik
Jouni Malinen
Krishna Malladi
Stefan Mangold
Mahalingam Mani
Jonn Martell
Naotaka Maruyama
Paul Marzec
Yoichi Matsumoto
Sudheer Matta
Thomas Maufer
Conrad Maxwell
Stephen McCann
Kelly McClellan
Gary McCoy
William McFarland
Timothy McGovern
Bill McIntosh
Justin McNew
Irina Medvedev
Pratik Mehta
Robert Meier
Graham Melvile
Klaus Meyer
Partho Mishra
David Mitton
Kenichi Miyoshi
Rishi Mohindra
Peter Molnar
Leo Monteban
Michael Montemurro
Rondal Moore
Tim Moore
Anthony Morelli
Mike Moreton
Yuichi Morioka
Steven Morley
Robert Moskowitz
Joseph Mueller
Syed Mujtaba
Willem Mulder
Peter Murphy
Peter Murray
Andrew Myers
Andrew Myles
Yukimasa Nagai
Katsuyoshi Naka
Makoto Nakahara
Michiharu Nakamura
Seigo Nakao
Hiroyuki Nakase
Sanjiv Nanda
Ravi Narasimhan
Slobodan Nedic
Robert Neilsen
David Nelson

Dan Nemits
Chiu Ngo
Tuan Nguyen
Qiang Ni
Gunnar Nitsche
Erwin Noble
Tzvetan Novkov
Ivan Oakes
Kei Obara
Karen O'Donoghue
Hiroshi Oguma
Jongtaek Oh
Sean O'Hara
Yoshihiro Ohtani
Chandra Olson
Timothy Olson
Hiroshi Ono
Peter Oomen
Lior Ophir
Satoshi Oyama
Michael Paljug
Stephen Palm
Jong Ae Park
Jonghun Park
Joon Goo Park
Taegon Park
Steve Parker
Glenn Parsons
Vijay Patel
Eldad Perahia
Sebastien Perrot
Joe Pitarresi
Leo Pluswick
Stephen Pope
James Portaro
Al Potter
Henry Ptasinski
Anuj Puri
Aleksandar Purkovic
Jim Raab
Ali Raissinia
Ajay Rajkumar
Noman Rangwala
Ivan Reede
Stanley Reible
Anthony Reid
Joe Repice
Edward Reuss
Valentine Rhodes
Maximilian Riegel
Edmund Ring
Carlos Rios
Stefan Rommer
Jon Rosdahl
John Sadowsky
Ali Sadri
Kazuyuki Sakoda
Shoji Sakurai
Kenichi Sakusabe
Hemanth Sampath
Sumeet Sandhu
Anil Sanwalka
Ryo Sawai
Tom Schaffnit
Brian Schreder
Sid Schrum
Erik Schylander
Michael Seals
Joe Sensendorf
N. Shankaranarayanan
Donald Shaver
Stephen Shellhammer
Tamara Shelton
Yangmin Shen
Ian Sherlock
Matthew Sherman
Ming Sheu
Shusaku Shimada
Matthew Shoemake
William Shvodian
D. J. Shyy
Thomas Siep
Floyd Simpson
Manoneet Singh
Hasse Sinivaara
Efstratios (Stan) Skafidas
David Skellern
Roger Skidmore
Donald Sloan
Kevin Smart
David Smith
Yoram Solomon
V. Somayazulu
Amjad Soomro
Robert Soranno
Gary Spiess
William Spurgeon
Dorothy Stanley
William Steck
Greg Steele
Adrian Stephens
William Stevens
Carl Stevenson
Fred Stivers
Warren Strand
Paul Struhsaker
Michael Su
Hiroki Sugimoto
Abhaya Sumanasena
Qinfang Sun
SK Sung
Shravan Surineni
Hirokazu Tagiri
Masahiro Takagi
Mineo Takai
Katsumi Takaoka
Daisuke Takeda
Nir Tal
Tsuyoshi Tamaki
Pek-Yew Tan
Wai-Cheung Tang
Takuma Tanimoto
Henry Taylor
James Taylor
Carl Temme
Stephan ten Brink
John Terry
Timothy Thornton
Jerry Thrasher
James Tomcik
Allen Tsai
Jean Tsao
Chih Tsien
Tom Tsoulogiannis
Kwei Tu
David Tung
Sandra Turner
Mike Tzamaloukas
Marcos Tzannes
Yusuke Uchida
Takashi Ueda
Naoki Urano
Hidemi Usuba
Chandra Vaidyanathan
Hans Van Leeuwen
Richard van Leeuwen
Richard Van Nee
Nico van Waes
Allert van Zelst
Narasimhan Venkatesh
Madan Venugopal
George Vlantis
Dennis Volpano
Tim Wakeley
Jesse Walker
Brad Wallace
Thierry Walrant
Vivek Wandile
Huaiyuan (Stephen) Wang
Stanley Wang
Christopher Ware
Fujio Watanabe
Mark Webster
Matthew Welborn
Bryan Wells
Filip Weytjens
Stephen Whitesell
Michael Wilhoyte
Michael Glenn Williams
Peter Williams
Richard Williams
James Wilson
Steven Wilson
Jack Winters
Jin Kue Wong
Timothy Wong
Patrick Worfolk
Charles Wright
Gang Wu
Yang Xiao
James Yee
Jung Yee
Kazim Yildiz
Jijun Yin
Kit Yong
Wonyong Yoon
Heejung Yu
Hon Yung
Erol Yurtkuran
Zhun Zhong
Glen Zorn
James Zyren

Major contributions were received from the following individuals:

Keith Amann
Mathilde Benveniste
A. Mark Bilstad
Jennifer A. Bray
Nancy Cam-Winget
Clint Chaplin
Ye Chen
Yi-Ming Chen
Greg Chesson
Sunghyun Choi
Xavier Perez Costa
Javier del Prado
Georg Dickmann
Wim Diepstraten
Andrew Estrada
Steve Emeott
John Fakatselis
Matthew Fischer
Michael Fischer
Wataru Gohda
Rajugopal Gubbi
Jin-Meng Ho
Maarten Hoeben
Frank P. Howley, Jr.
Peter Johannson
Jari Jokela
Bobby Jose
Srinivas Kandala
Duncan Kitchin
Jarkko Kneckt
Andrei Kojukhov
John Kowalski
Thomas Kuehnel
Jie Liang
Isaac Lim Wei Lih
Jouni Malinen
Thomas Maufer
Robert C. Meier
Robert Miller
Tim Moore
Andrew Myles
Ivan Oakes
Yoshihiro Ohtani
Henry Ptasinski
Ali Raissini
Sid Schrum
Matthew Sherman
Floyd Simpson
Aman Singla
Amjad Soomro
Dorothy Stanley
Adrian Stephens
Minoru Takemoto
Pek-Yew Tan
Toru Ueda
Richard van Leeuwen
Jesse Walker
Huaiyuan (Stephen) Wang
Christopher Ware
Menzo Wentink
Charles Wright
Shugong Xu

This amendment was balloted using individual balloting. The following members of the balloting committee voted on this amendment. Balloters may have voted for approval, disapproval, or abstention.

Tomoko Adachi
Toru Aihara
Keith Amann
Butch Anton
Eladio Arvelo
David Bagby
Daniel Bailey
Jay Bain
Steven Bard
John Barr
Les Baxter
Jan Boer
Gennaro Boggia
Lon Canaday
Edward Carley
Bill Carney
Clint Chaplin
Amalavoyal Chari
Aik Chindapol
Keith Chow
Terry Cole
Christopher Cooke
Todor Cooklev
Todd Cooper
Javier del Prado Pavon
Guru Dutt Dhingra
Georg Dickmann
Thomas Dineen
Sourav Dutta
Clint Early
Richard Eckard
Carl Eklund
John Eng
Michael Fischer
Avraham Freedman
John Fuller
Michele Gammel
Ernesto Garcia
Theodore Georgantas
Andrew Germano
James Gilb
Tim Godfrey
Rubén González-Benítez
Jose Gutierrez
Chris Guy
David Halasz
Christopher Hansen
Yasuo Harada
Atsushi Ito
Peeya Iwagoshi
Stanley Johnson
David Johnston
Bobby Jose
Joe Juisai
Thomas M. Kurihara
Srinivas Kandala
Diana Kang
Kevin Karcz
Stuart Kerry
Brian Kiernan
Duncan Kitchin
Cees Klik
John Kowalski
Joe Kubler
Colin Lanzl
Daniel Levesque
Jie Liang
Jan-Ray Liao
Randolph Little
Gregory Luri
Ryan Madron
Kevin Magee
Roger Marks
Michael McInnis
Ingolf Meier
George Miao
Apurva Mody
Leo Monteban
Jeremy Moore
Mike Moreton
Narayanan Murugesan
Andrew Myles
Paul Nikolich
Erwin Noble
Satoshi Obara
Bob O'Hara
Yoshihiro Ohtani
Satoshi Oyama
Stephen Palm
Roger Pandanda
Sebastien Perrot
Subbu Ponnuswamy
Vikram Punj
Terry Richards
Maximilian Riegel
David Rockwell
Mike Rudnick
Durga Satapathy
William Scanlon
Michael Scholles
Matthew Sherman
Gil Shultz
Thomas Siep

Yoram Solomon
Amjad Soomro
Thomas Starai
Adrian Stephens
Carl Stevenson
Masahiro Takagi
Minoru Takemoto
Pek-Yew Tan
Jerry Thrasher
James Tomcik
Toru Ueda
Scott Valcourt
Richard van Leeuwen
Hung-yu Wei
Menzo Wentink
Stephen Whitesell
Dave Willow
Harry Worstell
Shugong Xu
Jung Yee
Patrick Yu
Oren Yuen

When the IEEE-SA Standards Board approved this amendment on 22 September 2005, it had the following membership:

Steve M. Mills, *Chair*
Richard H. Hulett, *Vice Chair*
Don Wright, *Past Chair*
Judith Gorman, *Secretary*

Mark D. Bowman
Dennis B. Brophy
Joseph Bruder
Richard Cox
Bob Davis
Julian Forster*
Joanna N. Guenin
Mark S. Halpin
Raymond Hapeman
William B. Hopf
Lowell G. Johnson
Herman Koch
Joseph L. Koepfinger*
David J. Law
Daleep C. Mohla
Paul Nikolich
T. W. Olsen
Glenn Parsons
Ronald C. Petersen
Gary S. Robinson
Frank Stone
Malcolm V. Thaden
Richard L. Townsend
Joe D. Watson
Howard L. Wolfman

*Member Emeritus

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

Satish K. Aggarwal, *NRC Representative*
Richard DeBlasio, *DOE Representative*
Alan Cookson, *NIST Representative*

Michael D. Fisher
IEEE Standards Project Editor

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 8: Medium Access Control (MAC) Quality of Service Enhancements

[This amendment is based on IEEE Std 802.11™, 1999 Edition (Reaff 2003), as amended by IEEE Std 802.11a™-1999, IEEE Std 802.11b™-1999, IEEE Std 802.11b-1999/Cor 1-2001, IEEE Std 802.11d™-2001, IEEE Std 802.11g™-2003, IEEE Std 802.11h™-2003, IEEE Std 802.11i™-2004, and IEEE Std 802.11j™-2004.]

NOTE—The editing instructions contained in this amendment define how to merge the material contained herein into the existing base standard and its amendments to form the comprehensive standard.

The editing instructions are shown in ***bold italic***. Three editing instructions are used: change, delete, and insert. ***Change*** is used to make small corrections in existing text or tables. The editing instruction specifies the location of the change and describes what is being changed either by using ~~strikethrough~~ (to remove old material) or 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 instructions. Editorial notes will not be carried over into future editions.

1. Overview

1.2 Purpose

Insert the following text at the end of 1.2, as part of the dashed list:

— Defines the medium access control (MAC) procedures to support local area network (LAN) applications with quality of service (QoS) requirements, including the transport of voice, audio, and video over IEEE 802.11 wireless LANs (WLANs).

End of changes to Clause 1.

2. Normative references

Insert the following citation in alpha-numerical order in Clause 2:

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

End of changes to Clause 2.

3. Definitions

Change the definition of "coordination function" in 3.13 as follows:

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

Change the term "coordination function pollable" to "contention-free pollable" and change its definition in 3.14 as follows:

3.14 ~~coordination function~~contention-free (CF) pollable: A station (STA) that is able to ~~(1)~~ respond to a ~~coordination function~~CF poll with a data frame, if such a frame is queued and able to be generated~~, and (2) interpret acknowledgments in frames sent to or from a point coordinator~~.

Insert the following new definitions in alphabetical order into Clause 3, renumbering as necessary:

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

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

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

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

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

3.128 controlled access phase (CAP): A time period when the hybrid coordinator (HC) maintains control of the medium, after gaining medium access by sensing the channel to be idle for a point coordination

function (PCF) interframe space (PIFS) duration. It may span multiple consecutive transmission opportunities (TXOPs) and can contain polled TXOPs.

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

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

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

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

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

3.134 fragmentation: The process of partitioning a medium access control (MAC) service data unit (MSDU) or MAC management protocol data unit (MMPDU) into a sequence of smaller MAC protocol data units (MPDUs) prior to transmission. The process of recombining a set of fragment MPDUs into an MSDU or MMPDU is known as defragmentation.

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

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

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

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

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

3.140 non-access point (non-AP) quality of service (QoS) station (QSTA): A station (STA) that supports the QoS facility, but is not an access point (AP). A non-AP QSTA does not have an hybrid coordinator (HC) and uses the QoS AP (QAP) for the distribution system services (DSSs).

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

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

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

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

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

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

3.147 quality of service (QoS) access point (QAP): An access point (AP) that supports the QoS facility specified in this amendment. The functions of a QAP are a superset of the functions of a non-QAP (nQAP), and thus a QAP is able to function as an nQAP to non-QoS stations (nQSTAs).

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

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

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

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

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

3.153 service period (SP): A contiguous time during which one or more downlink unicast frames are transmitted to a quality of service (QoS) station (QSTA) and/or one or more transmission opportunities (TXOPs)

are granted to the same QSTA. SPs can be scheduled or unscheduled. For a non-access point (non-AP) QSTA, there can be at most one SP active at any time.

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

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

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

3.157 traffic identifier (TID): Any of the identifiers usable by higher layer entities to distinguish medium access control (MAC) service data units (MSDUs) to MAC entities that support quality of service (QoS) within the MAC data service. There are 16 possible TID values; 8 identify TCs, and the other 8 identify parameterized TSs. The TID is assigned to an MSDU in the layers above the MAC.

3.158 traffic specification (TSPEC): The quality of service (QoS) characteristics of a data flow to and from a non-access point (non-AP) QoS station (QSTA).

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

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

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

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

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

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

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

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

End of changes to Clause 3.

4. Abbreviations and acronyms

Delete the acronym CID from Clause 4 as follows:

~~CID~~ ~~connection identifier~~

Insert the following acronyms in alphabetical order in Clause 4:

AC	access category
ACI	access category index
ACM	admission control mandatory
ACU	admission control unit
ADDBA	add Block Acknowledgment
ADDTS	add traffic stream
AIFS	arbitration interframe space
AIFSN	arbitration interframe space number
APSD	automatic power-save delivery
APSD QSTAs	non-AP QoS stations that use automatic power-save delivery
BA	Block Acknowledgment
BAR	Block Acknowledgment request
CAP	controlled access phase
DELBA	delete Block Acknowledgment
DELTS	delete traffic stream
DLS	direct-link setup
DSCP	differentiated services code point
EDCA	enhanced distributed channel access
EDCAF	enhanced distributed channel access function
EOSP	end of service period
HCCA	HCF controlled channel access
HC	hybrid coordinator
HCF	hybrid coordination function
HEMM	HCCA, EDCA mixed mode
nQAP	non-QoS access point
nQBSS	non-QoS basic service set
nQSTA	non-QoS station
QAP	QoS access point
QBSS	QoS basic service set

QIBSS	QoS independent basic service set
QoS	quality of service
QLRC	QoS long retry counter
QSRC	QoS short retry counter
QSTA	QoS station
S-APSD	scheduled automatic power-save delivery
SI	service interval
SP	service period
TC	traffic category
TCLAS	traffic classification
TID	traffic identifier
TS	traffic stream
TSID	traffic stream identifier
TSPEC	traffic specification
TXOP	transmission opportunity
U-APSD	unscheduled automatic power-save delivery
UP	user priority

End of changes to Clause 4.

5. General description

5.1 General description of the architecture

5.1.1 How wireless LAN systems are different

Change the heading of 5.1.1.2 as shown:

5.1.1.2 ~~The m~~Media impact ~~the~~on design and performance

Change item b) of 5.1.1.2 as follows:

b) Are potentially unprotected from ~~outside~~ other signals that may be sharing the medium.

Insert the following text at the end of the lettered list in 5.1.1.2:

g) May experience interference from logically disjoint IEEE 802.11 networks operating in overlapping areas.

Insert the following text at the end of the final paragraph in 5.1.1.2:

When providing QoS services it should be understood that the MAC endeavors to provide QoS "service guarantees" within the limitations of the medium properties identified above. In other words, particularly in unlicensed spectrum, true guarantees are often not possible. However, gradations of service are always possible; and in sufficiently controlled environments, QoS guarantees can truly be made.

5.1.1.4 Interaction with other IEEE 802® layers

Insert the following paragraph at the end of 5.1.1.4:

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

5.2 Components of the IEEE 802.11 architecture

Insert after 5.2.4 the following subclause (5.2.5):

5.2.5 QoS basic service set (QBSS): The QoS network

The IEEE 802.11 QoS facility provides MAC enhancements to support LAN applications with QoS requirements. The QoS enhancements are available to QoS stations (QSTAs) associated with a QoS access point (QAP) in a QBSS. A subset of the QoS enhancements is available for use between QSTAs that are members of the same QoS independent basic service set (IBSS) (QIBSS). Because a QSTA implements a superset of station (STA) functionality, as defined in this standard, the QSTA may associate with a non-QoS access point (nQAP) in a non-QoS basic service set (BSS) (nQBSS), to provide non-QoS MAC data service when there is no QBSS with which to associate.

The enhancements that distinguish QSTAs from non-QoS STAs (nQSTAs) and QAPs from nQAPs are collectively termed the *QoS facility*. The quantity of certain, QoS-specific, mechanisms may vary among QoS implementations, as well as between QSTAs and QAPs, over ranges specified in subsequent clauses. All service primitives, frame formats, coordination function and frame exchange rules, and management interface functions except for the Block Acknowledgment (Block Ack) function, direct-link setup (DLS), and automatic power-save delivery (APSD) are part of the core QoS facilities. A QSTA or QAP must implement those core QoS facilities necessary for its QoS functions to interoperate with other QSTAs in the QBSS. Functions such as the Block Ack, DLS, and APSD are separate from the core QoS facilities; and the presence of these functions is indicated by QSTAs separately from the core QoS facilities. A comprehensive statement on mandatory and optional functionalities is available in Annex A.

This amendment provides two mechanisms for the support of applications with QoS requirements.

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

— Amount of time a STA senses the channel to be idle before backoff or transmission, or
— The length of the contention window to be used for the backoff, or
— The duration a STA may transmit after it acquires the channel.

These transmissions may also be subject to certain channel access restrictions in the form of admission control. Details of this mechanism are provided in 9.9.1.

The second mechanism, designated the *hybrid coordination function* (HCF) *controlled channel access* (HCCA), allows for the reservation of transmission opportunities (TXOPs) with the hybrid coordinator

(HC). A non-access point (non-AP) QSTA based on its requirements requests the HC for TXOPs, both for its own transmissions as well as for transmissions from the QAP to itself.[1] The request is initiated by the station management entity (SME) of the non-AP QSTA. The HC, which is collocated at the QAP, either accepts or rejects the request based on an admission control policy. If the request is accepted, the HC schedules TXOPs for both the QAP and the non-AP QSTA. For transmissions from the non-AP QSTA, the HC polls the non-AP QSTA based on the parameters supplied by the non-AP QSTA at the time of its request. For transmissions to the non-AP QSTA, the QAP directly obtains TXOPs from the collocated HC and delivers the queued frames to the non-AP QSTA, again based on the parameters supplied by the non-AP QSTA. Details of the mechanism are provided in 9.9.2 and 11.4. This mechanism may be used for applications such as voice and video, which may need periodic service from the HC. If the application constraints dictate the use of this mechanism, the application initiates this mechanism by using the management service primitives.

nQSTAs may associate in a QBSS, if allowed to associate by the QAP. All directed frames that are sent to nQSTAs by a QAP shall not use the frame formats associated with the QoS facility.

A QSTA associated in an nQBSS shall act as an nQSTA.

5.3 Logical service interfaces

Insert the following items at the end of the lettered list in 5.3:

l) Higher layer timer synchronization (QoS facility only)
m) QoS traffic scheduling (QoS facility only)

5.3.1 Station service (SS)

Insert the following items at the end of the lettered list in 5.3.1:

g) Higher layer timer synchronization (QoS facility only)
h) QoS traffic scheduling (QoS facility only)

5.3.2 Distribution system service (DSS)

Insert the following item at the end of the lettered list in 5.3.2:

f) QoS traffic scheduling (QoS facility only)

5.4 Overview of the services

Change the first paragraph in 5.4 as follows:

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

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

5.4.1 Distribution of messages within a DS

Insert after 5.4.1.2 the following subclause (5.4.1.3):

5.4.1.3 QoS traffic scheduling

QoS traffic scheduling provides intra-QBSS QoS frame transfers under the HCF, using either contention-based or controlled channel access. At each TXOP, a traffic scheduling entity at the QSTA selects a frame for transmission, from the set of frames at the heads of a plurality of traffic queues, based on requested UP and/or parameter values in the traffic specification (TSPEC) for the requested MSDU. Additional information is available in 9.9.

5.4.2 Services that support the distribution service

5.4.2.2 Association

Change the last paragraph of 5.4.2.2 as follows:

A STA learns what access points (APs) are present and what operational capabilities are available from each of those APs and then requests ~~to~~ the establishment of an association with an AP of appropriate capabilities by invoking the association service. For details of how a STA learns about what APs are present, see 11.1.3.

Insert after 5.4.4.2 the following subclauses (5.4.5 and 5.4.6):

5.4.5 Traffic differentiation and QoS support

IEEE 802.11 uses a shared medium and provides differentiated control of access to the medium to handle data transfers with QoS requirements. The QoS facility (per MSDU traffic class and TSPEC negotiation) allows an IEEE 802.11 LAN to become part of a larger network providing end-to-end QoS delivery or to function as an independent network providing transport on a per-link basis with specified QoS commitments. The specifications regarding the integration and operability of the QoS facility in IEEE 802.11 specification with any other end-to-end QoS delivery mechanism like Resource Reservation Protocol (RSVP) are beyond the scope of this amendment.

5.4.6 Support for higher layer timer synchronization

Some applications, e.g., the transport and rendering of audio or video streams, require synchronized timers shared among different STAs. Greater accuracy (in terms of jitter bounds) or finer timer granularity than that provided by a BSS timing synchronization function (TSF) may be an additional requirement. In support of such applications, this amendment defines a MAC service that enables layers above the MAC to accurately synchronize application-dependent timers shared among QSTAs. The service is usable by more than one application at a time.

Although the timer synchronization methods and accuracy requirements are application-dependent and are beyond the scope of this amendment, they rely on an indication from each QSTA's MAC that is provided essentially simultaneously, via multicast, to the QSTAs. The MAC accomplishes this by indicating the occurrence of the end of the last symbol of particular data frames; the data frames of interest are identified by their MAC header Address 1 field when it contains a group address previously registered with the MAC. The last symbol is observed[2] on the air by QSTAs within a BSS while the delay between the observation and the delivery of the indication is known within a MAC by design (and communicated to the application by implementation-dependent means). The common reference point in time provided by the end of last symbol indication is the essential building block upon which a variety of application-dependent timer synchronization methods may be based.

[2]The synchronization indication (observed time) within the BSS will vary slightly due to propagation.

5.5 Relationships between services

Change item vi) of Class 1 management frames in 5.5 as follows:

2) Management frames
   vi) Spectrum Management Action.

Change item i) of Class 1 data frames in 5.5 as follows:

3) Data frames
   i) Data: Data frames between STAs in an IBSS with frame control (FC) bits "To DS" and "From DS" both false.

Change item c) (Class 3 frames) in 5.5 as follows:

c) Class 3 frames (if and only if associated; allowed only from within State 3):
   1) Data frames
      i) Data subtypes: Data frames allowed. That is, the "To DS" and/or "From DS" FC bits may be set to true to utilize DSSs.
      ii) QoS data subtypes allowed to/from non-AP QSTA(s) that are associated with QAP(s).
      iii) Data frames between QSTAs in a QBSS with FC bits "To DS" and "From DS" both false.
   2) Management frames
      i) Deauthentication: Deauthentication notification when in State 3 implies disassociation as well, changing the STA's state from 3 to 1. The STA shall become authenticated again prior to another association.
      ii) QoS, DLS, and Block Ack Action
   3) Control frames
      i) Power save (PS)-Poll
      ii) Block Ack (BlockAck)
      iii) Block Ack Request (BlockAckReq)

5.6 Differences between ESS and IBSS LANs

Change the final paragraph in 5.6 as follows:

The services that apply to an IBSS are the SSs. A QIBSS supports operation under the HCF using TXOPs gained through the EDCA mechanism. The parameters that control differentiation of traffic classes using EDCA are fixed. A QIBSS has no HC and does not support polled TXOP operation and setting up of TSPEC.

5.7 Message information contents that support the services

5.7.1 Data

Change the text of 5.7.1 as shown:

For a STA to send data to another STA, it sends ~~a~~one of the following data messages~~,~~ as shown below:

Data messages

— Message type: Data
— Message subtype: Data
— Information items:
  • IEEE source address (SA) of message
  • IEEE destination address (DA) of message
  • BSS identification (BSSID)
— Direction of message: From QSTA or nQSTA to nQSTA

QoS data messages

— Message type: Data
— Message subtype: QoS Data
— Information items:
  • IEEE SA of message
  • IEEE DA of message
  • BSSID
  • Traffic identifier (TID)
— Direction of message: From QSTA to QSTA

5.7.2 Association

Insert in 5.7.2 under "Association request" the following "Information items" text just after "ESSID":

- Requester's capabilities

Insert in 5.7.2 under "Association response" the following "Information items" text just after the entry beginning "If the association...":

- Responder's capabilities

5.7.3 Reassociation

Insert in 5.7.3 under "Reassociation request" the following "Information items" text just after "ESSID":

- Requester's capabilities

Insert in 5.7.3 under "Reassociation response" the following "Information items" text just after the entry beginning "If the reassociation...":

- Responder's capabilities

End of changes to Clause 5.

6. MAC service definition

6.1 Overview of MAC services

Change the heading and text of 6.1.1 as follows:

6.1.1 ~~Asynchronous d~~Data service

This service provides peer LLC entities with the ability to exchange MSDUs. To support this service, the local MAC uses the underlying PHY-level services to transport an MSDU to a peer MAC entity, where it will be delivered to the peer LLC. Such asynchronous MSDU transport is performed on a ~~best-effort~~ connectionless basis. By default, MSDU transport is on a best-effort basis. However, the QoS facility uses a traffic identifier (TID) to specify differentiated services on a per-MSDU basis. The QoS facility also permits more synchronous behavior to be supported on a connection-oriented basis using TSPECs. There are no guarantees that the submitted MSDU will be delivered successfully. Broadcast and multicast transport is part of the~~ asynchronous~~ data service provided by the MAC. Due to the characteristics of the wireless medium (WM), broadcast and multicast MSDUs may experience a lower QoS, compared to that of unicast MSDUs. All STAs will support the ~~asynchronous~~ data service, but only QSTAs in a QBSS differentiate their MSDU delivery according to the designated traffic category or traffic stream (TS) of individual MSDUs.

Because operation of certain functions of the MAC may cause reordering of some MSDUs, as discussed in more detail below, in nQSTAs, there are two service classes within the ~~asynchronous~~ data service. By selecting the desired service class, each LLC entity initiating the transfer of MSDUs is able to control whether MAC entities are or are not allowed to reorder those MSDUs.

There are two service classes available in a QSTA: QoSAck and QoSNoAck. The service classes are used to signal if the MSDU is to be transmitted with or without using the MAC-level acknowledgment.

In QSTAs either associated in a QBSS or having membership in a QIBSS, the MAC uses a set of rules that tends to cause higher UP MSDUs in a BSS to be sent before lower UP MSDUs in the BSS. The MAC sublayer entities determine the UPs for MSDUs based on the TID values provided with those MSDUs. If a TSPEC has been provided for a TS, via the MAC sublayer management entity, the MAC attempts to deliver MSDUs belonging to that TS in accordance with the QoS parameter values contained in the TSPEC. In a QBSS with some QSTAs supporting the QoS facility and others not supporting the QoS facility, in delivering an MSDU to an nQSTA, the QSTA uses the access category (AC) corresponding to the UP of the MSDU.

Insert after 6.1.1 the following subclauses (6.1.1.1 through 6.1.1.3):

6.1.1.1 Determination of UP

The QoS facility supports eight priority values, referred to as *UPs*. The values a UP may take are the integer values from 0 to 7 and are identical to the IEEE 802.1D priority tags. An MSDU with a particular UP is said to belong to a traffic category (TC) with that UP. The UP is provided with each MSDU at the medium access control service access point (MAC_SAP) either directly, in the UP parameter, or indirectly, in a TSPEC designated by the UP parameter.

6.1.1.1.1 Determination of UP of received frames at the QAP sent by other STAs in the BSS

The received unicast frames at the QAP may be as follows:

a) Non-QoS subtypes, in which case the QAP shall assign to them a priority of Contention, if they are received during the contention period (CP), or ContentionFree, if they are received during the contention-free period (CFP).

b) QoS subtypes, in which case the QAP shall infer the UP value from the TID in the QoS Control field directly for TID values between 0 and 7. For TID values between 8 and 15, the QAP shall extract the UP value in the UP subfield of the TS Info field in the associated TSPEC or from the UP field in the associated TCLAS (traffic classification) element, as applicable.

QAPs deliver the UP with the received MSDUs to the DS.

6.1.1.2 Interpretation of priority parameter in MAC service primitives

The value of the priority parameter in the MAC service primitives (see 6.2) may be a non-integer value of either Contention or ContentionFree or may be any integer value in the range 0 through 15.

When the priority parameter has an integer value, it is used in the TID subfields that appear in certain frames that are used to deliver and to control the delivery of QoS data across the WM.

Priority parameter and TID subfield values 0 through 7 are interpreted as UPs for the MSDUs. Outgoing MSDUs with UP values 0 through 7 are handled by MAC entities at QSTAs in accordance with the UP.

Priority parameter and TID subfield values 8 through 15 specify TIDs that are also TS identifiers (TSIDs) and select the TSPEC for the TS designated by the TID. Outgoing MSDUs with priority parameter values 8 through 15 are handled by MAC entities at QSTAs in accordance with the UP value determined from the UP subfield as well as other parameter values in the selected TSPEC. When an MSDU arrives with a priority value between 8 and 15 and for which there is no TSPEC defined, then the MSDU will be sent with priority parameter set to 0.

The non-integer values of the priority parameter are allowed at all nQSTAs. The use of priority value of ContentionFree is deprecated at QSTAs. The integer values of the priority parameter (i.e., TID) are supported only at QSTAs that are either associated in a QBSS or members of a QIBSS. A range of 0 through 15 is supported by QSTAs associated in a QBSS; whereas a range of 0 through 7 is supported by QSTAs that are members of a QIBSS. If a QSTA is associated in an nQBSS, the QSTA is functioning as an nQSTA, so the priority value is always Contention or ContentionFree.

At QSTAs associated in a QBSS, MSDUs with a priority of Contention are considered equivalent to MSDUs with TID 0, and those with a priority of ContentionFree are delivered using the contention-free delivery if a point coordinator (PC) is present in the QAP. If a PC is not present, MSDUs with a priority of ContentionFree shall be delivered using an UP of 0. At QSTAs associated in an nQBSS, all MSDUs with an integer priority are considered equivalent to MSDUs with a priority of Contention.

If a QSTA is associated in a QBSS, the MSDUs it receives in QoS data frames are reported with the TID value contained in the MAC header of that frame. The MSDUs such a QSTA receives in non-QoS data frames are reported to LLC with a priority of Contention, if they are received during the CP, or Contention-Free, if they are received during the CFP.

6.1.1.3 Interpretation of service class parameter in MAC service primitives in a QSTA

In QSTAs, the value of the service class parameter in the MAC service primitive (see 6.2) may be a non-integer value of QoSAck or QoSNoAck.

When an MSDU is received from the MAC_SAP and the recipient STA is a QSTA with the service class set to

— QoSAck, the MSDU is transmitted using a QoS data frame with the Ack Policy subfield in the QoS Control field set to either Normal Acknowledgment (Normal Ack) or Block Ack.

— QoSNoAck, the MSDU is transmitted using a QoS data frame with the Ack Policy subfield in the QoS Control field set to No Acknowledgment (No Ack). If the sender QSTA is a QAP and the frame has a multicast/broadcast DA, then the MSDU is buffered for transmission and is also sent to the DS.

When an MSDU is received from the MAC_SAP and the recipient STA is not a QSTA, the MSDU is transmitted using a non-QoS data frame.

When a QoS data frame is received from another QSTA, the service class parameter in MA-UNITDATA.indication primitive is set to

— QoSAck, if the frame is a QoS data frame with the Ack Policy subfield in the QoS Control field set to either Normal Ack or Block Ack.

— QoSNoAck, if the frame is a QoS data frame with the Ack Policy subfield in the QoS Control field set to No Ack. This service class is also used where the DA parameter is a broadcast/multicast address.

When a non-QoS data frame is received from an nQSTA, the service class parameter in MA-UNITDATA.indication primitive is set to

— QoSAck, if the frame is a unicast frame and is acknowledged by the QSTA.

— QoSNoAck, if the frame is a broadcast/multicast frame and is not acknowledged by the QSTA.

Note that the broadcast/multicast frames sent by nQSTA are not acknowledged regardless of the service class parameter in MA-UNITDATA.indication primitive.

6.1.3 MSDU ordering

Change the text of 6.1.3 as follows:

The services provided by the MAC sublayer permit, and may in certain cases require, the reordering of MSDUs.

In an nQSTA, t~~T~~he MAC does not intentionally reorder MSDUs except as may be necessary to improve the likelihood of successful delivery based on the current operational (power management) mode of the designated recipient STA(s). The sole effect of this reordering (if any), for the set of MSDUs received at the MAC service interface of any single STA, is a change in the delivery order of broadcast and multicast MSDUs, relative to directed MSDUs, originating from a single source STA address. If a higher layer protocol using the ~~asynchronous~~ data service cannot tolerate this possible reordering, the optional StrictlyOrdered service class should be used. MSDUs transferred between any pair of STAs using the StrictlyOrdered service class are not subject to the relative reordering that is possible when the ReorderableMulticast service class is used. However, the desire to receive MSDUs sent using the StrictlyOrdered service class at a STA precludes simultaneous use of the MAC power management facilities at that STA.

In QSTAs operating in a QBSS, there are two service classes, designated as QoSAck and QoSNoAck (see 6.1.1.3 for more information). The MSDUs are reordered, not only to improve the likelihood of successful delivery based on the current operational mode of the designated recipient STA(s), but also to honor the priority parameters, specified in the MA-UNITDATA.request primitive, of the individual MSDUs. The effects of this reordering (if any), for the set of MSDUs received at the MAC service interface of any single STA, are

a) A change in the delivery order of broadcast and multicast MSDUs, relative to unicast MSDUs,

b) The reordering of MSDUs with different TID values, originating from a single source STA address, and

c) The reordering of broadcast and multicast MSDUs with the same TID but different service classes.

There shall be no reordering of unicast MSDUs with the same TID value and addressed to the same destination.

QSTAs operating in an nQBSS shall follow the reordering rules as defined for an nQSTA.

In order for the MAC to operate properly, the DS must meet the requirements of ISO/IEC 15802-1:1995.

Operational restrictions that ensure the appropriate ordering of MSDUs are specified in 9.8.

6.1.4 MAC data service architecture

Change the text in 6.1.4, including Figure 11g, as shown:

The MAC data plane architecture (i.e., processes that involve transport of all or part of an MSDU) is shown in Figure 11g. During transmission, an MSDU goes through some or all of the following processes: frame delivery deferral during power save (PS) mode, sequence number assignment, fragmentation, encryption, integrity protection, and frame formatting. IEEE 802.1X may block the MSDU at the Controlled Port. At some point, the data frames that contain all or part of the MSDU are queued per AC/TS. This queuing may be at any of the three points indicated in Figure 11g.

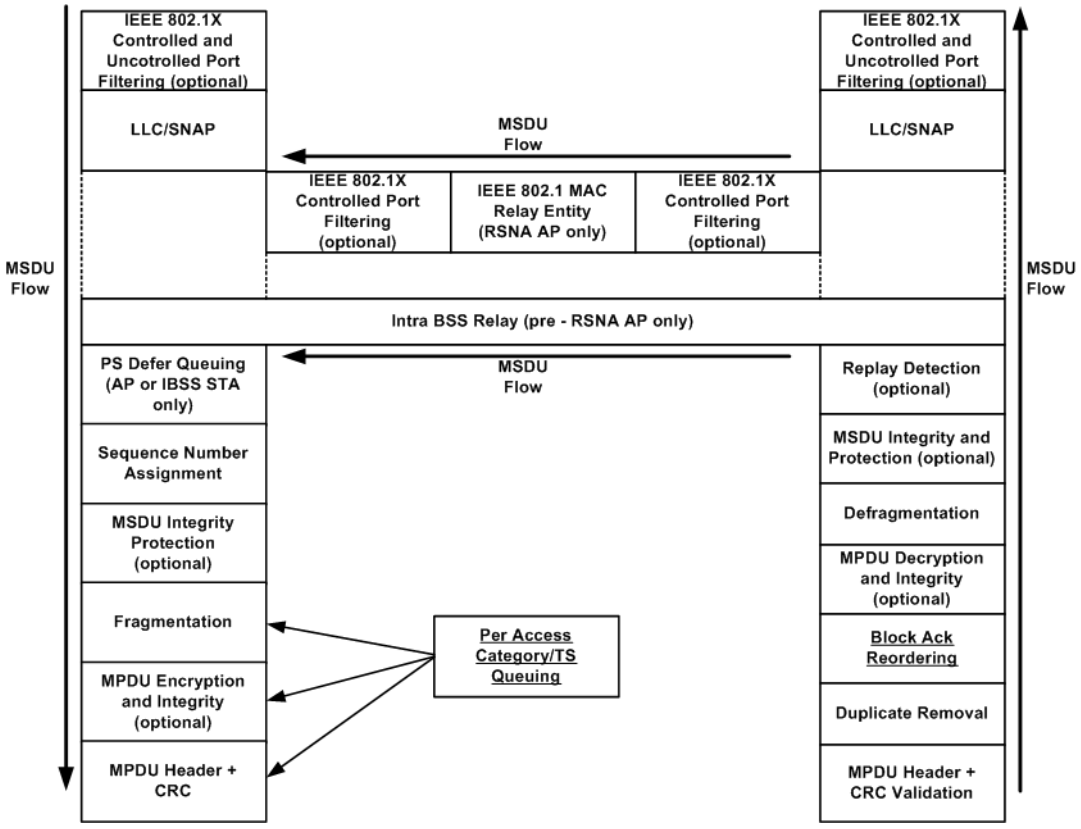


Figure 11g—MAC data plane architecture

During reception, a received data frame goes through processes of MAC protocol data unit (MPDU) header + cyclic redundancy code (CRC) validation, duplicate removal, possible reordering if the Block Ack mechanism is used, decryption, defragmentation, integrity checking, and replay detection. After replay detection (or defragmentation if security is used), the MSDU is delivered to the MAC_SAP or to the DS. The IEEE 802.1X Controlled/Uncontrolled Ports discard the MSDU if the Controlled Port is not enabled or if the MSDU does not represent an IEEE 802.1X frame. Temporal Key Integrity Protocol (TKIP) and Counter

Mode (CTR) with Cipher-Block Chaining with Message Authentication Code (CBC-MAC) Protocol (CCMP) MPDU frame order enforcement occurs after decryption, but prior to MSDU defragmentation; therefore, defragmentation will fail if MPDUs arrive out of order.

6.2 Detailed service specification

6.2.1 MAC data services

6.2.1.1 MA-UNITDATA.request

6.2.1.1.2 Semantics of the service primitive

Change the final two paragraphs of 6.2.1.1.2 as follows:

The priority parameter specifies the priority desired for the data unit transfer. ~~IEEE 802.11 allows two values: Contention or ContentionFree.~~ The allowed values of priority are described in 6.1.1.2.

The service class parameter specifies the service class desired for the data unit transfer. ~~IEEE 802.11 allows two values: ReorderableMulticast or StrictlyOrdered~~The allowed values of service class are described in 6.1.1.3 and 6.1.3.

6.2.1.1.4 Effect of receipt

Change the text of 6.2.1.1.4 as follows:

On~~The~~ receipt of this primitive, ~~causes~~ the MAC sublayer entity determines whether the request can be fulfilled according to the requested parameters. A request that cannot be fulfilled according to the requested parameters is discarded, and this action is indicated to the LLC sublayer entity using an MA-UNITDATA-STATUS.indication primitive that describes why the MAC was unable to fulfill the request. If the request can be fulfilled according to the requested parameters, the MAC sublayer entity ~~to~~ appends all MAC specified fields~~,~~ (including DA, SA, FCS, and all fields that are unique to IEEE 802.11), ~~and~~ passes the properly formatted frame to the lower layers for transfer to a peer MAC sublayer entity or entities (see 6.1.4), and indicates this action to the LLC sublayer entity using an MA-UNITDATA-STATUS.indication primitive with transmission status set to Successful.

6.2.1.2 MA-UNITDATA.indication

6.2.1.2.2 Semantics of the service primitive

Change the final three paragraphs of 6.2.1.2.2 as follows:

The reception status parameter indicates the success or failure of the received frame for those frames that IEEE 802.11 reports via a MA-UNITDATA.indication primitive. This MAC only reports "success" ~~when~~ because all failures of reception are discarded without generating MA-UNITDATA.indication primitive.

The priority parameter specifies the receive processing priority that was used for the data unit transfer. ~~IEEE 802.11 allows two values: Contention or ContentionFree.~~The allowed values of priority are described in 6.1.1.2.

The service class parameter specifies the receive service class that was used for the data unit transfer. I~~EEE 802.11 allows two values: ReorderableMulticast or StrictlyOrdered~~The allowed values of service class are described in 6.1.1.3 and 6.1.3.

6.2.1.3 MA-UNITDATA-STATUS.indication

6.2.1.3.2 Semantics of the service primitive

Change the final three paragraphs, including the lettered list, in 6.2.1.3.2 as follows:

The transmission status parameter ~~will be~~ is used to pass status information back to the local requesting LLC sublayer entity. IEEE 802.11 specifies the following values for transmission status:

a) Successful;

b) Undeliverable (for unacknowledged directed MSDUs when the ~~a~~dot11ShortRetryLimit~~Max~~ or ~~a~~dot11LongRetryLimit~~Max~~ retry limit would otherwise be exceeded);

c) Undeliverable (excessive data length);

d) Undeliverable (non-null source routing);

e) Undeliverable: unsupported priority (for priorities other than Contention or ContentionFree at an nQSTA; or for priorities other than Contention, ContentionFree, or an integer between and including 0 and 15 at a QSTA);

f) Undeliverable: unsupported service class (for service classes other than ReorderableMulticast or StrictlyOrdered~~)~~ for nQSTAs and service classes other than QoSAck or QoSNoAck for QSTAs);

g) Unavailable priority (for ContentionFree when no PC or HC is available, or an integer between and including 1 and 15 at a QSTA that is associated in an nQBSS, or an integer between and including 8 and 15 at a QSTA that is a member of an IBSS, in which case the MSDU is transmitted with a provided priority of Contention);

h) Undeliverable: unavailable service class (for StrictlyOrdered service when the STA's power management mode is other than "active" for nQSTAs; QSTAs do not return this value as they do not provide the StrictlyOrdered service);

i) Undeliverable (TransmitMSDUTimer reached ~~a~~dot11MaxTransmitMSDULifetime before successful delivery);

j) Undeliverable (no BSS available);

k) Undeliverable (cannot encrypt with a null key);

l) Undeliverable (MSDU transmit timer reached the value in dot11EDCATableMSDULifetime or dot11QAPEDCATableMSDULifetime prior to a successful delivery or the TransmitMSDUTimer reached the delay bound prior to successful delivery);

m) Undeliverable (Block Ack timeout value has been exceeded).

The provided priority parameter specifies the priority that was used for the associated data unit transfer (Contention, ~~or~~ ContentionFree, or an integer between and including 0 and 15).

The provided service class parameter specifies the class of service used for the associated data unit transfer. In nQSTAs, the value of this parameter is ~~(~~ReorderableMulticast or StrictlyOrdered~~)~~. In QSTAs, it is QoSAck or QoSNoAck.

End of changes to Clause 6.

7. Frame formats

Change the introductory text of Clause 7 as follows:

The format of the MAC frames is specified in this clause. ~~All stations~~A STA shall be able to properly construct a subset of the frames specified in this subclause for transmission and to decode a (potentially

different) subset of the frames specified in this subclause upon validation following reception~~, as specified in this clause~~. The particular subset of these frames that a STA constructs and decodes is determined by the functions supported by that particular STA, as specified in 7.5. All STAs shall be able to validate every received frame using the frame check sequence (FCS) and to interpret certain fields from the MAC headers of all frames.

7.1 MAC frame formats

Change the text of 7.1 as follows:

Each frame consists of the following basic components:

a) A *MAC header,* which comprises frame control, duration, address, ~~and~~ sequence control information, and, for QoS data frames, QoS control information;

b) A variable length *frame body*, which contains information specific to the frame *type* and *subtype*;

c) A *frame check sequence* (FCS), which contains an IEEE 32-bit CRC.

7.1.1 Conventions

Insert in 7.1.1 the following text before the final paragraph:

Reception, in references to frames or fields within frames (e.g., received beacon frames or a received Duration/ID field), applies to MPDUs or MAC management protocol data units (MMPDUs) indicated from the PHY layer without error and validated by FCS within the MAC sublayer. Without further qualification, *reception* by the MAC sublayer implies that the frame contents are valid, and that the protocol version is supported (see 7.1.3.1.1), with no implication regarding frame addressing or regarding whether the frame type or other fields in the MAC header are meaningful to the MAC entity that has received the frame.

Parentheses enclosing portions of names or acronyms are used to designate a set of related names that vary based on the inclusion of the parenthesized portion. For example,

— *QoS +CF-Poll frame* refers to the three QoS data subtypes that include "+CF-Poll": the QoS Data+CF-Poll frame, subtype 1010; QoS Data+CF-Ack+CF-Poll frame, subtype 1011; and QoS CF-Ack+CF-Poll frame, subtype 1111.

— *QoS CF-Poll frame* refers specifically to the QoS CF-Poll frame, subtype 1110.

— *QoS (+)CF-Poll frame* refers to all four QoS data subtypes with CF-Poll: the QoS CF-Poll frame, subtype 1110; the QoS CF-Ack+CF-Poll frame, subtype 1111; the QoS Data+CF-Poll frame, subtype 1010; and the QoS Data+CF-Ack+CF-Poll frame, subtype 1011.

— *QoS (+)Null frame* refers to all three QoS data subtypes with "no data": the QoS Null (no data) frame, subtype 1100; the QoS CF-Poll (no data) frame, subtype 1110; and the QoS CF-Ack+CF-Poll frame, subtype 1111.

— *QoS +CF-Ack frame* refers to the three QoS data subtypes that include "+CF-Ack": the QoS Data+CF-Ack frame, subtype 1001; QoS Data+CF-Ack+CF-Poll frame, subtype 1011; and QoS CF-Ack+CF-Poll frame, subtype 1111.

— Whereas *(QoS) CF-Poll frame* refers to the QoS CF-Poll frame, subtype 1110, and the CF-Poll frame, subtype 0110.

7.1.2 General frame format

Change the text in 7.1.2, including Figure 12, as follows:

The MAC frame format comprises a set of fields that occur in a fixed order in all frames. Figure 12 depicts the general MAC frame format. The first three fields (Frame Control, Duration/ID, and Address 1) and the last field (FCS) in Figure 12 constitute the minimal frame format and are present in all frames, including reserved types and subtypes. The fields Address 2, Address 3, Sequence Control, Address 4, QoS Control, and Frame Body are present only in certain frame types and subtypes. Each field is defined in 7.1.3. The format of each of the individual subtypes of each frame types is defined in 7.2. The components of management frame bodies are defined in 7.3. The formats of management frames of subtype Action are defined in 7.4

The Frame Body field is of variable size. The maximum frame body size is determined by the maximum MSDU size (2304 octets) plus any overhead from security encapsulation.

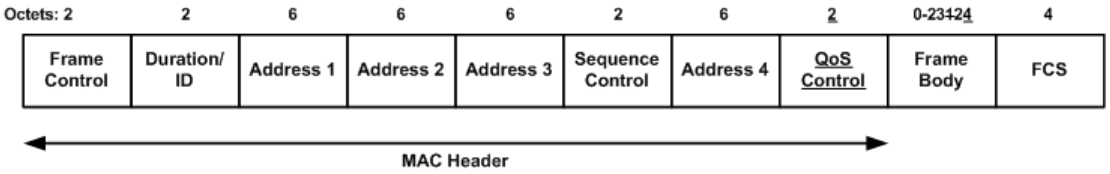


Figure 12—MAC frame format

7.1.3 Frame fields

7.1.3.1 Frame Control field

7.1.3.1.2 Type and Subtype fields

Change the text of 7.1.3.1.2, including Table 1, as shown:

The Type field is 2 bits in length, and the Subtype field is 4 bits in length. The Type and Subtype fields together identify the function of the frame. There are three frame types: control, data, and management. Each of the frame types have several defined subtypes. In data frames, the most significant bit (MSB) of the Subtype field, b7, is defined as the QoS subfield. Table 1 defines the valid combinations of type and subtype. (The numeric values in Table 1 are shown in binary.)

Table 1—Valid type and subtype combinations

Type value b3 b2	Type description	Subtype value b7 b6 b5 b4	Subtype description
00	Management	0000	Association request
00	Management	0001	Association response
00	Management	0010	Reassociation request
00	Management	0011	Reassociation response
00	Management	0100	Probe request
00	Management	0101	Probe response
00	Management	0110–0111	Reserved
00	Management	1000	Beacon

Table 1—Valid type and subtype combinations *(continued)*

Type value b3 b2	Type description	Subtype value b7 b6 b5 b4	Subtype description
00	Management	1001	Announcement traffic indication message (ATIM)
00	Management	1010	Disassociation
00	Management	1011	Authentication
00	Management	1100	Deauthentication
00	Management	1101	Action
00	Management	1110–1111	Reserved
01	Control	0000–~~10~~0111	Reserved
01	Control	1000	Block Ack Request (BlockAckReq)
01	Control	1001	Block Ack (BlockAck)
01	Control	1010	Power Save Poll (PS-Poll)
01	Control	1011	Request To Send (RTS)
01	Control	1100	Clear To Send (CTS)
01	Control	1101	Acknowledgment (ACK)
01	Control	1110	Contention-Free (CF)-End
01	Control	1111	CF-End + CF-Ack
10	Data	0000	Data
10	Data	0001	Data + CF-Ack
10	Data	0010	Data + CF-Poll
10	Data	0011	Data + CF-Ack + CF-Poll
10	Data	0100	Null ~~function~~ (no data)
10	Data	0101	CF-Ack (no data)
10	Data	0110	CF-Poll (no data)
10	Data	0111	CF-Ack + CF-Poll (no data)
10	Data	1000–~~1111~~	QoS Data~~Reserved~~
10	Data	1001	QoS Data + CF-Ack
10	Data	1010	QoS Data + CF-Poll
10	Data	1011	QoS Data + CF-Ack + CF-Poll
10	Data	1100	QoS Null (no data)
10	Data	1101	Reserved
10	Data	1110	QoS CF-Poll (no data)
10	Data	1111	QoS CF-Ack + CF-Poll (no data)
11	Reserved	0000–1111	Reserved

Each Subtype field bit position is used to indicate a specific modification of the basic data frame (subtype 0). Frame Control bit 4 is set to 1 in data subtypes that include +CF-Ack, bit 5 is set to 1 in data subtypes which include +CF-Poll, bit 6 is set to 1 in data subtypes that contain no Frame Body field, and bit 7 is set to 1 in the QoS data subtypes, which have QoS Control fields in their MAC headers.

7.1.3.1.3 To DS field

Change the text in 7.1.3.1.3 as follows:

The To DS field is one bit in length and is set to 1 in data ~~type~~ frames destined for the DS. This includes all data ~~type~~ frames sent to an AP, including all data frames sent by nQSTAs associated with an AP and all data frames being sent between APs using the WM as a wireless distribution system (WDS). The To DS field is set to 0 in all other frames. For additional details, see Table 2.

7.1.3.1.4 From DS field

Change the text in 7.1.3.1.4, including Table 2, as follows:

The From DS field is 1 bit in length and is set to 1 in data ~~type~~ frames exiting the DS. This includes all data frames sent by APs, including frames sent using the WM as a WDS. ~~It~~The From DS field is set to 0 in all other frames.

The permitted To/From DS bit combinations and their meanings are given in Table 2.

Table 2—To/From DS combinations in data ~~type~~ frames

To/From DS values	Meaning
To DS = 0 From DS = 0	A data frame direct from one STA to another STA within the same IBSS, or a data frame direct from one non-AP QSTA to another non-AP QSTA within the same QBSS, as well as all management and control ~~type~~ frames.
To DS = 1 From DS = 0	~~Data frame destined for the DS.~~A data frame from a STA to the DS via an AP.
To DS = 0 From DS = 1	~~Data frame exiting the DS.~~A data frame from the DS to a STA via an AP.
To DS = 1 From DS = 1	WDS data frame being distributed from one AP to another AP via the WM.

7.1.3.1.8 More Data field

Insert the following paragraph at the end of 7.1.3.1.8:

For a non-AP QSTA that has the More Data Ack subfield set in its QoS Capability information element and also has APSD enabled, a QAP may set the More Data field to 1 in ACK frames to this non-AP QSTA to indicate that the QAP has a pending transmission for the non-AP QSTA.

7.1.3.1.10 Order field

Change the text of 7.1.3.1.10 as shown:

The Order field is 1 bit in length and is set to 1 in any non-QoS data ~~type~~ frame that contains an MSDU, or fragment thereof, which is being transferred using the StrictlyOrdered service class. This field is set to 0 in all other frames. All QSTAs set this subfield to 0.

7.1.3.2 Duration/ID field

Change the text of 7.1.3.2, including Table 3 (note that the order of the left three columns has been reversed), as follows:

The Duration/ID field is 16 bits in length. The contents of this field ~~are~~ vary with frame type and subtype, with whether the frame is transmitted during the CFP, and with the QoS capabilities of the sending STA. The contents of the field are defined as follows:

a) In control ~~type~~ frames of subtype Power Save (PS)-Poll the Duration/ID field carries the association identity (AID) of the STA that transmitted the frame in the 14 least significant bits (LSB), ~~with~~and the 2 most significant bits (MSB) both set to 1. The value of the AID is in the range 1–2007.

b) In frames transmitted by the PC and nQSTAs, during the CFP, the Duration/ID field is set to a fixed value of 32 768.

c) In all other frames sent by nQSTAs and control frames sent by QSTAs, the Duration/ID field contains a duration value as defined for each frame type in 7.2.

d) In data and management frames sent by QSTAs, the Duration/ID field contains a duration value as defined for each frame type in 7.1.4. ~~For frames transmitted during the contention-free period (CFP), the duration field is set to 32768.~~

When~~ever~~ the contents of a received Duration/ID field, treated as an unsigned integer and without regard for address values, type, and subtype (even when type or subtype contain reserved values), are less than 32 768, the duration value is used to update the network allocation vector (NAV) according to the procedures defined in ~~Clause~~ 9.2.5.4 or 9.9.2.2.1, as appropriate.

When the contents of a received Duration/ID field, treated as an unsigned integer, are greater than 32 768, the contents are interpreted as appropriate for the frame type and subtype or ignored if the receiving MAC entity does not have a defined interpretation for that type and subtype.

The encoding of the Duration/ID field is given in Table 3.

Table 3—Duration/ID field encoding

Bit 0–13	Bit 14	Bits 15	Usage
0–32767		0	Duration value (in microseconds) within all frames other than PS-Poll frames transmitted during the CP, and under HCF for frames transmitted during the CFP.
0	0	1	Fixed value under point coordination function (PCF) within frames transmitted during the CFP
1–16 383	0	1	Reserved
0	1	1	Reserved
1–2 007	1	1	AID in PS-Poll frames
2 008–16 383	1	1	Reserved

7.1.3.4 Sequence Control field

Insert the following sentence at the end of the text in 7.1.3.4:

Sequence Control field is not present in control frames.

7.1.3.4.1 Sequence Number field

Change the text of 7.1.3.4.1 as shown:

The Sequence Number field is a 12-bit field indicating the sequence number of an MSDU or MMPDU. Each MSDU or MMPDU transmitted by a STA is assigned a sequence number. Sequence numbers are not assigned to control frames, as the Sequence Control field is not present.

~~Sequence numbers are~~nQSTAs, as well as QSTAs operating as nQSTAs because they are in an nQBSS or nQIBSS, assign~~ed~~ sequence numbers, to management frames and data frames (QoS subfield is set to 0), from a single modulo-4096 counter, starting at 0 and incrementing by 1 for each MSDU or MMPDU.

QSTAs associated in a QBSS maintain one modulo-4096 counter, per TID, per unique receiver (specified by the Address 1 field of the MAC header). Sequence numbers for QoS data frames are assigned using the counter identified by the TID subfield of the QoS Control field of the frame, and that counter is incremented by 1 for each MSDU belonging to that TID. Sequence numbers for management frames, QoS data frames with a broadcast/multicast address in the Address 1 field, and all non-QoS data frames sent by QSTAs are assigned using an additional single modulo-4096 counter, starting at 0 and incrementing by 1 for each MSDU or MMPDU. Sequence numbers for QoS (+)Null frames may be set to any value.

Each fragment of an MSDU or MMPDU contains the assigned sequence number. The sequence number remains constant in all retransmissions of an MSDU, MMPDU, or fragment thereof.

Insert after 7.1.3.4.2 the following new subclauses as 7.1.3.5 through 7.1.3.5.7, including the new figures in those subclauses (instructions for renumbering the existing 7.1.3.5 are given after this new text):

7.1.3.5 QoS Control field

The QoS Control field is a 16-bit field that identifies the TC or TS to which the frame belongs and various other QoS-related information about the frame that varies by frame type and subtype. The QoS Control field is present in all data frames in which the QoS subfield is set to 1 (see 7.1.3.1.2). Each QoS Control field comprises five subfields, as defined for the particular sender (HC or non-AP QSTA) and frame type and subtype. The usage of these subfields and the various possible layouts of the QoS Control field are described 7.1.3.5.1 through 7.1.3.5.7 and illustrated in Table 3a.

Table 3a—QoS Control field

Applicable frame (sub) types	Bits 0–3	Bit 4	Bit 5-6	Bit 7	Bits 8–15
QoS (+)CF-Poll frames sent by HC	TID	EOSP	Ack Policy	Reserved	TXOP Limit
QoS Data, QoS Null, and QoS Data+CF-Ack frames sent by HC	TID	EOSP	Ack Policy	Reserved	QAP PS Buffer State
QoS data frames sent by non-AP QSTAs	TID	0	Ack Policy	Reserved	TXOP Duration Requested
	TID	1	Ack Policy	Reserved	Queue Size

7.1.3.5.1 TID subfield

The TID subfield identifies the TC or TS to which the corresponding MSDU, or fragment thereof, in the Frame Body field belongs. The TID subfield also identifies the TC or TS of traffic for which a TXOP is

being requested, through the setting of TXOP duration requested or queue size. The encoding of the TID subfield depends on the access policy (see 7.3.2.28) and is shown in Table 3b. Additional information on the interpretation of the contents of this field appears in 6.1.1.2.[3]

Table 3b—TID subfield

Access policy	Usage	Allowed values in bits 0–3 (TID subfield)
EDCA	UP for either TC or TS, regardless of whether admission control is required	0–7
HCCA	TSID	8–15
HEMM	TSID, regardless of the access mechanism used	8–15

For QoS Data+CF-Poll, the TID subfield in the QoS Control field indicates the TID of the data. For all QoS (+)CF-Poll frames of subtype Null, the TID subfield in the QoS Control field indicates the TID for which the poll is intended. The requirement to respond to that TID is nonbinding, and a QSTA may respond with any frame.

7.1.3.5.2 EOSP (end of service period) subfield

The EOSP subfield is 1 bit in length and is used by the HC to indicate the end of the current service period (SP). The HC sets the EOSP subfield to 1 in its transmission and retransmissions of the SP's final frame to end a scheduled/unscheduled SP and sets it to 0 otherwise.

7.1.3.5.3 Ack Policy subfield

The Ack Policy subfield is 2 bits in length and identifies the acknowledgement policy that is followed upon the delivery of the MPDU. The interpretation of these 2 bits is given in Table 3c.

Table 3c—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. The addressed recipient returns an ACK or QoS +CF-Ack frame after a short inter-frame space (SIFS) period, according to the procedures defined in 9.2.8, 9.3.3 and 9.9.2.3. The Ack Policy subfield is set to this value in all directed frames in which the sender requires acknowledgment. For QoS Null (no data) frames, this is the only permissible value for the Ack Policy subfield.
1	0	No Ack The addressed recipient takes no action upon receipt of the frame. More details 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 broadcast and multicast frames that use the QoS frame format.

[3]The presence of the TID subfield allows an IEEE 802.11 QBSS to function as a LAN that is able to signal the priority.

Table 3c—Ack Policy subfield in QoS Control field of QoS data frames *(continued)*

Bits in QoS Control field		Meaning
Bit 5	Bit 6	
0	1	No explicit acknowledgment. There may be a response frame to the frame that is received, but it is neither the ACK nor any data frame of subtype +CF-Ack. For QoS CF-Poll and QoS CF-Ack+CF-Poll data frames, this is the only permissible value for the Ack Policy subfield.
1	1	Block Ack The addressed recipient takes no action upon the receipt of the frame except for recording the state. The recipient can expect a BlockAckReq frame in the future to which it responds using the procedure described in 9.10.

An MSDU is sent using an acknowledgment policy of Normal Ack or Block Ack if the service class parameter in MA-UNITDATA.request primitive is set to QoSAck and of No Ack if the service class parameter in MA-UNITDATA.request primitive is set to QoSNoAck.

7.1.3.5.4 TXOP Limit subfield

The TXOP Limit subfield is an 8-bit field that is present in QoS data frames of subtypes that include CF-Poll and specifies the time limit on a TXOP granted by a QoS (+)CF-Poll frame from an HC in a QBSS. In QoS data frames with subtypes that include CF-Poll, the addressed QSTA is granted a TXOP that begins a SIFS period after this frame and lasts no longer than the number of 32 μs periods specified by the TXOP limit value. The range of time values is 32 μs to 8160 μs. A TXOP limit value of 0 implies that one MPDU or one QoS Null frame is to be transmitted immediately following the QoS (+)CF-Poll frame. The TXOP limit is inclusive of the PHY and IFS overhead, and a QAP should account for the overhead when granting TXOPs.

7.1.3.5.5 Queue Size subfield

The Queue Size subfield is an 8-bit field that indicates the amount of buffered traffic for a given TC or TS at the non-AP QSTA sending this frame. The Queue Size subfield is present in QoS data frames sent by STAs associated in a QBSS with bit 4 of the QoS Control field set to 1. The QAP may use information contained in the Queue Size subfield to determine the TXOP duration assigned to non-AP QSTA.

The queue size value is the total size, rounded up to the nearest multiple of 256 octets and expressed in units of 256 octets, of all MSDUs buffered at the QSTA (excluding the MSDU of the present QoS data frame) in the delivery queue used for MSDUs with TID values equal to the value in the TID subfield of this QoS Control field. A queue size value of 0 is used solely to indicate the absence of any buffered traffic in the queue used for the specified TID. A queue size value of 254 is used for all sizes greater than 64 768 octets. A queue size value of 255 is used to indicate an unspecified or unknown size. If a QoS data frame is fragmented, the queue size value may remain constant in all fragments even if the amount of queued traffic changes as successive fragments are transmitted.

7.1.3.5.6 TXOP Duration Requested subfield

The TXOP Duration Requested subfield is an 8-bit field that indicates the duration, in units of 32 μs, that the sending STA desires for its next TXOP for the specified TID. The range of time values is 32 μs to 8160 μs. If the calculated TXOP duration requested is not a factor of 32 μs, that value is rounded up to the next higher integer that is a factor of 32 μs. The TXOP Duration Requested subfield is present in QoS data frames sent by non-AP QSTAs associated in a QBSS with bit 4 of the QoS Control field set to 0. The QAP may choose

to assign a TXOP duration shorter than that requested in the TXOP Duration Requested subfield. A value of 0 in the TXOP Duration Requested subfield indicates that no TXOP is requested for the MSDUs for the specified TID in the current SP.

TXOP Duration Requested subfield values are not cumulative. A TXOP duration requested for a particular TID supersedes any prior TXOP duration requested for that TID. A value of 0 in the TXOP Duration Requested subfield may be used to cancel a pending unsatisfied TXOP request when its MSDU is no longer queued for transmission. The TXOP duration requested is inclusive of the PHY and IFS overhead, and a QSTA should account for this when attempting to determine whether a given transmission fits within a specified TXOP duration.

7.1.3.5.7 QAP PS Buffer State subfield

The QAP PS Buffer State subfield, defined in Figure 14a, is an 8-bit field that indicates the PS buffer state at the QAP for a non-AP QSTA. The QAP PS Buffer State subfield is further subdivided into three subfields: Buffer State Indicated, Highest-Priority Buffered AC, and QAP Buffered Load.

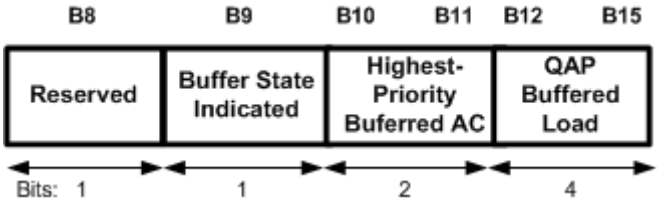


Figure 14a—QAP PS Buffer State subfield

The Buffered State Indicated subfield is 1 bit in length and is used to indicate whether the QAP PS buffer state is specified. A value of 1 indicates that the QAP PS buffer state is specified.

The Highest-Priority Buffered AC subfield is 2 bits in length and is used to indicate the AC of the highest priority traffic remaining that is buffered at the QAP, excluding the MSDU of the present frame.

The QAP Buffered Load subfield is 4 bits in length and is used to indicate the total buffer size, rounded up to the nearest multiple of 4096 octets and expressed in units of 4096 octets, of all MSDUs buffered at the QAP (excluding the MSDU of the present QoS data frame). A QAP Buffered Load field value of 15 indicates that the buffer size is greater than 57 344 octets. A QAP Buffered Load subfield value of 0 is used solely to indicate the absence of any buffered traffic for the indicated highest priority buffered AC when the Buffer State Indicated bit is 1.

When the Buffered State Indicated subfield is set to 0, the Highest-Priority Buffered AC subfield and the QAP Buffered Load subfield are reserved; and the values of these subfields are either unspecified or unknown.

Change the following subclause numbers due to the insertion above of the new subclause, 7.1.3.5:

7.1.3.6 ~~7.1.3.5~~ Frame Body field

7.1.3.7 ~~7.1.3.6~~ FCS field

Insert after 7.1.3.7 the following new subclause (7.1.4):

7.1.4 Duration/ID field in data and management frames

Within all data frames containing QoS CF-Poll, the Duration/ID field value is set to one of the following:

— One SIFS duration plus the TXOP limit, if the TXOP limit is nonzero, or

— The time required for the transmission of one MPDU of nominal MSDU size and the associated ACK frame plus two SIFS intervals, if the TXOP limit is zero.

Within all data or management frames sent in a CP by the QSTAs outside of a controlled access phase (CAP), following a contention access of the channel, the Duration/ID field is set to one of the following values:

a) For management frames, frames with QoS Data subfield set to 0, and unicast data frames with Ack Policy subfield set to Normal Ack,
   1) The time required for the transmission of one ACK frame (including appropriate IFS values), if the frame is the final fragment of the TXOP, or
   2) The time required for the transmission of one ACK frame plus the time required for the transmission of the following MPDU and its response if required (including appropriate IFS values).

b) For unicast data frames with the Ack Policy subfield set to No Ack or Block Ack and for multicast/broadcast frames,
   1) Zero, if the frame is the final fragment of the TXOP, or
   2) The time required for the transmission of the following MPDU and its response frame, if required (including appropriate IFS values).

c) The minimum of
   1) The time required for the transmission the pending MPDUs of the AC and the associated ACKs, if any, and applicable SIFS durations, and
   2) The time limit imposed by the MIB attribute dot11EDCATableTXOPLimit (dot11EDCAQAPTableTXOPLimit for the QAP) for that AC minus the already used time within the TXOP.

Within all data or management frames sent under HCCA, to ensure NAV protection for the entire CAP, the Duration/ID field is set to one of the following values:

— The remaining duration of the TXOP, if the frame is a nonfinal frame in a TXOP with multiple frame exchanges.

— The actual remaining time needed for this frame exchange sequence, if the frame is the sole or final frame in the TXOP.

7.2 Format of individual frame types

7.2.1 Control frames

7.2.1.1 Request to Send (RTS) frame format

Change the final paragraph of 7.2.1.1 as shown:

For all RTS frames sent by nQSTAs, t~~T~~he duration value is the time, in microseconds, required to transmit the pending data or management frame, plus one Clear to Send (CTS) frame, plus one ACK frame, plus three SIFS intervals. If the calculated duration includes a fractional microsecond, that value is rounded up to the next higher integer. For all RTS frames sent by QSTAs under EDCA, following a contention access of the channel, the duration value is set in the following manner:

- If the NAV protection is desired for only the first or sole frame in the TXOP, the duration value is set to the time, in microseconds, required to transmit the pending frame, plus one CTS frame, plus one ACK frame if required, plus three SIFS intervals.
- Otherwise, the duration value is set to the remaining duration of the TXOP.

For all RTS frames sent under HCCA, the duration value is set to one of the following values:

- If the pending frame is the final frame, the duration value is set to the time, in microseconds, required to transmit the pending frame, plus one CTS frame, plus one ACK frame if required, plus three SIFS intervals.
- If the pending frame is not the final frame in the TXOP, the duration value is set to the remaining duration of the TXOP.

7.2.1.2 Clear to Send (CTS) frame format

Change the final two paragraphs of 7.2.1.2 as shown:

For all CTS frames sent in response to RTS frames, t~~T~~he duration value is the value obtained from the Duration field of the immediately previous RTS frame, minus the time, in microseconds, required to transmit the CTS frame and its SIFS interval. If the calculated duration includes a fractional microsecond, that value is rounded up to the next higher integer.

At an nQSTA, i~~I~~f the CTS is the first frame in the exchange and the pending data or management frame requires acknowledgment, the duration value is the time, in microseconds, required to transmit the pending data or management frame, plus one SIFS interval, plus one ACK frame, and an additional SIFS interval. At an nQSTA, i~~I~~f the CTS is the first frame in the exchange and the pending data or management frame does not require acknowledgment, the duration value is the time, in microseconds, required to transmit the pending data or management frame, plus one SIFS interval. If the calculated duration includes a fractional microsecond, that value is rounded up to the next higher integer.

For all CTS frames sent by QSTAs as the first frame in the exchange under EDCA, the duration value is set in the following manner:

- If the NAV protection is desired for only the first or sole frame in the TXOP the duration value is set to
  - The time, in microseconds, required to transmit the pending frame, plus one SIFS interval, plus the response frame (ACK or Block Ack), plus an additional SIFS interval, if there is a response frame, or
  - The time, in microseconds, required to transmit the pending frame, plus one SIFS interval, if there is no response frame.
- Otherwise, the duration value is set to the remaining duration of the TXOP.

For CTS frames sent under HCCA, the duration value is set to one of the following values:

— If the pending frame is the sole frame in the TXOP, the duration value is set to
  — The time, in microseconds, required to transmit the pending frame, plus one SIFS interval, plus the response frame (ACK or Block Ack), plus an additional SIFS interval, if there is a response frame, or
  — The time, in microseconds, required to transmit the pending frame, plus one SIFS interval, if there is no response frame.
— If the pending frame is not the final frame in the TXOP, the duration value is set to the remaining duration of the TXOP.

7.2.1.3 Acknowledgment (ACK) frame format

Change the final two paragraphs of 7.2.1.3 as shown:

The RA field of the ACK frame is copied from the Address 2 field of the immediately previous directed data, management, BlockAckReq control, BlockAck control, or PS-Poll control frame.

For ACK frames sent by nQSTAs, i~~I~~f the More Fragments bit was set to 0 in the Frame Control field of the immediately previous directed data or management frame, the duration value is set to 0. ~~If the More Fragments bit was set to 1 in the Frame Control field of the immediately previous directed data or management frame,~~In all other ACK frames, the duration value is the value obtained from the Duration/ID field of the immediately previous data, ~~or~~ management, PS-Poll, BlockAckReq, or BlockAck frame minus the time, in microseconds, required to transmit the ACK frame and its SIFS interval. If the calculated duration includes a fractional microsecond, that value is rounded up to the next higher integer.

Insert after 7.2.1.6 the following new subclauses (7.2.1.7 and 7.2.1.8), including the new figures in those subclauses:

7.2.1.7 Block Ack Request (BlockAckReq) frame format

The frame format of the BlockAckReq frame is defined in Figure 21a.

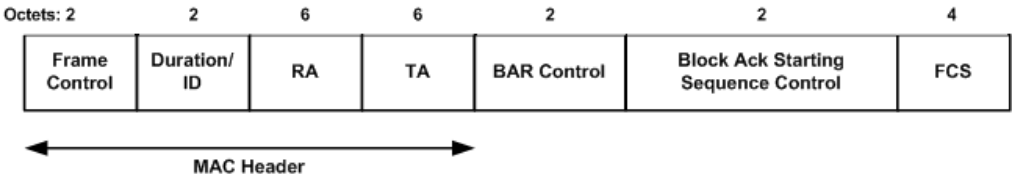


Figure 21a—BlockAckReq frame

The Duration/ID field value is greater than or equal to the time[4], in microseconds, required to transmit one ACK or BlockAck frame, as applicable, plus one SIFS interval. If the calculated duration includes a fractional microsecond, that value is rounded up to the next higher integer.

The RA field of the BlockAckReq frame is the address of the recipient QSTA.

The TA field is the address of the QSTA transmitting the BlockAckReq frame.

[4]To allow the possibility of time remaining in the TXOP, which the sender may use to schedule other transmissions.

The BAR Control field is shown in Figure 21b.

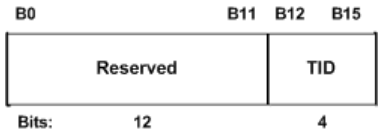


Figure 21b—BAR Control field

The TID subfield of the BAR Control field contains the TID for which a BlockAck frame is requested.

The Block Ack Starting Sequence Control field is shown in Figure 21c. The Starting Sequence number subfield is the sequence number of the first MSDU for which this BlockAckReq is sent. The Fragment Number subfield is always set to 0.

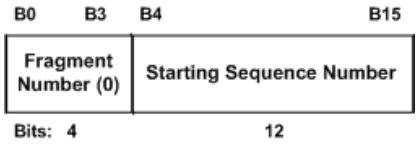


Figure 21c—Block Ack Starting Sequence Control field

7.2.1.8 Block Ack (BlockAck) frame format

The frame format of the BlockAck frame is defined in Figure 21d.

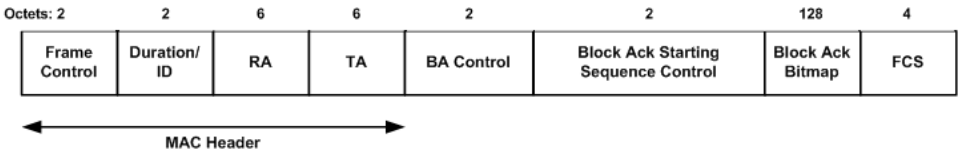


Figure 21d—BlockAck frame

If the BlockAck frame is sent in response to the BlockAckReq frame, the Duration/ID field value is the value obtained from the Duration/ID field of the immediate BlockAckReq frame, minus the time, in microseconds, required to transmit the BlockAck frame and its SIFS interval. If the BlockAck frame is not sent in response to the BlockAckReq, the Duration/ID field value is greater than (subject to the TXOP limit) or equal to the time[5] for transmission of an ACK frame plus a SIFS interval. If the calculated duration includes a fractional microsecond, that value is rounded to the next higher integer.

The RA field of the BlockAck frame is the address of the recipient QSTA that requested the Block Ack.

The TA field is the address of the QSTA transmitting the BlockAck frame.

[5]See Footnote 4.

The BA Control field defined in Figure 21e consists of the TID subfield.

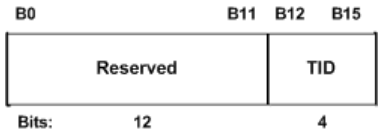


Figure 21e—BA Control field

The Block Ack Starting Sequence Control field is defined in 7.2.1.7 and is set to the same value as in the immediately previously received BlockAckReq frame.

The Block Ack Bitmap field is 128 octets in length and is used to indicate the receiving status of up to 64 MSDUs. Bit position *n* of the Block Ack bitmap, if set to 1, acknowledges receipt of an MPDU with an MPDU sequence control value equal to (Block Ack Starting Sequence Control + *n*). Bit position n of the Block Ack bitmap, if set to 0, indicates that an MPDU with MPDU sequence control value equal to (Block Ack Starting Sequence Control + n) has not been received. For unused fragment numbers of an MSDU, the corresponding bits in the bitmap are set to 0.

7.2.2 Data frames

Change the text of 7.2.2, including Figure 22 and Table 4, as shown:

The frame format for a data frame is ~~in~~dependent on the QoS subfield of the Subtype field and is as defined in Figure 22.

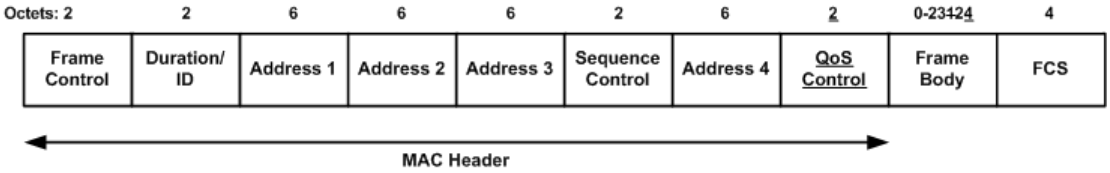


Figure 22—Data frame

Data frames with a value of 1 in the QoS subfield of the Subtype field are collectively referred to as *QoS data frames*. Each of these data subtypes contains QoS in their names, and this frame format is distinguished by the presence of a QoS Control field in the MAC header. Data frames with a value of 0 in the QoS subfield of the Subtype field do not have the QoS Control field.

A QSTA always uses QoS data frames for data transmissions to other QSTAs. A QSTA uses frames with the QoS subfield of the Subtype field set to 0 for data transmissions to nQSTAs. An nQSTA always uses frames with the QoS subfield of the Subtype field set to 0 for data transmissions to other STAs. All STAs use frames with the QoS subfield of the Subtype field set to 0 for broadcast data frames unless a transmitting QSTA knows that all STAs in a QBSS have QoS capability, in which case the transmitting QSTAs uses QoS data frames. All STAs use frames with the QoS subfield of the Subtype field set to 0 for multicast data frames unless it is known to the transmitter that all STAs in the QBSS that are members of the multicast group have QoS capability, in which case QSTAs use QoS data frames.

The content of the address fields of data frames are dependent upon the values of the To DS and From DS bits in the Frame Control field and are defined in Table 4. Where the content of a field is shown as not applicable (N/A), the field is omitted. Note that Address 1 always holds the receiver address of the intended

receiver (or, in the case of multicast frames, receivers), and that Address 2 always holds the address of the STA that is transmitting the frame.

Table 4—Address field contents

To DS	From DS	Address 1	Address 2	Address 3	Address 4	Description
0	0	DA	SA	BSSID	N/A	STA to STA traffic in an IBSS and QSTA-to-QSTA traffic in a QBSS
0	1	DA	BSSID	SA	N/A	AP-to-STA traffic in a BSS
1	0	BSSID	SA	DA	N/A	STA-to-AP traffic in a BSS
1	1	RA	TA	DA	SA	WDS traffic between APs

A STA uses the contents of the Address 1 field to perform address matching for receive decisions. In cases where the Address 1 field contains a group address, the BSSID also is validated to ensure that the broadcast or multicast originated in the same BSS.

A STA uses the contents of the Address 2 field to direct the acknowledgment if an acknowledgment is necessary.

The DA is the destination of the MSDU (or fragment thereof) in the Frame Body field.

The SA is the address of the MAC entity that initiated the MSDU (or fragment thereof) in the Frame Body field.

The RA is the address of the STA contained in the AP in the WDS that is the next immediate intended recipient of the frame.

The TA is the address of the STA contained in the AP in the WDS that is transmitting the frame.

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.

The Sequence Control field is defined in 7.1.3.4. The Sequence Control field for QoS (+)Null frames is ignored by the receiver upon reception.

The QoS Control field is defined in 7.1.3.5.

The frame body consists of the MSDU, or a fragment thereof, and a security header and trailer (if and only if the Protected Frame subfield in the Frame Control field is set to 1). The frame body is null (0 octets in length) in data frames of subtype Null ~~function~~ (no data), CF-Ack (no data), CF-Poll (no data), and CF-Ack+CF-Poll (no data), regardless of the encoding of the QoS subfield in the Frame Control field.

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 omitted; 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 and for the corresponding four QoS data frame subtypes, the Frame Body field contains all of, or a fragment of, an MSDU after any encapsulation for security.

The maximum length of the Frame Body field can be determined from the maximum MSDU length 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).

Within all data frames sent by STAs during the CFP under point coordination function (PCF), the Duration/ ID field is set to 32 768. Within all data frames sent by the QSTA, the Duration/ID field contains a duration value as defined in 7.1.4. Within all data ~~type~~ frames sent during the CP by nQSTAs, the Duration/ID field is set according to the following rules:

— If the Address 1 field contains a group address, the duration value is set to 0.

— If the More Fragments bit is set to 0 in the Frame Control field of a frame and the Address 1 field contains an individual address, the duration value is set to the time, in microseconds, required to transmit one ACK frame, plus one SIFS interval.

— If the More Fragments bit is set to 1 in the Frame Control field of a frame and the Address 1 field contains an individual address, the duration value is set to the time, in microseconds, required to transmit the next fragment of this data frame, plus two ACK frames, plus three SIFS intervals.

The duration value calculation for the data frame is based on the rules in 9.6 that determine the data rate at which the control frames in the frame exchange sequence are transmitted. If the calculated duration includes a fractional microsecond, that value is rounded up to the next higher integer. All STAs process Duration/ID field values less than or equal to 32 767 from valid data frames to update their NAV settings as appropriate under the coordination function rules.

7.2.3 Management frames

Change the text in 7.2.3 as shown:

The frame format for management frames is independent of frame subtype and is as defined in Figure 23.

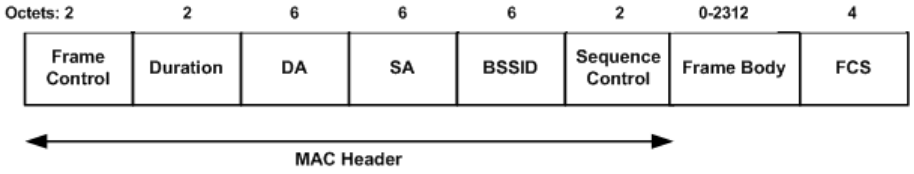


Figure 23—Management frame format

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 type is other than Beacon, the BSSID also is validated to ensure that the broadcast or multicast originated in the same BSS. If the frame type is Beacon, other address matching rules apply, as specified in 11.1.2.3.

The address fields for management frames do not vary by frame subtype.

The BSSID of the management 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) In management frames of subtype Probe Request, the BSSID is either a specific BSSID or the broadcast BSSID as defined in the procedures specified in ~~Clause 10~~11.1.3.2.

The DA is the destination of the frame.

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

Within all management ~~type~~ frames sent by STAs during the CFP under PCF, the Duration field is set to the value 32 768. Within all management frames sent by the QSTA, the Duration field contains a duration value as defined in 7.1.4. Within all management ~~type~~ frames sent during the CP by nQSTAs, the Duration field is set according to the following rules:

— If the DA field contains a group address, the duration value is set to 0.

— If the More Fragments bit is set to 0 in the Frame Control field of a frame and the DA field contains an individual address, the duration value is set to the time, in microseconds, required to transmit one ACK frame, plus one SIFS interval.

— If the More Fragments bit is set to 1 in the Frame Control field of a frame, and the DA field contains an individual address, the duration value is the time, in microseconds, required to transmit the next fragment of this management frame, plus two ACK frames, plus three SIFS intervals.

The duration value calculation for the management frame is based on the rules in 9.6 that determine the data rate at which the control frames in the frame exchange sequence are transmitted. If the calculated duration includes a fractional microsecond, that value is rounded up to the next higher integer. All STAs process Duration field values less than or equal to 32 767 from valid management frames to update their NAV settings as appropriate under the coordination function rules.

The frame body consists of the fixed fields ~~and~~ followed by the information elements defined for each management frame subtype. All fixed fields and information elements are mandatory unless stated otherwise~~,~~ and ~~they can~~ only appear in the specified, relative order. STAs that encounter~~ing~~ an element ~~type~~ID they do not ~~understand~~recognize in the frame body of a received management frame ignore that element and continue to parse the remainder of the management frame body (if any) for additional information elements with recognizable element IDs. ~~Element type codes not explicitly defined in this standard are reserved and do not appear in any frames.~~ Unused element ID codes are reserved.

Gaps may exist in the ordering of fixed fields and elements within frames. The order that remains shall be ascending.

7.2.3.1 Beacon frame format

Change the text in 7.2.3.1 as follows:

The frame body of a management frame of subtype Beacon contains the information shown in Table 5. If the dot11MultiDomainCapabilityEnabled attribute is true, a STA shall include a Country Information element in the transmission of Beacon frames. Optionally, the Beacon frame format may also include the information described in either or both of FH Parameters ~~orders 12~~ and FH Pattern Table elements~~13~~. If ~~the information in~~ both FH Parameters ~~orders 12~~ and FH Pattern Table elements~~13~~ are sent, they shall describe the same hopping pattern. Note that the information described in FH Parameters and FH Pattern Table elements~~orders 12 and 13~~ may be also contained in the Probe Response frame.

Insert in 7.2.3.1 the following rows in Table 5:

Table 5—Beacon frame body

Order	Information	Notes
22	QBSS Load	The QBSS Load information element is only present within Beacon frames generated by QAPs. The QBSS Load element is present when dot11QosOptionImplemented and dot11QBSSLoadImplemented are both true.
23	EDCA Parameter Set	The EDCA Parameter Set information element is only present within Beacon frames generated by QAPs. The EDCA Parameter Set element is present when dot11QosOptionImplemented is true and the QoS Capability element is not present.
24	QoS Capability	The QoS Capability information element is only present within Beacon frames generated by QAPs. The QoS Capability element is present when dot11QosOptionImplemented is true and EDCA Parameter Set element is not present.

7.2.3.4 Association Request frame format

Insert in 7.2.3.4 the following rows in Table 7:

Table 7—Association Request frame body

Order	Information
9	QoS Capability

7.2.3.5 Association Response frame format

Insert in 7.2.3.5 the following rows in Table 8:

Table 8—Association Response frame body

Order	Information
6	EDCA Parameter Set

7.2.3.6 Reassociation Request frame format

Insert in 7.2.3.6 the following rows in Table 9:

Table 9—Reassociation Request frame body

Order	Information
10	QoS Capability

7.2.3.7 Reassociation Response frame format

Insert in 7.2.3.7 the following rows in Table 10:

Table 10—Reassociation Response frame body

Order	Information
6	EDCA Parameter Set

7.2.3.9 Probe Response frame format

Insert in 7.2.3.9 the following rows in Table 12:

Table 12—Probe Response frame body

Order	Information	Notes
23	QBSS Load	The QBSS Load element is present when dot11QosOptionImplemented and dot11QBSSLoadImplemented are both true. The QBSS Load information element is always present within Probe Response frames generated by QAPs when dot11QBSSLoadImplemented is true.
24	EDCA Parameter Set	The EDCA Parameter Set element is present when dot11QosOptionImplemented is true. The EDCA Parameter Set information element is always present within Probe Response frames generated by QAPs.

7.3 Management frame body components

Change the text in 7.3 as shown:

Within management frames except in Action frames, fixed-length mandatory frame body components are defined as fixed fields; variable length mandatory and all optional frame body components are defined as information elements. Action frames allow variable fields as explained in 7.4.

7.3.1 Fixed fields

7.3.1.4 Capability Information field

Change the text in 7.3.1.4, including Figure 27, Table 16, and Table 17, as shown:

The Capability Information field contains a number of subfields that are used to indicate requested or advertised optional capabilities.

The length of the Capability Information field is 2 octets. ~~The Capability Information field consists of the following subfields: ESS, IBSS, CF-Pollable, CF-Poll Request, Privacy, Short Preamble, PBCC, Channel Agility, Spectrum Management, Short Slot Time, and DSSS-OFDM. The remaining bits in the Capability Information field are reserved.~~ The format of the Capability Information field is defined~~as illustrated~~ in Figure 27. No subfield is supplied for ERP as a STA supports ERP operation if it includes all the Clause 19 mandatory rates in its supported rate set.

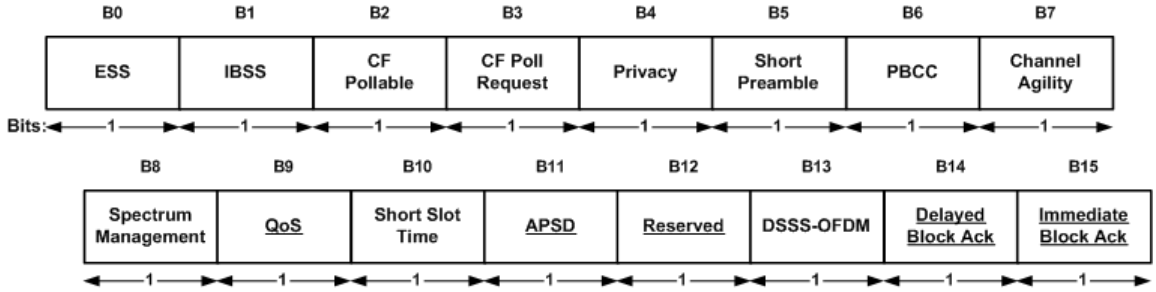


Figure 27—Capability Information fixed field

Each Capability Information subfield is interpreted according to~~only in~~ the management frame subtype~~s for which the transmission rules are~~ as defined in this subclause.

APs set the ESS subfield to 1 and the IBSS subfield to 0 within transmitted Beacon or Probe Response management frames. STAs within an IBSS set the ESS subfield to 0 and the IBSS subfield to 1 in transmitted Beacon or Probe Response management frames.

STAs set the QoS, CF-Pollable, and CF-Poll Request subfields in Association and Reassociation Request management frames according to Table 16.

Table 16—STA usage of QoS, CF-Pollable, and CF-Poll Request

QoS	CF-Pollable	CF-Poll Request	Meaning
0	0	0	STA is not CF-Pollable
0	0	1	STA is CF-Pollable, not requesting to be placed on the CF-Polling list
0	1	0	STA is CF-Pollable, requesting to be placed on the CF-Polling list
0	1	1	STA is CF-Pollable, requesting never to be polled
1	0	0	QSTA requesting association in a QBSS.
1	0	1	Reserved
1	1	0	Reserved
1	1	1	Reserved

APs set the CF-Pollable and CF-Poll Request subfields in Beacon~~,~~ and Probe Response~~, Association Response, and Reassociation Response~~ management frames according to Table 17. An nQAP sets the CF-Pollable and CF-Poll Request subfield values in Association Response and Reassociation Response management frames equal to the values in the last Beacon or Probe Response frame that it transmitted.

Table 17—AP usage of QoS, CF-Pollable, and CF-Poll Request

QoS	CF-Pollable	CF-Poll Request	Meaning
0	0	0	No PC at nQAP
0	0	1	PC at nQAP for delivery only (no polling)
0	1	0	PC at nQAP for delivery and polling
0	1	1	Reserved
1	0	0	QAP (HC) does not use CFP for delivery of unicast data frames

Table 17—AP usage of QoS, CF-Pollable, and CF-Poll Request *(continued)*

QoS	CF-Pollable	CF-Poll Request	Meaning
1	0	1	QAP (HC) uses CFP for delivery, but does not send CF-Polls to nQSTAs
1	1	0	QAP (HC) uses CFP for delivery, and sends CF-Polls to nQSTAs
1	1	1	Reserved

APs set the Privacy subfield to 1 within transmitted Beacon, Probe Response, Association Response and Reassociation Response management frames if data confidentiality is required for all data ~~type~~ frames exchanged within the BSS. If data confidentiality is not required, the Privacy subfield is set to 0.

In an RSNA, non-AP STAs in an ESS set the Privacy subfield to 0 within transmitted Association and Reassociation Request management frames. APs ignore the Privacy subfield within received Association and Reassociation Request management frames.

QSTAs within an ESS set the Privacy subfield to 1 in DLS Request and DLS Response frames if encryption is required for all data frames exchanged. If encryption is not required, the Privacy subfield is set to 0.

STAs within an IBSS set the Privacy subfield to 1 in transmitted Beacon or Probe Response management frames if data confidentiality is required for all data ~~type~~ frames exchanged within the IBSS. If data confidentiality is not required, STAs in an IBSSS set the Privacy subfield to 0 within these management frames.

STAs that include the RSN information element in Beacon and Probe Response frames shall set the Privacy subfield to 1 in any frame that includes the RSN information element.

APs (as well as STAs in IBSSs) shall set the Short Preamble subfield to 1 in transmitted Beacon, Probe Response, Association Response and Reassociation Response ~~management~~ MMPDUs to indicate that the use of the short preamble, as described in 18.2.2.2, is allowed within this BSS. To indicate that the use of the short preamble is not allowed the Short Preamble subfield shall be set to 0 in Beacon, Probe Response, Association Response, and Reassociation Response ~~management~~ MMPDUs transmitted within the BSS.

ERP STAs shall set the MIB variable dot11ShortPreambleOptionImplemented to true as all ERP ~~STAs~~ devices support both long and short preamble formats.

STAs shall set the Short Preamble subfield to 1 in transmitted Association Request and Reassociation Request ~~MMPDU~~management frames and in DLS Request and DLS Response frames when the MIB attribute dot11ShortPreambleOptionImplemented is true. Otherwise, STAs shall set the Short Preamble subfield to 0 ~~in transmitted Association Request and Reassociation Request MMPDUs~~.

APs (as well as STAs in IBSSs) shall set the PBCC subfield to 1 in transmitted Beacon, Probe Response, Association Response, and Reassociation Response management ~~MMPDU~~frames to indicate that the PBCC modulation option, as described in 18.4.6.6 and 19.6, is allowed within this BSS. To indicate that the PBCC modulation option is not allowed, the PBCC subfield shall be set to 0 ~~in Beacon, Probe Response, Association Response, and Reassociation Response management MMPDUs transmitted within the BSS~~.

STAs shall set the PBCC subfield to 1 in transmitted Association Request, Reassociation Request, ~~MMPDUs~~ DLS Request, and DLS Response frames when the MIB attribute dot11PBCCOptionImplemented is true. Otherwise, STAs shall set the PBCC subfield to 0 ~~in transmitted Association Request and Reassociation Request MMPDUs~~.

Bit 7 of the Capabilities Information field shall be used to indicate the usage of Channel Agility by the High Rate direct sequence spread spectrum (HR/DSSS) PHY or ERP. STAs shall set the Channel Agility bit to 1 when Channel Agility is in use and shall set it to 0 otherwise.

A STA shall set the Spectrum Management subfield in the Capability Information field to 1 if the STA's dot11SpectrumManagementRequired is true; otherwise, it shall be set to 0.

STAs set the QoS subfield to 1 within the Capability Information field when the MIB attribute dot11QosOptionImplemented is true and set it to 0 otherwise.

STAs shall set the Short Slot Time subfield to 1 in transmitted Association Request, ~~and~~ Reassociation Request, DLS Request, and DLS Response MMPDUs when the MIB attributes dot11ShortSlotTimeOptionImplemented and dot11ShortSlotTimeOptionEnabled are true. Otherwise, the STA shall set the Short Slot Time subfield to 0 in transmitted Association Request and Reassociation Request MMPDUs.

If a STA that does not support Short Slot Time associates, the AP shall use long slot time beginning at the first Beacon subsequent to the association of the long slot time STA. APs shall set the Short Slot Time subfield in transmitted Beacon, Probe Response, Association Response, and Reassociation Response MMPDUs to indicate the currently used slot time value within this BSS.

STAs shall set the MAC variable aSlotTime to the short slot value upon transmission or reception of Beacon, Probe Response, Association Response, and Reassociation Response MMPDUs from the BSS that the STA has joined or started and that have the short slot subfield set to 1 when the MIB attribute dot11ShortSlotTimeOptionImplemented is true. STAs shall set the MAC variable aSlotTime to the long slot value upon transmission or reception of Beacon, Probe Response, Association Response, and Reassociation Response MMPDUs from the BSS that the STA has joined or started and that have the short slot subfield set to 0 when the MIB attribute dot11ShortSlotTimeOptionImplemented is true. STAs shall set the MAC variable aSlotTime to the long slot value at all times when the MIB attribute dot11ShortSlotTimeOptionImplemented is false. When the dot11ShortSlotTimeOptionImplemented MIB attribute is not present, or when the PHY supports only a single slot time value, then the STA shall set the MAC variable aSlotTime to the slot value appropriate for the attached PHY.

For IBSS, the Short Slot Time subfield shall be set to 0.

QAPs set the APSD subfield to 1 within the Capability Information field when the MIB attribute dot11APSDOptionImplemented is true and set it to 0 otherwise. STAs always set this subfield to 0.

QSTAs set the Immediate Block Ack subfield to 1 within the Capability Information field when the MIB attribute dot11ImmediateBlockAckOptionImplemented is true and set it to 0 otherwise.

APs as well as STAs in IBSSs shall set the DSSS-OFDM subfield to 1 in transmitted Beacon, Probe Response, Association Response, and Reassociation Response management MMPDUs to indicate that the use of direct sequence spread spectrum with orthogonal frequency division multiplexing (DSSS-OFDM), as described in 19.7, is allowed within this BSS or by STAs that want to use DSSS-OFDM within an IBSS. To indicate that the use of DSSS-OFDM is not allowed, the DSSS-OFDM subfield shall be set to 0 in Beacon, Probe Response, Association Response, and Reassociation Response MMPDUs transmitted within the BSS.

STAs shall set the DSSS-OFDM subfield to 1 in transmitted Association Request, ~~and~~ Reassociation Request, DLS Request, and DLS Response MMPDUs when the MIB attributes dot11DSSS-OFDM-OptionImplemented and dot11DSSS-OFDMOptionEnabled are true. Otherwise, STAs shall set the DSSS-OFDM subfield to 0 in transmitted Association Request and Reassociation Request MMPDUs.

QSTAs set the Delayed Block Ack subfield to 1 within the Capability Information field when the MIB attribute dot11DelayedBlockAckOptionImplemented is true and set it to 0 otherwise.

Unused bits of the Capability Information field are reserved.

7.3.1.7 Reason Code field

Change the first paragraph in 7.3.1.7 as shown:

This Reason Code field is used to indicate the reason that an unsolicited notification management frame of type Disassociation, ~~or~~ Deauthentication, DELTS, DELBA, or DLS Teardown was generated. The length of the Reason Code field is 2 octets. The Reason Code field is illustrated in Figure 30.

Insert the following reason codes in Table 18 as shown:

Table 18—Reason codes

Reason code	Meaning
25–31	Reserved
32	Disassociated for unspecified, QoS-related reason
33	Disassociated because QAP lacks sufficient bandwidth for this QSTA
34	Disassociated because excessive number of frames need to be acknowledged, but are not acknowledged due to AP transmissions and/or poor channel conditions
35	Disassociated because QSTA is transmitting outside the limits of its TXOPs
36	Requested from peer QSTA as the QSTA is leaving the QBSS (or resetting)
37	Requested from peer QSTA as it does not want to use the mechanism
38	Requested from peer QSTA as the QSTA received frames using the mechanism for which a setup is required
39	Requested from peer QSTA due to timeout
45	Peer QSTA does not support the requested cipher suite
46–65 535	Reserved

7.3.1.9 Status Code field

Insert the following status codes in Table 19 as shown:

Table 19—Status codes

Status code	Meaning
27–31	Reserved
32	Unspecified, QoS-related failure
33	Association denied because QAP has insufficient bandwidth to handle another QSTA

Table 19—Status codes *(continued)*

Status code	Meaning
34	Association denied due to excessive frame loss rates and/or poor conditions on current operating channel
35	Association (with QBSS) denied because the requesting STA does not support the QoS facility
37	The request has been declined
38	The request has not been successful as one or more parameters have invalid values
39	The TS has not been created because the request cannot be honored; however, a suggested TSPEC is provided so that the initiating QSTA may attempt to set another TS with the suggested changes to the TSPEC
47	The TS has not been created; however, the HC may be capable of creating a TS, in response to a request, after the time indicated in the TS Delay element
48	Direct link is not allowed in the BSS by policy
49	The Destination STA is not present within this QBSS
50	The Destination STA is not a QSTA
51–65 535	Reserved

7.3.1.11 Action field

Change the second paragraph in 7.3.1.11 as shown:

The Category field ~~shall be~~ is set to one of the nonreserved values shown in Table 19a. Action frames of a given category are referred to as *<category name> Action frames*. For example, frames in the QoS category are called *QoS Action frames*.

If a STA receives a unicast Action frame with an unrecognized Category field or some other syntactic error and the MSB of the Category field set equal to 0, then the STA returns the Action frame to the source without change except that the MSB of the Category field ~~shall be~~ is set equal to 1.

Change Table 19a in 7.3.1.11, including reversing the order of the left two columns, as shown:

Table 19a—Category values

~~Value~~ Code	~~Name~~Meaning	See subclause
0	Spectrum management	7.4.1
1	QoS	7.4.2
2	DLS	7.4.3
3	Block Ack	7.4.4
4–127	Reserved	—
128–255	Error	—

Insert the following sentence at the end of 7.3.1.11:

Details on the additional fixed fields used within the Action field are given in 7.3.2.12 through 7.2.1.17.

Insert after 7.3.1.11 the following new subclauses (7.3.1.12 through 7.3.1.17), including the figures and table in those subclauses:

7.3.1.12 Dialog Token field

The Dialog Token field is used for matching action responses with action requests when there are multiple, concurrent action requests. The length of the Dialog Token field is 1 octet. The Dialog Token field is illustrated in Figure 33b. The value of the Dialog Token field in a request Action frame is arbitrary. The value of the Dialog Token field in a response frame is copied from each request Action frame.

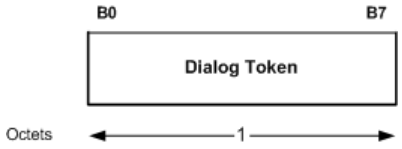


Figure 33b—Dialog Token fixed field

7.3.1.13 DLS Timeout Value field

The DLS Timeout Value field is used in the DLS Request frame to indicate the timeout value for the direct link. The length of the DLS Timeout Value field is 2 octets. The DLS Timeout Value field is illustrated in Figure 33c.

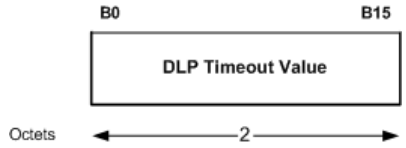


Figure 33c—DLS Timeout Value fixed field

The DLS Timeout Value field contains the duration, in seconds, after which the direct link is terminated, if there are no frame exchanges within this duration with the peer. A value of 0 implies that the direct link is never to be terminated based on a timeout.

7.3.1.14 Block Ack Parameter Set field

The Block Ack Parameter Set field is used in ADDBA frames to signal the parameters for setting up a Block Ack. The length of the Block Ack Parameter Set field is 2 octets. The Block Ack Parameter Set field is illustrated in Figure 33d.

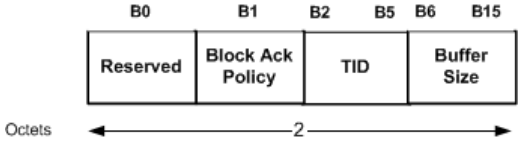


Figure 33d—Block Ack Parameter Set fixed field

The Block Ack Policy subfield is set to 1 for immediate Block Ack and 0 for delayed Block Ack. The Block Ack Policy subfield value assigned by the originator of the QoS data frames is advisory.

The TID subfield contains the value of the TC or TS for which the Block Ack is being requested.

The Buffer Size subfield indicates the number of buffers of size 2304 octets available for this particular TID.[6]

In an ADDBA Request frame, the Buffer Size subfield is intended to provide guidance for the frame receiver to decide its reordering buffer size and is advisory only. If the Buffer Size subfield is set to 0, it implies that the originator of the Block Ack has no information to specify its value.

In an ADDBA Response frame, when the Status Code field is set to 0, the Buffer Size subfield is set to a value of at least 1.

7.3.1.15 Block Ack Timeout Value field

The Block Ack Timeout Value field is used in the ADDBA Request frame to indicate the timeout value for Block Ack. The length of the Block Ack Timeout Value field is 2 octets. The Block Ack Timeout Value field is illustrated in Figure 33e.

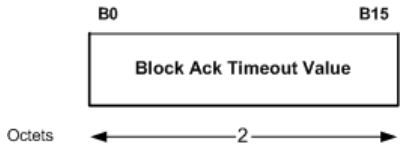


Figure 33e—Block Ack Timeout Value fixed field

The Block Ack Timeout Value field contains the duration, in TUs, after which the Block Ack setup is terminated, if there are no frame exchanges (see 11.5.3) within this duration using this Block Ack agreement. A value of 0 disables the timeout.

7.3.1.16 DELBA Parameter Set field

The DELBA Parameter Set field is used in a DELBA frame to terminate an already setup Block Ack. The length of the DELBA Parameters field is 2 octets. The DELBA Parameters field is illustrated in Figure 33f.

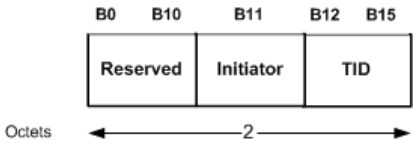


Figure 33f—DELBA Parameters fixed field

The Initiator subfield indicates if the originator or the recipient of the data is sending this frame. It is set to 1 to indicate the originator and is set to 0 to indicate the recipient. The TID subfield indicates the TSID or the UP for which the Block Ack has been originally set up.

[6]For buffer size, the recipient of data advertises a single scalar number that is the number of maximum-size fragment buffers available. Every buffered MPDU will consume one of these buffers regardless of whether the frame contains a whole MSDU or a fragment of an MSDU. In other words, ten maximum-size unfragmented MSDUs will consume the same amount of buffer space at the recipient as 10 small fragments.

7.3.1.17 QoS Info field

The QoS Info field is 1 octet in length and contains capability information bits. The contents of the field are dependent on whether the QSTA is a QAP or a non-AP QSTA.

The format of the QoS Info field, when sent by the QAP, is defined in Figure 33g.

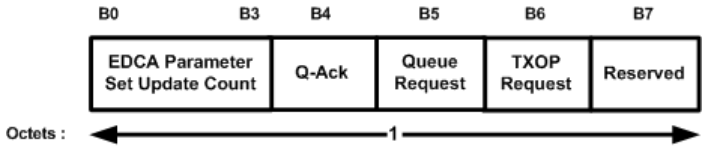


Figure 33g—QoS Info field when sent by a QAP

In the Beacon frame, the EDCA Parameter Set Update Count subfield is initially set by the QAP to 0 and is incremented every time any of the AC parameters changes.

QAPs set the Q-Ack subfield to 1 when the MIB attribute dot11QAckOptionImplemented is true and set it to 0 otherwise.

QAPs set the Queue Request subfield to 1 if they can process a nonzero Queue Size subfield in the QoS Control field in QoS data frames and set it to 0 otherwise.

QAPs set the TXOP Request subfield to 1 if they can process a nonzero TXOP Duration Requested subfield in the QoS Control field in QoS data frames and set it to 0 otherwise.

The format of the QoS Info field, when sent by the non-AP QSTA, is defined in Figure 33h.

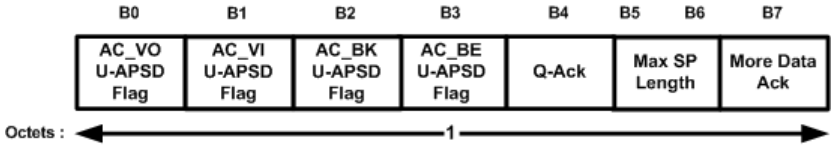


Figure 33h—QoS Info field when set by a non-AP QSTA

Each of the ACs U-APSD Flag subfields is 1 bit in length and set to 1 in (Re)Association Request frames to indicate that the corresponding AC (AC_BE, AC_BK, AC_VI, or AC_VO) is both trigger-enabled and delivery-enabled. It is set to 0 in (Re)Association Request frames to indicate that the corresponding AC is neither trigger-enabled nor delivery-enabled. A TSPEC as described in 11.2.1.4 is to be used to make a particular AC exclusively either trigger-enabled or delivery-enabled. These subfields are always set to 0 when the APSD subfield in the Capability Information field is set to 0.

Non-AP QSTAs set the Q-Ack subfield to 1 when the MIB attribute dot11QAckOptionImplemented is true and set it to 0 otherwise.

The Max SP Length subfield is 2 bits in length and indicates the maximum number of total buffered MSDUs and MMPDUs the QAP may deliver to a non-AP QSTA during any SP triggered by the non-AP QSTA. This subfield is reserved when the APSD subfield in the Capability Information field is set to 0. This subfield is also reserved when all four U-APSD flags are set to 0. If the APSD subfield in the Capability Information field is set to 1 and at least one of the four U-APSD flags is set to 1, the settings of the values in the Max SP Length subfield are defined in Table 19b.

Table 19b—Settings of the Max SP Length subfield

Bit 5	Bit 6	Usage
0	0	QAP may deliver all buffered MSDUs and MMPDUs.
1	0	QAP may deliver a maximum of two MSDUs and MMPDUs per SP.
0	1	QAP may deliver a maximum of four MSDUs and MMPDUs per SP.
1	1	QAP may deliver a maximum of six MSDUs and MMPDUs per SP.

Non-AP QSTAs set the More Data Ack subfield to 1 to indicate that they can process Ack frames with the More Data bit in the Frame Control field set to 1 and will remain in the awake state. Non-AP QSTAs set the More Data Ack subfield to 0 otherwise. For QAPs, the More Data Ack subfield is reserved.

7.3.2 Information elements

Insert the following element IDs in Table 20 in 7.3.2 as shown:

Table 20—Element IDs

Information element	Element ID
QBSS Load	11
EDCA Parameter Set	12
TSPEC	13
TCLAS	14
Schedule	15
TS Delay	43
TCLAS Processing	44
Reserved	45
QoS Capability	46
Reserved	47

Insert in 7.3.2 the following paragraph after Table 20:

A STA that encounters an unknown or reserved element ID value in a management frame received without error shall ignore that element and shall parse any remaining management frame body for additional information elements with recognizable element ID values. The frame body components specified for many management subtypes result in elements ordered by ascending element ID.

7.3.2.25 RSN information element

7.3.2.25.3 RSN capabilities

Change the sentence before Table 20dd in 7.3.2.25.3 as follows:

The number of replay counters per STAKeySA is the same as the number of replay counters per PTKSA ~~or GTKSA~~.

Insert after 7.3.2.25.4 the following new subclauses (7.3.2.26 through 7.3.2.33), including the new figures and tables in those subclauses:

7.3.2.26 QBSS Load element

The QBSS Load element contains information on the current STA population and traffic levels in the QBSS. The element information format is defined in Figure 46td. This element may be used by the non-AP QSTA for vendor-specific AP selection algorithm when roaming.

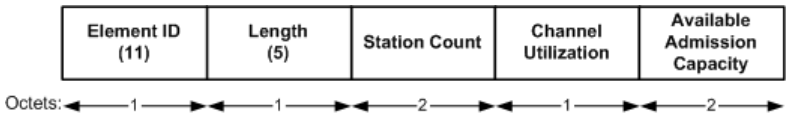


Figure 46td—QBSS Load element format

The STA Count field is interpreted as an unsigned integer that indicates the total number of STAs currently associated with this QBSS.

The Channel Utilization field is defined as the percentage of time, normalized to 255, the QAP sensed the medium was busy, as indicated by either the physical or virtual carrier sense (CS) mechanism. This percentage is computed using the formula,

((channel busy time/(dot11ChannelUtilizationBeaconIntervals * dot11BeaconPeriod * 1024)) * 255),

where *channel busy time* is defined to be the number of microseconds during which the CS mechanism, as defined in 9.2.1, has indicated a channel busy indication, and the MIB attribute dot11ChannelUtilizationBeaconIntervals represents the number of consecutive beacon intervals during which the channel busy time is measured. The default value of dot11ChannelUtilizationBeaconIntervals is defined in Annex D.

The Available Admission Capacity field is 2 octets long and contains an unsigned integer that specifies the remaining amount of medium time available via explicit admission control, in units of 32 μs/s. The field is helpful for roaming non-AP QSTAs to select a QAP that is likely to accept future admission control requests, but it does not represent a guarantee that the HC will admit these requests.

7.3.2.27 EDCA Parameter Set element

The EDCA Parameter Set element provides information needed by non-AP QSTAs for proper operation of the QoS facility during the CP. The format of the EDCA Parameter Set element is defined in Figure 46te.

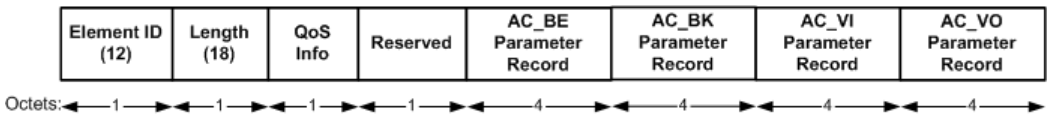


Figure 46te—EDCA Parameter Set element

The EDCA Parameter Set element is used by the QAP to establish policy (by changing default MIB attribute values), to change policies when accepting new STAs or new traffic, or to adapt to changes in offered load. The most recent EDCA parameter set element received by a non-AP QSTA is used to update the appropriate MIB values.

The format of the QoS Info field is defined in 7.3.1.17. The QoS Info field contains the EDCA Parameter Set Update Count subfield, which is initially set to 0 and is incremented each time any of the AC parameters changes. This subfield is used by non-AP QSTAs to determine whether the EDCA parameter set has changed and requires updating the appropriate MIB attributes.

The formats of AC_BE, AC_BK, AC_VI, and AC_VO Parameter Record fields are identical and are illustrated in Figure 46tf.

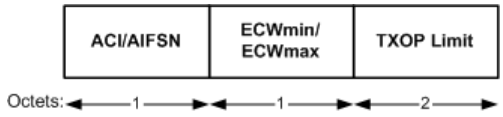


Figure 46tf—AC_BE, AC_BK, AC_VI, and AC_VO Parameter Record field format

The format of the ACI/AIFSN field is illustrated in Figure 46tg.

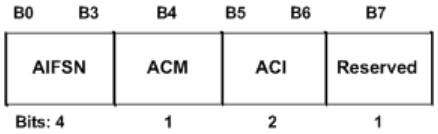


Figure 46tg—ACI/AIFSN field

The value of the AC index (ACI) references the AC to which all parameters in this record correspond. The mapping between ACI and AC is defined in Table 20de. The ACM (admission control mandatory) subfield indicates that admission control is required for the AC. If the ACM subfield is set to 0, then there is no admission control for the corresponding AC. If the ACM subfield is set to 1, admission control has to be used prior to transmission using the access parameters specified for this AC. The AIFSN subfield indicates the number of slots after a SIFS duration a non-AP QSTA should defer before either invoking a backoff or starting a transmission. The minimum value for the AIFSN subfield is 2.

Table 20de—ACI-to-AC coding

ACI	AC	Description
00	AC_BE	Best effort
01	AC_BK	Background
10	AC_VI	Video
11	AC_VO	Voice

The ECWmin and ECWmax fields are illustrated in Figure 46th.

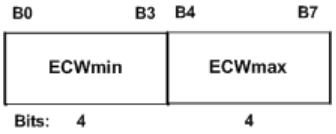


Figure 46th—ECWmin and ECWmax fields

The ECWmin and ECWmax fields encode the values of CWmin and CWmax, respectively, in an exponent form. The ECWmin and ECWmax values are defined so that

$$CWmin = 2^{ECWmin} - 1$$

$$CWmax = 2^{ECWmax} - 1$$

Hence the minimum encoded value of CWmin and CWmax is 0, and the maximum value is 32 767.

The value of the TXOP Limit field is specified as an unsigned integer, with the least significant octet transmitted first, in units of 32 μs. A TXOP Limit field value of 0 indicates that a single MSDU or MMPDU, in addition to a possible RTS/CTS exchange or CTS to itself, may be transmitted at any rate for each TXOP.

The default values used by non-AP QSTAs for the parameters in the EDCA Parameter Set element are defined in Table 20df[7].

Table 20df—Default EDCA Parameter Set element parameter values

AC	CWmin	CWmax	AIFSN	TXOP limit		
				For PHYs defined in Clause 15 and Clause 18	For PHYs defined in Clause 17 and Clause 19	Other PHYs
AC_BK	aCWmin	aCWmax	7	0	0	0
AC_BE	aCWmin	aCWmax	3	0	0	0
AC_VI	(aCWmin+1)/2 – 1	aCWmin	2	6.016 ms	3.008 ms	0
AC_VO	(aCWmin+1)/4 – 1	(aCWmin+1)/2 – 1	2	3.264 ms	1.504 ms	0

7.3.2.28 TSPEC element

The TSPEC element contains the set of parameters that define the characteristics and QoS expectations of a traffic flow, in the context of a particular non-AP QSTA, for use by the HC and non-AP QSTA(s) in support of QoS traffic transfer using the procedures defined in 11.4. The element information format comprises the items as defined in this subclause, and the structure is defined in Figure 46ti.

[7]The default values for TXOP limit are expressed in milliseconds and are multiples of 32 μs.

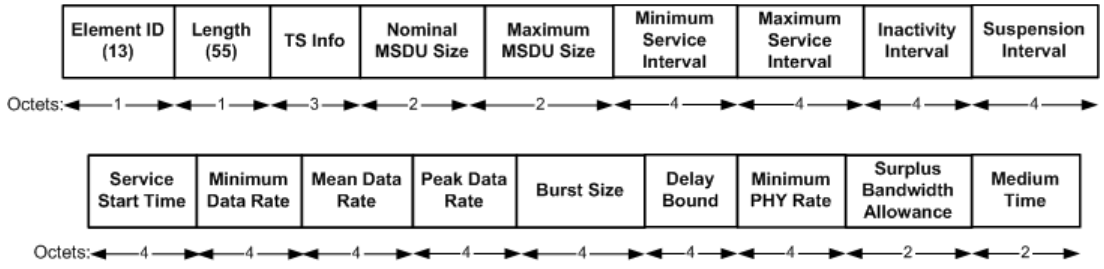


Figure 46ti—TSPEC element format

The TSPEC allows a set of parameters more extensive than may be needed, or may be available, for any particular instance of parameterized QoS traffic. Unless indicated otherwise, fields that follow the TS Info field are set to 0 for any unspecified parameter values. Non-AP QSTAs set the value of any parameters to unspecified if they have no information for setting that parameter. The HC may change the value of parameters that have been set unspecified by the QSTA to any value that it deems appropriate, including leaving them unspecified.

The structure of the TS Info field is defined in Figure 46tj.

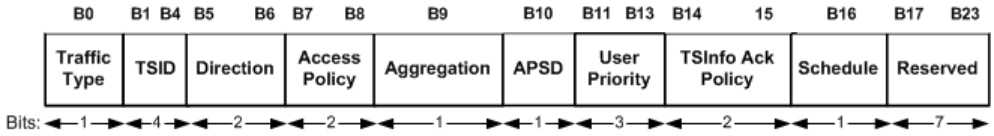


Figure 46tj—TS Info field

— The Traffic Type subfield is a single bit and is set to 1 for a periodic traffic pattern (e.g., isochronous TS of MSDUs, with constant or variable sizes, that are originated at fixed rate) or set to 0 for an aperiodic, or unspecified, traffic pattern (e.g., asynchronous TS of low-duty cycles).

— The TSID subfield is 4 bits in length and contains the TSID values in the format defined in 7.1.3.5.1. Note that the MSB (bit 4 in TS Info field) of the TSID subfield is always set to 1. The combination of the TSID and Direction subfields identify the TS, in the context of the non-AP QSTA, to which the TSPEC applies. The same TSID may be used for multiple TSs at different non-AP QSTAs. A non-AP QSTA may use the TSID subfield value for a downlink TSPEC and either an uplink or a direct-link TSPEC at the same time. A non-AP QSTA shall not use the same TSID for both uplink and direct-link TS. A bidirectional link request is equivalent to a downlink TS and an uplink TS, each with the same TSID and parameters.

— The Direction subfield specifies the direction of data carried by the TS as defined in Table 20dg.

Table 20dg—Direction subfield encoding

Bit 5	Bit 6	Usage
0	0	Uplink (MSDUs are sent from the non-AP QSTA to HC)
1	0	Downlink (MSDUs are sent from the HC to the non-AP QSTA)
0	1	Direct link (MSDUs are sent from the non-AP QSTA to another non-AP QSTA)
1	1	Bidirectional link (equivalent to a downlink request plus an uplink request, each direction having the same parameters). The fields in the TSPEC element specify resources for a single direction. Double the specified resources are required to support both streams.

— The Access Policy subfield is 2 bits in length, specifies the access that would be used for the TS, and is defined in Table 20dh.

Table 20dh—Access Policy subfield

Bit 7	Bit 8	Usage
0	0	Reserved
1	0	Contention-based channel access (EDCA)
0	1	Controlled channel access (HCCA)
1	1	HCCA, EDCA mixed mode (HEMM)

— The Aggregation subfield is 1 bit in length. The Aggregation subfield is valid only when access method is HCCA or when the access method is EDCA and the Schedule subfield is set to 1. It is set to 1 by a non-AP QSTA to indicate that an aggregate schedule is required. It is set to 1 by the QAP if an aggregate schedule is being provided to the non-AP QSTA. It is set to 0 otherwise. In all other cases, the Aggregation subfield is reserved.

— The APSD subfield is a single bit and is set to 1 to indicate that automatic PS delivery is to be used for the traffic associated with the TSPEC and set to 0 otherwise.

— The UP subfield is 3 bits and indicates the actual value of the UP to be used for the transport of MSDUs belonging to this TS in cases where relative prioritization is required. When the TCLAS element is present in the request, the UP subfield in TS Info field of the TSPEC element is reserved.

— The TS Info Ack Policy subfield is 2 bits in length and indicates whether MAC acknowledgments are required for MPDUs belonging to this TID and the desired form of those acknowledgments. The encoding of the TS Info Ack Policy subfield is shown in Table 20di. If the TS Info Ack Policy subfield is set to Block Ack and the type of Block Ack policy is unknown to the HC, the HC shall assume, for TXOP scheduling, that the immediate Block Ack policy is being used (see 9.10).

Table 20di—TS Info Ack Policy subfield encoding

Bit 14	Bit 15	Usage
0	0	Normal IEEE 802.11 acknowledgment The addressed recipient returns an ACK or QoS +CF-Ack frame after a SIFS period, according to the procedures defined in 9.2.8, 9.3.3, and 9.9.2.3.
1	0	No Ack The recipient(s) do not acknowledge the transmission.
0	1	Reserved
1	1	Block Ack A separate Block Ack setup mechanism described in 9.10 is used.

— The Schedule subfield is 1 bit in length and specifies the requested type of schedule. The setting of the subfield when the access policy is EDCA is shown in Table 20dj. When the Access Policy subfield is set to any value other than EDCA, the Schedule subfield is reserved. When the Schedule and APSD subfields are set to 1, the QAP shall set the aggregation bit to 1, indicating that an aggregate schedule is being provided to the non-AP QSTA.

The configuration of APSD=0, Schedule=1 is reserved.

Table 20dj—Setting of Schedule subfield

APSD	Schedule	Usage
0	0	No Schedule
1	0	Unscheduled APSD
0	1	Reserved
1	1	Scheduled APSD

The Nominal MSDU Size field is 2 octets long, contains an unsigned integer that specifies the nominal size, in octets, of MSDUs belonging to the TS under this TSPEC, and is defined in Figure 46tk. If the Fixed subfield is set to 1, then the size of the MSDU is fixed and is indicated by the Size subfield. If the Fixed subfield is set to 0, then the size of the MSDU might not be fixed and the Size subfield indicates the nominal MSDU size. If both the Fixed and Size subfields are set to 0, then the nominal MSDU size is unspecified.

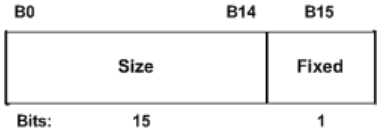


Figure 46tk—Nominal MSDU Size field

The Maximum MSDU Size field is 2 octets long and contains an unsigned integer that specifies the maximum size, in octets, of MSDUs belonging to the TS under this TSPEC.

The Minimum Service Interval field is 4 octets long and contains an unsigned integer that specifies the minimum interval, in microseconds, between the start of two successive SPs.

The Maximum Service Interval field is 4 octets long and contains an unsigned integer that specifies the maximum interval, in microseconds, between the start of two successive SPs.

The Inactivity Interval field is 4 octets long and contains an unsigned integer that specifies the minimum amount of time, in microseconds, that may elapse without arrival or transfer of an MPDU belonging to the TS before this TS is deleted by the MAC entity at the HC.

The Suspension Interval field is 4 octets long and contains an unsigned integer that specifies the minimum amount of time, in microseconds, that may elapse without arrival or transfer of an MSDU belonging to the TS before the generation of successive QoS(+)CF-Poll is stopped for this TS. A value of 4 294 967 295 ($= 2^{32} - 1$) disables the suspension interval, indicating that polling for the TS is not to be interrupted based on inactivity. The value of the suspension interval is always less than or equal to the inactivity interval.

The Service Start Time field is 4 octets and contains an unsigned integer that specifies the time, expressed in microseconds, when the SP starts. The service start time indicates to QAP the time when a non-AP QSTA first expects to be ready to send frames and a power-saving non-AP QSTA will be awake to receive frames. This may help the QAP to schedule service so that the MSDUs encounter small delays in the MAC and help the power-saving non-AP QSTAs to reduce power consumption. The field represents the four lower order bytes of the TSF timer at the start of the SP. If APSD subfield is set to 0, this field is also set to 0 (unspecified).

The Minimum Data Rate field is 4 octets long and contains an unsigned integer that specifies the lowest data rate specified at the MAC_SAP, in bits per second, for transport of MSDUs belonging to this TS within the bounds of this TSPEC. The minimum data rate does not include the MAC and PHY overheads incurred in transferring the MSDUs.

The Mean Data Rate[8] field is 4 octets long and contains an unsigned integer that specifies the average data rate specified at the MAC_SAP, in bits per second, for transport of MSDUs belonging to this TS within the bounds of this TSPEC. The mean data rate does not include the MAC and PHY overheads incurred in transferring the MSDUs.

The Peak Data Rate field is 4 octets long and contains an unsigned integer that specifies the maximum allowable data rate, in bits per second, for transfer of MSDUs belonging to this TS within the bounds of this TSPEC. If p is the peak rate in bits per second, then the maximum amount of data, belonging to this TS, arriving in any time interval $[t1,t2]$, where $t1 < t2$ and $t2 - t1 > 1$ TU, does not exceed $p * (t2 - t1)$ bits.

The Burst Size field is 4 octets long and contains an unsigned integer that specifies the maximum burst, in octets, of the MSDUs belonging to this TS that arrive at the MAC_SAP at the peak data rate. A value of 0 indicates that there are no bursts.

The Delay Bound field is 4 octets long and contains an unsigned integer that specifies the maximum amount of time, in microseconds, allowed to transport an MSDU belonging to the TS in this TSPEC, measured between the time marking the arrival of the MSDU at the local MAC sublayer from the local MAC_SAP and the time of completion of the successful transmission or retransmission of the MSDU to the destination. The completion of the MSDU transmission includes the relevant acknowledgment frame transmission time, if present.

The Minimum PHY Rate field is 4 octets long and contains an unsigned integer that specifies the desired minimum PHY rate to use for this TS, in bits per second, that is required for transport of the MSDUs belonging to the TS in this TSPEC[9].

The Surplus Bandwidth Allowance field is 2 octets long and specifies the excess allocation of time (and bandwidth) over and above the stated application rates required to transport an MSDU belonging to the TS in this TSPEC. This field is represented as an unsigned binary number and, when specified, is greater than 0. The 13 least significant bits (LSBs) indicate the decimal part while the three MSBs indicate the integer part of the number. This field takes into account the retransmissions, as the rate information does not include retransmissions. It represents the ratio of over-the-air bandwidth (i.e., time that the scheduler allocates for the transmission of MSDUs at the required rates) to bandwidth of the transported MSDUs required for successful transmission (i.e., time that would be necessary at the minimum PHY rate if there were no errors on the channel) to meet throughput and delay bounds under this TSPEC, when specified. As such, it should be greater than unity. A value of 1 indicates that no additional allocation of time is requested.

The Medium Time field is a 16-bit unsigned integer and contains the amount of time admitted to access the medium, in units of 32 μs/s. This field is reserved in the ADDTS Request frame and is set by the HC in the ADDTS Response frame. The derivation of this field is described in K.2.2. This field is not used for controlled channel access.

[8]The mean data rate, the peak data rate, and the burst size are the parameters of the token bucket model, which provides standard terminology for describing the behavior of a traffic source. The token bucket model is described in IETF RFC 2212 [B20], IETF RFC 2215 [B21], and IETF RFC 3290 [B23].

[9]This rate information is intended to ensure that the TSPEC parameter values resulting from an admission control negotiation are sufficient to provide the required throughput for the TS. In a typical implementation, a TS is admitted only if the defined traffic volume can be accommodated at the specified rate within an amount of WM occupancy time that the admissions control entity is willing to allocate to this TS.

The UP, Minimum Data Rate, Mean Data Rate, Peak Data Rate, Burst Size, Minimum PHY Rate, and Delay Bound fields in a TSPEC element express the QoS expectations requested by a non-AP QSTA, if this TSPEC was issued by that non-AP QSTA, or provided by the HC, if this TSPEC was issued by the HC, when these fields are specified with nonzero values. Unspecified parameters in these fields as indicated by a zero value indicate that the non-AP QSTA does not have specific requirements for these parameters if the TSPEC was issued by that non-AP QSTA or that the HC does not provide any specific values for these parameters if the TSPEC was issued by the HC.

7.3.2.29 TCLAS element

The TCLAS element specifies an information element that contains a set of parameters necessary to identify incoming MSDUs (from a higher layer in all QSTAs or from the DS in an AP) with a particular TS to which they belong. If required, the TCLAS element is provided in ADDTS Request and ADDTS Response frames only for the downlink or bidirectional links. TCLAS element need not be provided for the uplink or direct-link transmissions. The structure of this element is shown in Figure 46tl.

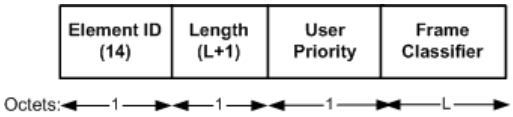


Figure 46tl—TCLAS element format

The UP field contains the value of the UP of the associated MSDUs.

The Frame Classifier field is 3–255 octets in length and is defined in Figure 46tm.

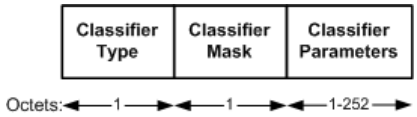


Figure 46tm—Frame Classifier field

The Frame Classifier field comprises the following subfields: Classifier Type, Classifier Mask, and Classifier Parameters. The Classifier Type subfield is 1 octet in length and specifies the type of classifier parameters in this TCLAS as defined in Table 20dk. Three classifier types are defined later in this subclause.

Table 20dk—Frame classifier type

Classifier type	Classifier parameters
0	Ethernet parameters
1	TCP/UDP IP parameters
2	IEEE 802.1D/Q parameters
3–255	Reserved

The Classifier Mask subfield specifies a bitmap where bits that are set to 1 identify a subset of the classifier parameters whose values must match those of the corresponding parameters in a given MSDU for that MSDU to be classified to the TS of the affiliated TSPEC. The bitmap is ordered from the LSB to the MSB,

with each bit pointing to one of the classifier parameters of the same relative position as shown in this subclause based on classifier type. An incoming MSDU that failed to be classified to a particular TS may be classified to another active TS based on the frame classifier for that TS. If, however, all the frame classifiers for the active TS have been exhausted, the MSDU does not belong to any active TS and is classified to be a best-effort MSDU. In cases where there are more bits in the bitmap than classifier parameters that follow, the MSBs that do not point to any classifier parameters are reserved.

For Classifier Type 0, the classifier parameters are the following parameters contained in an Ethernet packet header: Source Address, Destination Address, and Type ("Ethernet" [B17]). The Frame Classifier field for Classifier Type 0 is defined in Figure 46tn. It has a length of 16 octets.

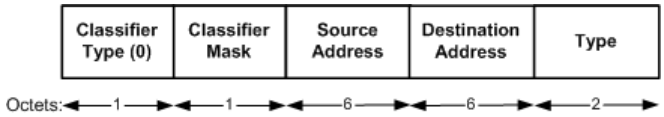


Figure 46tn—Frame Classifier field of Classifier Type 0

For Classifier Type 1, frame classifier is defined for both IPv4 and IPv6, shown in Figure 46to and Figure 46tp, and distinguished by the Version field. The subfields in the classifier parameters are represented and transmitted in the big-endian format. The classifier parameters are the following parameters:

— In a TCP or UDP header: Source Address, Destination Address, Source Port, Destination Port, and Version, plus

— One of the following:

  — In an IPv4 header: Differentiated Services Code Point (DSCP) (IETF RFC 2472 [B22]) and Protocol, or

  — In an IPv6 header: Flow Label.

The DSCP field will contain the value in the 6 LSBs, and the 2 MSBs are set to 0. The 2 MSBs of the DSCP field are ignored for frame classification.

Figure 46to—Frame Classifier field of Classifier Type 1 for traffic over IPv4

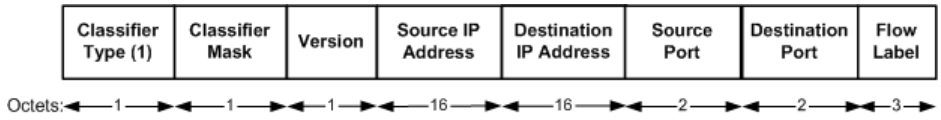


Figure 46tp—Frame Classifier field of Classifier Type 1 for traffic over IPv6

For Classifier Type 2, the Classifier Parameters are the following parameters in an IEEE 802.1Q tag header: IEEE 802.1D user priority and IEEE 802.1Q VLAN ID. The Frame Classifier field for Classifier Type 2 is defined in Figure 46tq.

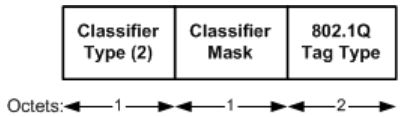


Figure 46tq—Frame Classifier field of Classifier Type 2

7.3.2.30 TS Delay element

The TS Delay element is used in the ADDTS Response frame transmitted by the HC to a non-AP QSTA and indicates the time after which the ADDTS may be retried. The TS Delay element is defined in Figure 46tr.

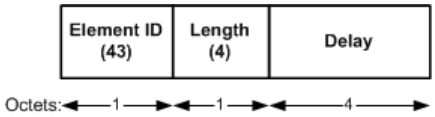


Figure 46tr—TS Delay element

The Delay field is 4 octets long and specifies the amount of time, in TUs, a non-AP QSTA should wait before it reinitiates setup of a TS.

The TS Delay element is set to 0 when a QAP does not want to serve any TSPECs for an indeterminate time and it does not know this time a priori.

7.3.2.31 TCLAS Processing element

The TCLAS Processing element is present in the ADDTS Request and Response frames if there are multiple TCLASs associated with the request. It indicates how an MSDU received from higher layers should be processed by the classifier. The TCLAS Processing element is defined in Figure 46ts.

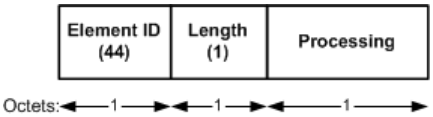


Figure 46ts—TCLAS Processing element

The Processing subfield is 1 octet long. The encoding of the Processing subfield is shown in Table 20dl.

Table 20dl—Encoding of Processing subfield

Processing subfield value	Meaning
0	Incoming MSDU's higher layer parameters have to match to the parameters in all the associated TCLAS elements.
1	Incoming MSDU's higher layer parameters have to match to at least one of the associated TCLAS elements.
2	Incoming MSDUs that do not belong to any other TS are classified to the TS for which this TCLAS Processing element is used. In this case, there will not be any associated TCLAS elements.
3–255	Reserved

7.3.2.32 Schedule element

The Schedule element is transmitted by the HC to a non-AP QSTA to announce the schedule that the HC/QAP follows for admitted streams originating from or destined to that non-AP QSTA in the future. The information in this element may be used by the non-AP QSTA for power management, internal scheduling, or any other purpose. The element information format is shown in Figure 46tt.

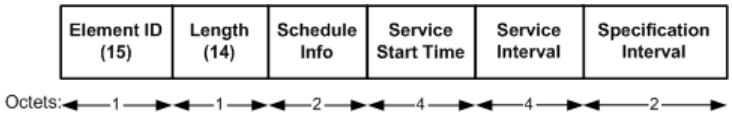


Figure 46tt—Schedule element

The Schedule Info field is shown in Figure 46tu.

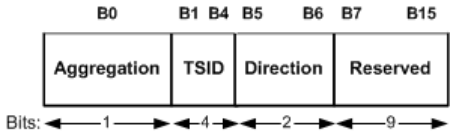


Figure 46tu—Schedule Info field

The Aggregation subfield is set to 1 if the schedule is an aggregate schedule for all TSIDs associated with the non-AP QSTA to which the frame is directed. It is set to 0 otherwise. The TSID subfield is as defined in 7.1.3.5.1 and indicates the TSID for which this schedule applies. The Direction subfield is as defined in 7.3.2.28 and defines the direction of the TSPEC associated with the schedule. The TSID and Direction subfields are valid only when the Aggregation subfield is set to 0. If the Aggregation subfield is set to 1, the TSID and Direction subfields are reserved.

The Service Start Time field is 4 octets and indicates the anticipated time, expressed in microseconds, when service starts and represents the lower order 4 bytes of the TSF timer value at the start of the first SP. The QAP uses this field to confirm or modify the service start time indicated in the TSPEC request.

The Service Interval field is 4 octets and indicates the time, expressed in microseconds, between two successive SPs and represents the measured time from the start of one SP to the start of the next SP.

The Specification Interval field is 2 octets long and contains an unsigned integer that specifies the time interval, in TUs, to verify schedule conformance.

The HC may set the Service Start Time field and the Service Interval field to 0 (unspecified) for nonpower-saving STAs.

7.3.2.33 QoS Capability element

The QoS Capability element contains a number of subfields that are used to advertise optional QoS capabilities at a QSTA. The QoS Capability element is present in Beacon frames that do not contain the EDCA Parameter Set element and in (Re)Association Request frames. The QoS Capability element is defined in Figure 46tv.

The QoS Info field is 1 octet in length and is defined in 7.3.1.17.

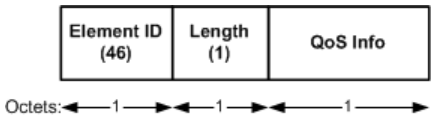


Figure 46tv—QoS Capability element format

7.4 Action frame format details

Insert after 7.4.1.5 the following new subclauses (7.4.2 through 7.5), including the tables in those subclauses:

7.4.2 QoS Action frame details

Several Action frame formats are defined for QoS purposes. The Action field values associated with each frame format within the QoS category are defined in Table 20ea.

Table 20ea—QoS Action field values

Action field value	Meaning
0	ADDTS Request
1	ADDTS Response
2	DELTS
3	Schedule
4–255	Reserved

7.4.2.1 ADDTS Request frame format

The ADDTS frames are used to carry TSPEC and optionally TCLAS elements to set up and maintain TSs using the procedures defined in 11.4.

The frame body of the ADDTS Request frame contains the information shown in Table 20eb.

Table 20eb—ADDTS Request frame body

Order	Information
1	Category
2	Action
3	Dialog Token
4	TSPEC
5–n	TCLAS (optional)
n + 1	TCLAS Processing (optional)

The Category field is set to 1 (representing QoS).

The Action field is set to 0 (representing ADDTS request).

The Dialog Token, TCLAS, and TCLAS Processing fields of this frame are contained in an MLME-ADDTS.request primitive that causes the frame to be sent. Some of the TSPEC parameters are contained in the MLME-ADDTS.request primitive while the other parameters (i.e., Surplus Bandwidth Allowance, Minimum Service Interval, Maximum Service Interval, and Minimum PHY Rate) are generated within the MAC.

The TSPEC element, defined in 7.3.2.28, and the optional TCLAS element, defined in 7.3.2.29, contain the QoS parameters that define the TS. The TS is identified by the TSID and Direction fields within the TSPEC element. The TCLAS element is optional at the discretion of the non-AP QSTA that sends the ADDTS Request frame, regardless of the setting of the access policy (EDCA or HCCA). There may be one or more TCLAS elements in the ADDTS frame. The TCLAS Processing element is present when there are more than one TCLAS element and is defined in 7.3.2.31.

7.4.2.2 ADDTS Response frame format

The ADDTS Response frame is transmitted in response to an ADDTS request frame. The frame body of the ADDTS Response frame contains the information shown in Table 20ec.

Table 20ec—ADDTS Response frame body

Order	Information
1	Category
2	Action
3	Dialog Token
4	Status Code
5	TS Delay
6	TSPEC
7–n	TCLAS (optional)
n + 1	TCLAS Processing (optional)
n + 2	Schedule

The Category field is set to 1(representing QoS).

The Action field is set to 1 (representing ADDTS response).

The Status Code field is defined in 7.3.1.9.

The Dialog Token, TS Delay, TSPEC, TCLAS, and TCLAS Processing fields in this frame are contained in an MLME-ADDTS.response primitive that causes the frame to be sent. The TS Delay information element is present in an ADDTS Response frame only if the status code is set to 47.

The Schedule element, defined in 7.3.2.32, is present in an ADDTS Response frame only if the status code is set to 0 (i.e., when the TS is admitted).

7.4.2.3 DELTS frame format

The DELTS frame is used to delete a TS using the procedures defined in 11.4.7.

The frame body of a DELTS frame contains the information shown in Table 20ed.

Table 20ed—DELTS frame body

Order	Information
1	Category
2	Action
3	TS Info
4	Reason Code

The Category field is set to 1 (representing QoS).

The Action field is set to 2 (representing DELTS).

The TS Info field is defined in 7.3.2.28.

The Reason Code field is defined in 7.3.1.7.

A DELTS frame is used to delete a TS characterized by the TS Info field in the frame. A DELTS frame may be sent from the HC to the source STA of that TS, or vice versa, to indicate an imperative request, to which no response is required from the recipient STA.

7.4.2.4 Schedule frame format

The Schedule frame is transmitted by the HC to a non-AP QSTA to announce the schedule of delivery of data and polls. The frame body of the Schedule frame contains the information shown in Table 20ee.

Table 20ee—Schedule frame body

Order	Information
1	Category
2	Action
3	Schedule

The Category field is set to 1 (representing QoS).

The Action field is set to 3 (representing Schedule).

The Schedule element is defined in 7.3.2.32.

7.4.3 DLS Action frame details

Several Action frame formats are defined for DLS management purposes. An Action field, in the octet field immediately after the Category field, differentiates the formats. The Action field values associated with each frame format are defined in Table 20ef.

Table 20ef—DLS Action field values

Action field value	Meaning
0	DLS Request
1	DLS Response
2	DLS Teardown
3–255	Reserved

7.4.3.1 DLS Request frame format

The DLS Request frame is used to set up a direct link with a peer MAC. The frame body of the DLS Request frame contains the information shown in Table 20eg.

Table 20eg—DLS Request frame body

Order	Information
1	Category
2	Action
3	Destination MACAddress
4	Source MACAddress
5	Capability Information
6	DLS Timeout Value
7	Supported rates
8	Extended Supported Rates

The Category field is set to 2 (representing DLS).

The Action field is set to 0 (representing DLS request).

The Destination MAC Address field value is the MAC address of the target destination.

The Source MAC Address field value is the MAC address of the initiating STA.

The Capability Information field value is the capability information of the originator of the request.

The DLS Timeout Value field is defined in 7.3.1.13.

The Supported Rates and Extended Supported Rates fields contain the supported rates information of the originator.

7.4.3.2 DLS Response frame format

The DLS Response frame is sent in response to a DLS Request frame. The frame body of a DLS Response frame contains the information shown in Table 20eh.

Table 20eh—DLS Response frame body

Order	Information
1	Category
2	Action
3	Status Code
4	Destination MACAddress
5	Source MACAddress
6	Capability Information
7	Supported rates
8	Extended Supported rates

The Category field is set to 2 (representing DLS).

The Action field is set to 1 (representing DLS response).

The Status Code field is defined in 7.3.1.9.

The Destination MAC Address field value and the Source MAC Address field value are copied from the corresponding fields in the DLS Request frame.

The Capability Information field is the capability information of the target destination. This information is included only if the DLS result code corresponds to SUCCESS (DLS status code 0).

The Supported Rates and Extended Supported Rates fields contain the supported rates information of the target destination. This information is included only if the DLS result code corresponds to SUCCESS (DLS status code 0).

7.4.3.3 DLS Teardown frame format

The DLS Teardown frame is sent to terminate a direct link with a peer MAC. The frame body of the DLS Teardown frame contains the information shown in Table 20ei.

Table 20ei—DLS Teardown frame body

Order	Information
1	Category
2	Action
3	Destination MACAddress
4	Source MACAddress
5	Reason Code

The Category field is set to 2 (representing DLS).

The Action field is set to 2 (representing DLS teardown).

The Destination MAC Address field value is the MAC address of the target destination.

The Source MAC Address field value is the MAC address of the initiating STA.

The Reason Code field is defined in 7.3.1.7.

7.4.4 Block Ack Action frame details

The ADDBA frames are used to set up Block Ack for a specific TC or TS. The Action field values associated with each frame format within the Block Ack category are defined in Table 20ej.

Table 20ej—Block Ack Action field values

Action field values	Meaning
0	ADDBA Request
1	ADDBA Response
2	DELBA
3–255	Reserved

7.4.4.1 ADDBA Request frame format

An ADDBA Request frame is sent by an originator of Block Ack to another QSTA. The frame body of an ADDBA Request frame contains the information shown in Table 20ek.

Table 20ek—ADDBA Request frame body

Order	Information
1	Category
2	Action
3	Dialog Token
4	Block Ack Parameter Set
5	Block Ack Timeout Value
6	Block Ack Starting Sequence Control

The Category field is set to 3 (representing Block Ack).

The Action field is set to 0 (representing ADDBA request).

The Dialog Token field is set to a nonzero value chosen by the STA.

The Block Ack Parameter Set field is defined in 7.3.1.14.

The Block Ack Timeout Value field is defined in 7.3.1.15.

The Block Ack Starting Sequence Control field is defined in 7.2.1.7.

7.4.4.2 ADDBA Response frame format

The ADDBA Response frame is sent in response to an ADDBA Request frame. The frame body of an ADDBA Response frame contains the information shown in Table 20el.

Table 20el—ADDBA Response frame body

Order	Information
1	Category
2	Action
3	Dialog Token
4	Status Code
5	Block Ack Parameter Set
6	Block Ack Timeout Value

The Category field is set to 3 (representing Block Ack).

The Action field is set to 1 (representing ADDBA response).

The Dialog Token field value is copied from the corresponding received ADDBA Request frame.

The Status Code field is defined in 7.3.1.9.

The Block Ack Parameter Set field is defined in 7.3.1.14.

The Block Ack Timeout Value field is defined in 7.3.1.15.

7.4.4.3 DELBA frame format

The DELBA frame is sent by either the originator of the traffic or the recipient to terminate the Block Ack participation. The frame body of a DELBA frame format contains the information shown in Table 20em.

Table 20em—DELBA frame body

Order	Information
1	Category
2	Action
3	DELBA Parameter Set
4	Reason Code

The Category field is set to 3 (representing DELBA).

The Action field is set to 2 (representing DELBA).

The DELBA Parameters field is defined in 7.3.1.16.

The Reason Code field is defined in 7.3.1.7.

7.5 Frame usage

Table 20en shows which frame subtypes are transmitted and received by different kinds of MAC entities operating in the different types of BSS and under the available coordination functions.

Table 20en—Frame subtype usage by BSS type, MAC entity type, and coordination function

Frame subtype	IBSS		Infrastructure BSS							
	non-QoS	QoS	non-QoS				QoS			
	CP	CP	CP		CFP		CP		CFP	
	STA	QSTA	STA	AP	STA	AP	QSTA	QAP	QSTA	QAP
(Re)Association Request	- - -	- - -	T	R	- - -	- - -	T	R	- - -	- - -
(Re)Association Response	- - -	- - -	R	T	R	T	R	T	R	T

Table 20en—Frame subtype usage by BSS type, MAC entity type, and coordination function *(continued)*

Frame subtype	IBSS		Infrastructure BSS							
	non-QoS	QoS	non-QoS				QoS			
	CP	CP	CP		CFP		CP		CFP	
	STA	QSTA	STA	AP	STA	AP	QSTA	QAP	QSTA	QAP
Probe Request	T, R	T, R	T	R	- - -	- - -	T	R	- - -	- - -
Probe Response	Tbe, R	Tbe, R	R	T	R	T	R	T	R	T
Beacon	Tb, R	Tb, R	R	T	R	T	R	T, R	R	T, R
ATIM	T, R	T, R	- - -	- - -	- - -	- - -	- - -	- - -	- - -	- - -
Disassociation	T, R	T, R	T, R	T, R	T, R	T, R	T, R	T, R	T, R	T, R
Authentication	T, R	T, R	T, R	T, R	T, R	T, R	T, R	T, R	T, R	T, R
Deauthentication	T, R	T, R	T, R	T, R	T, R	T, R	T, R	T, R	T, R	T, R
ADDTS Request	- - -	- - -	- - -	- - -	- - -	- - -	T	R	T	R
ADDTS Response	- - -	- - -	- - -	- - -	- - -	- - -	R	T	R	T
DELTS	- - -	- - -	- - -	- - -	- - -	- - -	T, R	T, R	T, R	T, R
Schedule	- - -	- - -	- - -	- - -	- - -	- - -	R	T	R	T
DLS Action	- - -	- - -	- - -	- - -	- - -	- - -	T, R	T, R	T, R	T, R
Block Ack Action	- - -	T, R	- - -	- - -	- - -	- - -	T, R	T, R	T, R	T, R
BlockAckReq/ BlockAck	- - -	T, R	- - -	- - -	- - -	- - -	T, R	T, R	T, R	T, R
PS-Poll	- - -	- - -	T	R	- - -	- - -	T	R	T	R
RTS	T, R	T, R	T, R	T, R	- - -	- - -	T, R	T, R	T, R	T, R
CTS	T, R	T, R	T, R	T, R	- - -	- - -	T, R	T, R	T, R	T, R
ACK	T, R	T, R	T, R	T, R	T, R	T, R	T, R	T, R	T, R	T, R
CF-End	(R)	(R)	(R)	(R)	R	T	(R)	(R)	R	T
CF-End+CF-Ack	(R)	(R)	(R)	(R)	R	T	(R)	(R)	R	T
Null	T, R	T, R	T, R	T, R	T, R	T, R	T, R	T, R	T, R	T, R
Data	T, R	T, R	T, R	T, R	T, R	T, R	T, R	T, R	T, R	T, R
(Data+)CF-Poll(+CF-Ack)	- - -	- - -	- - -	- - -	R	T	- - -	- - -	- - -	- - -
(Data+)CF-Ack	- - -	- - -	- - -	- - -	T, R	T, R	- - -	- - -	T, R, Rda, Rq	T, R, Rda, Rq
QoS Null	- - -	T, R	- - -	- - -	- - -	- - -	T, R	T, R	T, R	T, R
QoS Data	- - -	T, R	- - -	- - -	- - -	- - -	T, R	T, R	T, R	T, R
QoS (Data+)CF-Poll	- - -	- - -	- - -	- - -	- - -	- - -	R	T	R	T

Table 20en—Frame subtype usage by BSS type, MAC entity type, and coordination function *(continued)*

Frame subtype	IBSS		Infrastructure BSS							
	non-QoS	QoS	non-QoS				QoS			
	CP	CP	CP		CFP		CP		CFP	
	STA	QSTA	STA	AP	STA	AP	QSTA	QAP	QSTA	QAP
QoS (Data+)CF-Poll+CF-Ack	- - -	- - -	- - -	- - -	- - -	- - -	Rq, Rda	Tda, Tq	Rq, Rda	Tda, Tq
QoS Data+CF-Ack	- - -	- - -	- - -	- - -	- - -	- - -	T, Rq, Rda	Tda, Tq, R	T, Rq, Rda	Tda, Tq, R

Symbols:

T frame subtype is transmitted by MAC entity for column.

R frame subtype is received by MAC entity for column.

(R) frame subtype is received, but only from other BSSs, by MAC entity for column.

Tb frame subtype is transmitted by STA that most recently won beacon arbitration.

Tbe frame subtype is transmitted by a QSTA in an IBSS pursuant to receiving directed request.

Tda frame subtype is transmitted only if recipient of +CF-Ack function is addressee.

Rda frame subtype is received if QSTA is addressee.

Tq frame subtype is transmitted only if recipient of +CF-Ack function has set the Q-Ack subfield in QoS Capability element to 1.

Rq frame subtype is received if QSTA is not the addressee, but has set the Q-Ack subfield in QoS Capability element to 1.

- - - frame subtype is neither received nor transmitted by MAC entity for column.

End of changes to Clause 7.

8. Security

8.3 RSNA data confidentiality protocols

8.3.2 Temporal Key Integrity Protocol (TKIP)

8.3.2.3 TKIP MIC

8.3.2.3.1 Motivation for the TKIP MIC

Change the third item of the dashed list below Figure 43f in 8.3.2.3.1 as shown:

— the MSDU Priority ~~(Reserved for future use)~~; and

Change the paragraph above Figure 43g in 8.3.2.3.1 as shown:

The DA field, SA field, three reserved octets, and a 1-octet Priority field are used only for calculating the MIC. ~~The Priority field shall be 0 and reserved for future use.~~ The fields in Figure 43g are treated as an octet stream using the conventions described in 7.1.1.

8.3.2.6 TKIP replay protection procedures

Delete the following sentence from item g) of the lettered list in 8.3.2.6 as shown:

g) .... ~~IEEE 802.11 does not define a method to signal frame priority.~~

Insert the following sentence at the end of the lettered list in 8.3.2.6:

i) Block Ack reordering shall be completed prior to replay detection.

8.3.3 CTR with CBC-MAC Protocol (CCMP)

8.3.3.3 CCMP encapsulation

Change item c) of the lettered list in 8.3.3.3 as follows:

c) Construct the CCM Nonce block from the PN, A2, and Priority of the MPDU where A2 is MPDU Address 2. ~~The Priority is a reserved value set to 0.~~

8.3.3.3.2 Construct additional authentication data (AAD)

Change Figure 43p as shown:

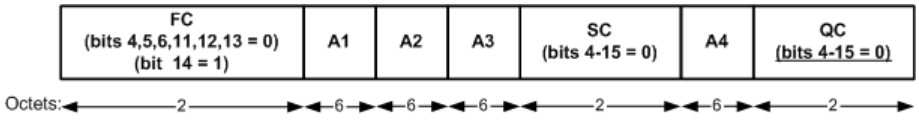


Figure 43p—AAD construction

Insert the following text and table after Figure 43p:

The length of the AAD varies depending on the presence or absence of the QC and A4 fields and is shown in Table 20eo:

Table 20eo—AAD length

QC field	A4 field	AAD length (octets)
Absent	Absent	22
Present	Absent	24
Absent	Present	28
Present	Present	30

Change the text in item g) of the lettered list in 8.3.3.3.2 as follows:

g) QC – QoS Control, if present, a 2-octet field that includes the MSDU priority~~; this field is reserved for future use~~. The QC TID is used in the construction of the AAD, and the remaining QC fields are set to 0 for the AAD calculation (bits 4 to 15 are set to 0).

Delete the last paragraph in 8.3.3.3.2 as follows:

~~The length of the AAD is 22 octets when no A4 field and no QC field exist and 28 octets long when the MPDU includes the A4 field.~~

8.3.3.3.3 Construct CCM Nonce

Change the first item in the dashed list in 8.3.3.3.3 as shown:

— ~~This Priority Octet field shall be 0 and reserved for future use with IEEE 802.11 frame prioritization.~~ The Priority Octet field shall be set to the fixed value 0 (0x00) when there is no QC field present in the MPDU header. When the QC field is present, bits 0 to 3 of the Priority Octet field shall be set to the value of the QC TID (bits 0 to 3 of the QC field). Bits 4 to 7 of the Proirity Octet field are reserved and shall be set to 0.

8.3.3.4 CCMP decapsulation

Change item c) in the lettered list in 8.3.3.4 as follows:

c) The Nonce value is constructed from A2, PN, and Priority Octet fields ~~(reserved and set to 0)~~.

8.3.3.4.3 PN and replay detection

Insert the following sentence at the end of the lettered list in 8.3.3.4.3:

g) Block Ack reordering shall be completed prior to replay detection.

8.4 RSNA security association management

8.4.1 Security associations

8.4.1.1 Security association definitions

8.4.1.1.4 STAKeySA

Change the second item in the dashed list in 8.4.1.1.4 as shown:

— ~~Pairwise~~STAKey cipher suite selector

8.5 Keys and key distribution

8.5.2 EAPOL-Key frames

Insert in 8.5.2 the following paragraph after the first paragraph (which includes a dashed list):

When priority processing of data frames is supported, a STA's SME should send EAPOL-Key frames at the highest priority.

8.5.5 STAKey Handshake

8.5.5.1 STAKey Request message

Change the final paragraph in 8.5.5.1 as shown:

A STA sends a protected STAKey Request message to the AP with the MAC address of the peer STA. On reception of a STAKey Request message, the AP verifies that the received Key Replay Counter field value is equal to or larger than its local copy of the counter. It then verifies that the MIC is valid and sets the local Key Replay Counter field value for the initiating STA to the received value. The AP verifies that the peer STA supports the indicated STAKey cipher suite. If the indicated STAKey cipher suite is not supported by the peer STA, the SME in the AP initiates an MLME-DLSTeardown.request primitive with a ReasonCode value of STAKEY_MISMATCH. Upon receipt of MLME-DLSTeardown.request primitive from the SME, the QAP shall inform the teardown of the direct link by sending the DLS Teardown frame to the two QSTAs using the direct link (also see 11.7.3.2).

End of changes to Clause 8.

9. MAC sublayer functional description

Change the introductory text of Clause 9 as shown:

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), and their coexistence in an IEEE 802.11 LAN are introduced in 9.1. These functions are expanded on in 9.2 (DCF), ~~and~~ 9.3 (PCF), and 9.9 (HCF)~~, and a complete functional description of each is provided~~. Fragmentation and defragmentation are defined~~covered~~ in 9.4 and 9.5. Multirate support is addressed in 9.6. ~~The allowable frame exchange sequences are listed in 9.7. Finally, a~~A number of additional restrictions to limit the cases in which MSDUs are reordered or discarded are described in ~~9.8~~9.7. Operation across regulatory domains is defined in 9.8. The Block Ack mechanism is described in 9.10. The No Ack mechanism is described in 9.11. The allowable frame exchange sequences are defined in 9.12. The protection mechanism is described in 9.13.

9.1 MAC architecture

Change the text of 9.1, including Figure 47, as shown:

The MAC architecture can be described as shown in Figure 47 as providing the PCF and HCF through the services of the DCF. Note that in an nQSTA, HCF is not present. In a QSTA implementation, both DCF and HCF are present. PCF is optional in all STAs.

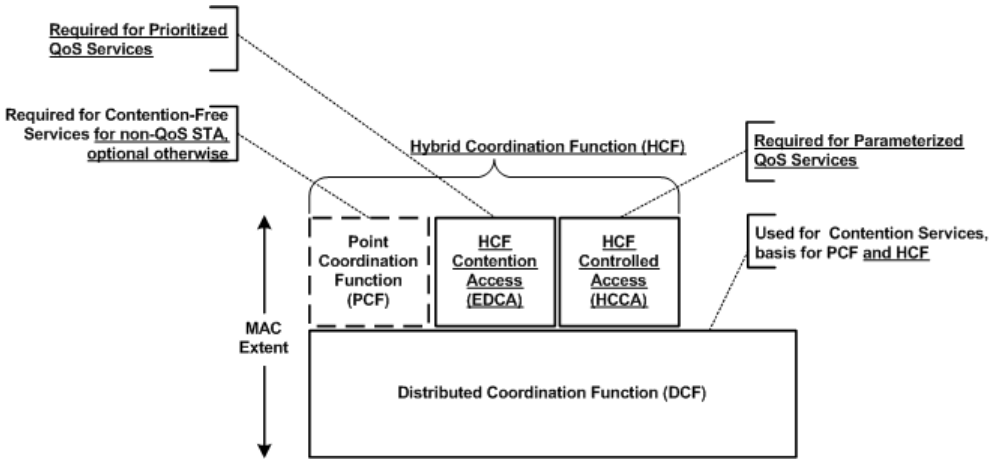


Figure 47—MAC architecture

Insert after 9.1.2 the following new subclauses as 9.1.3 through 9.1.3.2, including the table in those subclauses (instructions for renumbering the existing 9.1.3 are given after this new text):

9.1.3 Hybrid coordination function (HCF)

The QoS facility includes an additional coordination function called *HCF* that is only usable in QoS network (QBSS) configurations. The HCF shall be implemented in all QSTAs. 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.

QSTAs may obtain TXOPs using one or both of the channel access mechanisms specified in 9.9. If a TXOP is obtained using the contention-based channel access, it is defined as *EDCA TXOP*. If a TXOP is obtained using the controlled channel access, it is defined as *HCCA TXOP*. If an HCCA TXOP is obtained due to a QoS (+)CF-Poll frame from the HC, the TXOP is defined as a *polled TXOP*.

9.1.3.1 HCF contention-based channel access (EDCA)

The EDCA mechanism provides differentiated, distributed access to the WM for QSTAs using eight different UPs. The EDCA mechanism defines four access categories (ACs) that provide support for the delivery of traffic with UPs at the QSTAs. The AC is derived from the UPs as shown in Table 20i.

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 from the EDCA Parameter Set element or from the default values for the parameters when no EDCA Parameter Set element is received from the QAP of the QBSS with which the QSTA is associated, where

a) The parameters used by the EDCAF to control its operation are defined by MIB attribute table dot11QAPEDCATable at the QAP and by MIB attribute table dot11EDCATable at the non-AP QSTA.

b) The minimum specified idle duration time is not the constant value (DIFS) as defined for DCF, but is a distinct value (contained in the MIB attribute table dot11QAPEDCATableAIFSN for a QAP and in the MIB table dot11EDCATableAIFSN for a non-AP QSTA; see 9.9.1) assigned either by a management entity or by a QAP.

Table 20i—UP-to-AC mappings

Priority	UP (Same as 802.1D user priority)	802.1D designation	AC	Designation (informative)
Lowest ↓ Highest	1	BK	AC_BK	Background
	2	—	AC_BK	Background
	0	BE	AC_BE	Best Effort
	3	EE	AC_BE	Best Effort
	4	CL	AC_VI	Video
	5	VI	AC_VI	Video
	6	VO	AC_VO	Voice
	7	NC	AC_VO	Voice

c) The contention window limits aCWmin and aCWmax, from which the random backoff is computed, are not fixed per PHY, as with DCF, but are variable (contained in the MIB attribute tables dot11QAPEDCACWmin and dot11QAPEDCACWmax for a QAP and in the MIB attribute tables dot11EDCATableCWmin and dot11EDCATableCWmax for a non-AP QSTA) and assigned by a management entity or by a QAP.

d) Collisions between contending EDCAFs within a QSTA are resolved within the QSTA so that the data frames from the higher priority AC receives the TXOP and the data frames from the lower priority colliding AC(s) behave as if there were an external collision on the WM. Note, however, that this collision behavior does not include setting retry bits in the MAC headers of MPDUs at the head of the lower priority ACs, as would be done after a transmission attempt that was unsuccessful due to an actual external collision on the WM.

e) During an EDCA TXOP won by an EDCAF, a QSTA may initiate multiple frame exchange sequences to transmit MMPDUs and/or MSDUs within the same AC. The duration of this EDCA TXOP is bounded, for an AC, by the value in dot11QAPEDCATXOPLimit MIB variable for a QAP and in dot11EDCATableTXOPLimit MIB table for a non-AP QSTA. A value of 0 for this duration means that the EDCA TXOP is limited to a single MSDU or MMPDU at any rate in the operational set of the QBSS.

The QAP announces the EDCA parameters in selected Beacon frames and in all Probe Response and (Re)Association Response frames by the inclusion of the EDCA Parameter Set information element. If no such element is received, the QSTAs shall use the default values for the parameters. The fields following the QoS Info field in the EDCA Parameter Set information element shall be included in all Beacon frames occurring within two or more delivery traffic indication message (DTIM) periods following a change in AC parameters to assure that all QSTAs are able to receive the updated EDCA parameters. A QSTA shall update its MIB values of the EDCA parameters within an interval of time equal to one beacon interval after receiving an updated EDCA parameter set. QSTAs update the MIB attributes and store the EDCA Parameter Set update count value in the QoS Info field. A QAP may change the EDCA access parameters by changing the EDCA Parameter Set element in the Beacon and Probe Response frames. However, the QAP should change them only rarely. A QSTA shall use the EDCA Parameter Set Update Count Value subfield in the QoS Capability element of all Beacon frames to determine if the QSTA is using the current EDCA Parameter Values. If the EDCA Parameter Set update count value in the QoS Capability element is different from the value that has been stored, the QSTA shall query the updated EDCA parameter values by sending a Probe Request frame to the QAP.

The QAP may use a different set of EDCA parameters than it advertises to the QSTAs in its QBSS.

The management frames shall be sent using the access category AC_VO without being restricted by admission control procedures. A QSTA shall also send management frames using the access category AC_VO before associating with any BSS, 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. For the purpose of determining the proper AC for an RTS frame, the RTS frame shall inherit the UP of the data or management frame(s) that are included in the frame exchange sequence where the RTS is the first frame.

The operation rules of HCF contention-based channel access are defined in 9.9.1.

9.1.3.2 HCF controlled channel access (HCCA)

The HCCA mechanism uses a QoS-aware centralized coordinator, called a *hybrid coordinator* (HC), and operates under rules that are different from the PC of the PCF. The HC is collocated with the QAP of the QBSS and uses the HC's higher priority of access to the WM to initiate frame exchange sequences and to allocate TXOPs to itself and other QSTAs in order to provide limited-duration controlled access phase (CAP) for contention-free transfer of QoS data.

The HC traffic delivery and TXOP allocation may be scheduled during the CP and any locally generated CFP (generated optionally by the HC) to meet the QoS requirements of a particular TC or TS. TXOP allocations and contention-free transfers of QoS traffic can be based on the HC's QBSS-wide knowledge of the amounts of pending traffic belonging to different TS and/or TCs and are subject to QBSS-specific QoS policies.

A QAP may indicate availability of CF-Polls to nQSTAs, thereby providing non-QoS contention-free transfers during the CFP. This provisioning of contention-free transfers during the CFP to nQSTAs, however, is not recommended. Implementers are cautioned that QSTAs are not required to interpret data subtypes that include QoS +CF-Ack in frames not addressed to themselves unless they set the Q-Ack subfield in the QoS Capability information element to 1. QSTAs are also not required to interpret data subtypes that are non-QoS (+)CF-Poll frames (i.e., data frames with bits 7, 5, and 4 in the Frame Control field set to 0, 1, and 0, respectively); therefore, QSTAs cannot be treated as CF-Pollable STAs. This requires a QAP that provides non-QoS CF-polling to adhere to frame sequence restrictions considerably more complex than, and less efficient than, those specified for either PCF or HCF. In addition, the achievable service quality is likely to be degraded when nQSTAs are associated and being polled.

The HCF protects the transmissions during each CAP using the virtual CS mechanism.

A QSTA may initiate multiple frame exchange sequences during a polled TXOP of sufficient duration to perform more than one such sequence. The use of virtual CS by the HC provides improved protection of the CFP, in addition to the protection provided by having all STAs in the BSA setting their NAVs to dot11CFPMaxDuration at the target beacon transmission time (TBTT) of DTIM Beacon frames.

The operation rules of the HCCA are defined in 9.9.2.

Change the following subclause number due to the insertion above of the new subclause, 9.1.3, and change the heading and text as shown:

9.1.4 ~~9.1.3~~ Coexistence of DCF, ~~and~~ PCF, and HCF

The DCF and ~~the~~ a centralized coordination function (either PCF or HCF) shall coexist in a manner that permits both to operate concurrently within the same BSS. When a PC is operating in a BSS, the ~~two~~ PCF and

DCF access methods alternate, with a contention-free period (CFP) followed by a contention period (CP). This is described in greater detail in 9.3. When an HC is operating in a QBSS, it may generate an alternation of CFP and CP in the same way as a PC, using the DCF access method only during the CP. The HCF access methods (controlled and contention-based) operate sequentially when the channel is in CP. Sequential operation allows the polled and contention-based access methods to alternate, within intervals as short as the time to transmit a frame exchange sequence, under rules defined in 9.9.

Change the following subclause number due to the insertion above of the new subclause, 9.1.3, and change the text as shown (note that Figure 48 remains unchanged):

9.1.5 ~~9.1.4~~ Fragmentation/defragmentation overview

The process of partitioning a MSDU or an MMPDU into smaller MAC level frames, MPDUs, is called *fragmentation*. Fragmentation creates MPDUs smaller than the original MSDU or MMPDU length to increase reliability, by increasing the probability of successful transmission of the MSDU or MMPDU in cases where channel characteristics limit reception reliability for longer frames. QSTAs may use fragmentation to use the medium efficiently in consideration of the duration available in granted TXOPs, as long as the rules in 9.4 are followed. Fragmentation is accomplished at each immediate transmitter. The process of recombining MPDUs into a single MSDU or MMPDU is defined as *defragmentation*. Defragmentation is accomplished at each immediate recipient.

Only MPDUs with a unicast receiver address shall be fragmented. Broadcast/multicast frames shall not be fragmented even if their length exceeds ~~a~~dot11FragmentationThreshold.

When a directed MSDU is received from the LLC or a directed MMPDU is received from the MAC sublayer management entity (MLME) with a length greater than ~~a~~dot11FragmentationThreshold, the MSDU or MMPDU shall be fragmented. The MSDU or MMPDU is divided into MPDUs. Each fragment is a frame no larger than ~~a~~dot11FragmentationThreshold. It is possible that any fragment may be a frame smaller than ~~a~~dot11FragmentationThreshold. An illustration of fragmentation is shown in Figure 48.

The MPDUs resulting from the fragmentation of an MSDU or MMPDU are sent as independent transmissions, each of which is separately acknowledged. This permits transmission retries to occur per fragment, rather than per MSDU or MMPDU. Unless interrupted due to medium occupancy limitations for a given PHY or TXOP limitations for QSTA, the fragments of a single MSDU or MMPDU are sent as a burst during the CP, using a single invocation of the DCF or EDCA medium access procedure. The fragments of a single MSDU or MMPDU are either

— Sent during a CFP as individual frames obeying the rules of the PC medium access procedure or

— Sent as a burst in an EDCA or HCCA TXOP.

Change the following subclause number due to the insertion above of the new subclause, 9.1.3, and change the text as shown:

9.1.6 ~~9.1.5~~ MAC data service

The MAC data service provides the transport of MSDUs between MAC peer entities as characterized in 6.1.1.

The transmission process is started by receipt of an MA-UNITDATA.request primitive containing an MSDU and the associated parameters. This may cause one or more data MPDUs containing the MSDU to be transmitted following fragmentation and security encapsulation, as appropriate.

The MA-UNITDATA.indication primitive is generated in response to one or more received data MPDUs containing an MSDU following validation, address filtering, decryption, decapsulation, and defragmentation, as appropriate.

~~The MAC Data Service shall translate MAC service requests from LLC into input signals utilized by the MAC State Machines. The MAC Data Service shall also translate output signals from the MAC State Machines into service indications to LLC. The translations are given in the MAC Data Service State Machine defined in Annex C.~~

The MAC data service for QSTAs shall incorporate a TID with each MA-UNITDATA.request service. This TID associates the MSDU with the AC or TS queue for the indicated traffic.

9.2 DCF

9.2.3 Interframe space (IFS)

Change the text and Figure 49 in 9.2.3 as follows:

The time interval between frames is called the *IFS*. A STA shall determine that the medium is idle through the use of the CS function for the interval specified. ~~Four~~Five different IFSs are defined to provide priority levels for access to the wireless media. Figure 49 shows some of these relationships.

a) SIFS short interframe space
b) PIFS PCF interframe space
c) DIFS DCF interframe space
d) AIFS arbitration interframe space (used by the QoS facility)
e) EIFS extended interframe space

The different IFSs shall be independent of the STA bit rate. The IFS timings are ~~shall be~~ defined as time gaps on the medium, and the IFS timings except AIFS are~~shall be~~ fixed for each PHY (even in multirate-capable PHYs). The IFS values are determined from attributes specified by the PHY.

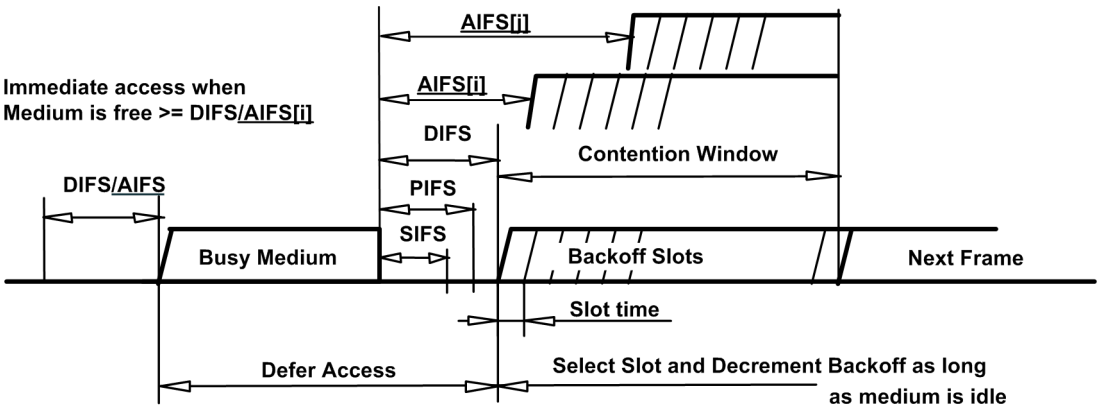


Figure 49—Some IFS relationships

Insert after 9.2.3.3 the following new subclause as 9.2.3.4 (instructions for renumbering the existing 9.2.3.4 are given after this new text):

9.2.3.4 Arbitration IFS (AIFS)

The AIFS shall be used by QSTAs to transmit all data frames (MPDUs), all management frames (MMPDUs), and the following control frames: PS-Poll, RTS, CTS (when not transmitted as a response to the RTS), BlockAckReq, and BlockAck (when not transmitted as a response to the BlockAckReq). A QSTA using the EDCA shall obtain a TXOP for an AC if the QSTA's CS mechanism (see 9.2.1) determines that the medium is idle at the AIFS[AC] slot boundary (see 9.9.1.3), after a correctly received frame, and the backoff time for that AC has expired.

A QSTA using the EDCA shall not transmit within an EIFS-DIFS+AIFS[AC] plus any backoff time after that QSTA determines that the medium is idle following reception of a frame for which the PHYRXEND.indication primitive reported an error or a frame for which the MAC FCS value was not correct, unless a subsequent reception of an error-free frame resynchronizes the STA. This resynchronization allows the QSTA to transmit using the AIFS[AC] following that subsequent frame, provided that the backoff time for that AC has expired.

A non-AP QSTA computes the time periods for each AIFS[AC] from the dot11EDCATableAIFSN attributes in the MIB. QSTAs update their dot11EDCATableAIFSN values using information in the most recent EDCA Parameter Set element of Beacon frames received from the QAP of the QBSS (see 7.3.2.27). A QAP computes the time periods for each AIFS[AC] from the dot11QAPEDCATableAIFSN attributes in its MIB.

Change the following subclause number due to the insertion above of the new subclause, 9.2.3.4, and change the text as shown:

9.2.3.5 ~~9.2.3.4~~ EIFS

The EIFS shall be used by the DCF, and the EIFS-DIFS+AIFS[AC] shall be used by the EDCA mechanism under HCF, when the PHY has indicated to the MAC that a frame transmission was begun that did not result in the correct reception of a complete MAC frame with a correct FCS value. The duration of an EIFS is defined in 9.2.10. The EIFS interval shall begin following indication by the PHY that the medium is idle after detection of the erroneous frame, without regard to the virtual CS mechanism. The EIFS is defined to provide enough time for another STA to acknowledge what was, to this STA, an incorrectly received frame before this STA commences transmission. Reception of an error-free frame during the EIFS resynchronizes the STA to the actual busy/idle state of the medium, so the EIFS is terminated and normal medium access (using DIFS or AIFS as appropriate and, if necessary, backoff) continues following reception of that frame.

9.2.5 DCF access procedure

9.2.5.2 Backoff procedure for DCF

Insert the following text at the beginning of 9.2.5.2:

This subclause describes backoff procedure that is to be invoked when DCF is used. For the backoff procedure when EDCA is used, see 9.9.1.5.

9.2.9 Duplicate detection and recovery

Change the text in 9.2.9 as follows:

Because MAC-level acknowledgments and retransmissions are incorporated into the protocol, there is the possibility that a frame may be received more than once. Such duplicate frames shall be filtered out within the ~~destination~~receiver MAC.

Duplicate frame filtering is facilitated through the inclusion of a Sequence Control field (consisting of a sequence number and fragment number) within data and management frames as well as TID subfield in the QoS Control field within QoS data frames. MPDUs that are part of the same MSDU shall have the same sequence number, and different MSDUs shall (with a high probability) have a different sequence number.

The sequence number, for management frames and for data frames with QoS subfield set to 0, is generated by the transmitting STA as an incrementing sequence of integers. In a QSTA, the sequence numbers for QoS (+)Data frames are generated by different counters for each TID and receiver pair and shall be incremented by one for each new MSDU corresponding to the TID/receiver pair.

The receiving STA shall keep a cache of recently received <Address 2, sequence-number, fragment-number> tuples. The receiving QSTA shall also keep a cache of recently received <Address 2, TID, sequence-number, fragment-number> tuples for all QSTAs from whom it has received QoS data frames. A receiving STA is required to keep only the most recent cache entry per <Address 2, sequence-number> pair, storing only the most recently received fragment number for that pair. A receiving QSTA is also required to keep only the most recent cache entry per <Address 2, TID, sequence-number> triple, storing only the most recently received fragment number for that triple. A receiving STA may omit tuples obtained from broadcast/multicast or ATIM frames from the cache.

A ~~destination~~non-QoS receiver STA shall reject as a duplicate frame any frame that has the Retry bit set in the Frame Control field and that matches an <Address 2, sequence-number, ~~and~~ fragment-number> tuple of an entry in the cache. A receiver QSTA shall also reject as a duplicate frame any frame that has the Retry bit set in the Frame Control field and that matches an <Address 2, TID, sequence-number, fragment-number> tuple of an entry in the cache.

There is a small possibility that a frame may be improperly rejected due to such a match; however, this occurrence would be rare and simply results in a lost frame (similar to an FCS error in other LAN protocols).

The ~~destination~~receiver STA shall perform the acknowledgment procedure on all successfully received frames requiring acknowledgment, even if the frame is discarded due to duplicate filtering.

9.6 Multirate support

Change the text in 9.6 as follows:

Some PHYs have multiple data transfer rate capabilities that allow implementations to perform dynamic rate switching with the objective of improving performance. The algorithm for performing rate switching is beyond the scope of this standard, but in order to ensure coexistence and interoperability on multirate-capable PHYs, this standard defines a set of rules that shall be followed by all STAs.

~~All c~~Control frames that initiate a frame exchange shall be transmitted at one of the rates in the BSSBasicRateSet parameter~~, unless~~ except in the following cases:

— When the transmitting STA's protection mechanism is enabled~~,~~ and the control frame is a protection mechanism frame

— When the control frame is a BlockAckReq or BlockAck frame

~~;~~ i~~I~~n ~~which~~ the former case, the control frame shall be transmitted at a rate according to the separate rules for determining the rates of transmission of protection frames in ~~9.10~~9.13. In the latter case, the control frame shall be transmitted in accordance with this subclause.

All frames with multicast and broadcast in the Address 1 field that have a UP of zero shall be transmitted at one of the rates included in the BSS basic rate set, regardless of their type or subtype.

All data frames of subtype (QoS) (+)CF-Poll sent in the CP shall be transmitted at one of the rates in the BSS basic rate set so that they will be understood by all STAs in the BSS, unless an RTS/CTS exchange has already been performed before the transmission of the data frame of subtype CF-Poll and the Duration field in the RTS frame covers the entire TXOP. All other d~~D~~ata, BlockAckReq, and BlockAck frames and/or management MPDUs with unicast in the Address 1 field shall be sent using~~on~~ any ~~supported~~ data rate subject to the following constraints: ~~selected by the rate switching mechanism~~. No STA shall transmit a unicast frame at a rate that is not supported by the receiver~~destination~~ STA, as reported in any Supported Rates and Extended Supported Rates element in the management frames. For frames of type (QoS) Data+CF-Ack, (QoS) Data+CF-Poll+CF-Ack, and (QoS) CF-Poll+CF-Ack, the rate chosen to transmit the frame should be supported by both the addressed recipient STA and the STA to which the ACK frame is intended. The BlockAck control frame shall be sent at the same rate as the BlockAckReq frame if it is sent in response to a BlockAckReq frame.

Under no circumstances shall a STA initiate transmission of a data or management frame at a data rate higher than the greatest rate in the OperationalRateSet, a parameter of the MLME-JOIN.request primitive.

To allow the transmitting STA to calculate the contents of the Duration/ID field, a STA responding to a received frame shall transmit its Control Response frame (either CTS or ACK) ~~frames~~, other than the BlockAck control frame, at the highest rate in the BSSBasicRateSet parameter that is less than or equal to the rate of the immediately previous frame in the frame exchange sequence (as defined in ~~9.7~~9.12) and that is of the same modulation type as the received frame. If no rate in the basic rate set meets these conditions, then the control frame sent in response to a received frame shall be transmitted at the highest mandatory rate of the PHY that is less than or equal to the rate of the received frame, and that is of the same modulation type as the received frame. In addition, the Control Response frame shall be sent using the same PHY options as the received frame, unless they conflict with the requirement to use the BSSBasicRateSet parameter.

An alternative rate for the control response frame may be used, provided that the duration of the control response frame at the alternative rate is the same as the duration of the control response frame at the originally chosen rate and the alternative rate is in either the BSSBasicRateSet parameter or the mandatory rate set of the PHY and the modulation of the control response frame at the alternative rate is the same type as that of the received frame.

For the Clause 17, Clause 18 and Clause 19 PHYs, the time required to transmit a frame for use in the Duration/ID field is determined using the PLME-TXTIME.request primitive (see 10.4.6) and the PLME-TXTIME.confirm primitive (see 10.4.7), both defined in 17.4.3, 18.3.4, 19.8.3.1, 19.8.3.2, or 19.8.3.3 depending on the PHY options. In QSTAs, the Duration/ID field may cover multiple frames and may involve using the PLME-TXTIME.request primitive several times.

Change the subclause order by moving 9.7 (Frame exchange sequences) to become 9.12 (see page 101), change the following subclause number due to the change in subclause order, and change the text as shown:

9.7 ~~9.8~~ MSDU transmission restrictions

To avoid reordering MSDUs between pairs of LLC entities and/or unnecessarily discarding MSDUs, the following restrictions shall be observed by any STA that is able to concurrently process multiple outstanding

MSDUs for transmission. Note that here the term "outstanding" refers to an MSDU or MMPDU that is eligible to be transmitted at a particular time. A STA may have any number (greater than or equal to one) of eligible MSDUs outstanding concurrently, subject to the restrictions below.

~~The~~ An nQSTA shall ensure that no more than one MSDU or MMPDU from a particular SA to a particular individual RA is outstanding at a time. Note that a simpler, more restrictive invariant to maintain is that no more than one MSDU with a particular individual RA may be outstanding at a time.

For all transmissions not using the acknowledgment policy of Block Ack, a QSTA shall ensure that no more than one MSDU or MMPDU with a particular TID from a particular SA to a particular individual RA is outstanding at any time. Note that a simpler, more restrictive invariant to maintain is that no more than one MSDU with any particular TID with a particular individual RA may be outstanding at any time. This restriction is not applicable for MSDUs that are to be transmitted using the Block Ack mechanism.

In a STA where the optional StrictlyOrdered service class has been implemented, that STA shall ensure that there is no group-addressed (multidestination) MSDU of the StrictlyOrdered service class outstanding from the SA of any other outstanding MSDU (either directed or group-addressed). This is because a group-addressed MSDU is implicitly addressed to a collection of peer STAs that could include any individual receiver address.

It is recommended that the STA select a value of aMaxMSDUTransmitLifetime that is sufficiently large that the STA does not discard MSDUs due to excessive Transmit MSDU timeouts under normal operating conditions.

For MSDUs belonging to the service class of QoSAck when the receiver is a QSTA, the QoS data frames that are used to send these MSDUs shall have the Ack Policy subfield in the QoS Control field set to Normal Ack or Block Ack. For MSDUs belonging to the service class of QoSNoAck when the receiver is a QSTA, the QoS data frames that are used to send these MSDUs shall have the Ack Policy subfield in the QoS Control field set to No Ack.

Change the following subclause number due to the change above in subclause order:

9.8 ~~9.9~~ Operation across regulatory domains

Insert after 9.8 the following new subclauses as 9.9 through 9.9.3.2, including the new figures in those subclauses:

9.9 HCF

This clause describes the QoS enhancements to the MAC functional description. QSTAs may access the channel in a more controlled manner, compared to a non-QSTA, to transmit MPDUs.

Under HCF, 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. The TXOP may be obtained by a QSTA winning an instance of EDCA contention (see 9.9.1) during the CP or by a non-AP QSTA receiving a QoS (+)CF-Poll frame (see 9.9.2) during the CP or CFP. The former is called *EDCA TXOP*, while the latter is called *HCCA TXOP* or *polled TXOP*. An HCCA TXOP shall not extend across a TBTT. A TXOP shall not exceed dot11MaxDwellTime (if using an FH PHY). The occurrence of a TBTT implies the end of the HCCA TXOP, after which the regular channel access procedure (EDCA or HCCA) is resumed. It is possible that no frame was transmitted during the TXOP. The foreshortened termination of the HCCA TXOP does not imply an error condition.

9.9.1 HCF contention-based channel access (EDCA)

9.9.1.1 Reference Implementation

The channel access protocol is derived from the DCF procedures described in 9.2.

A model of the reference implementation is shown in Figure 62a and illustrates a mapping from frame type or UP to AC: the four transmit queues and the four independent EDCAFs, one for each queue. The mapping of UP to the AC is described in 9.1.3.1 and Table 20i. The mapping of frame types to ACs is described in 9.1.3.1.

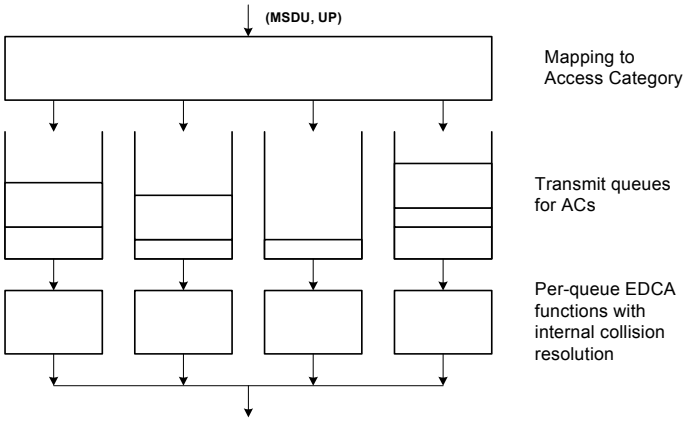


Figure 62a—Reference implementation model

9.9.1.2 EDCA TXOPs

There are two modes of EDCA TXOP defined, the initiation of the EDCA TXOP and the multiple frame transmission within an EDCA TXOP. An initiation of the TXOP occurs when the EDCA rules permit access to the medium. A multiple frame transmission within the TXOP occurs when an EDCAF retains the right to access the medium following the completion of a frame exchange sequence, such as on receipt of an ACK frame.

The TXOP limit duration values are advertised by the QAP in the EDCA Parameter Set information element in Beacon and Probe Response frames transmitted by the QAP. A TXOP limit value of 0 indicates that a single MSDU or MMPDU, in addition to a possible RTS/CTS exchange or CTS to itself, may be transmitted at any rate for each TXOP.

Non-AP QSTAs shall ensure that the duration of TXOPs obtained using the EDCA rules do not exceed the TXOP limit. The duration of a TXOP is the duration during which the TXOP holder maintains uninterrupted control of the medium, and it includes the time required to transmit frames sent as an immediate response to the TXOP holder's transmissions.

A QSTA shall fragment a unicast MSDU so that the transmission of the first MPDU of the TXOP does not cause the TXOP limit to be exceeded at the PHY rate selected for the initial transmission attempt of that MPDU. The TXOP limit may be exceeded, when using a lower PHY rate than selected for the initial transmission attempt of the first MPDU, for a retransmission of an MPDU, for the initial transmission of an MPDU if any previous MPDU in the current MSDU has been retransmitted, or for broadcast/multicast MSDUs. When the TXOP limit is exceeded due to the retransmission of an MPDU at a reduced PHY rate, the STA shall not transmit more than one MPDU in the TXOP.

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 QAP may send broadcast/multicast frames without using any protection mechanism. In a QIBSS, broadcast/multicast frames shall be sent one at a time, and backoff shall be performed after the transmission of each of the broadcast/multicast frames.

9.9.1.3 Obtaining an EDCA TXOP

Each channel access timer shall maintain a backoff function (timer), which has a value measured in backoff slots.

The duration AIFS[AC] is a duration derived from the value AIFSN[AC] by the relation

AIFS[AC] = AIFSN[AC] × aSlotTime + aSIFSTime.

The value of AIFSN[AC] shall be greater than or equal to 2 for non-AP QSTAs and is advertised by the QAP in the EDCA Parameter Set information element in Beacon and Probe Response frames transmitted by the QAP. The value of AIFSN[AC] shall be greater than or equal to 1 for QAPs. An EDCA TXOP is granted to an EDCAF when the EDCAF determines that it shall initiate the transmission of a frame exchange sequence. Transmission initiation shall be determined according to the following rules:

On specific slot boundaries, each EDCAF shall make a determination to perform one and only one of the following functions:

— Initiate the transmission of a frame exchange sequence for that access function.

— Decrement the backoff timer for that access function.

— Invoke the backoff procedure due to an internal collision.

— Do nothing for that access function.

The specific slot boundaries at which exactly one of these operations shall be performed are defined as follows, for each EDCAF:

a) Following AIFSN[AC] × aSlotTime – aRxTxTurnaroundTime of idle medium after SIFS (not necessarily idle medium during the SIFS duration) after the last busy medium on the antenna that was the result of a reception of a frame with a correct FCS.

b) Following EIFS – DIFS + AIFSN[AC] × aSlotTime + aSIFSTime – aRxTxTurnaroundTime of idle medium after the last indicated idle medium as determined by the physical CS mechanism that was the result of a frame reception that has resulted in FCS error, or PHY-RXEND.indication (RXERROR) primitive where the value of RXERROR is not NoError.

c) When any other EDCAF at this QSTA transmitted a frame requiring acknowledgment, the earlier of

   1) The end of the ACK-Timeout interval timed from the PHY_TXEND.confirm primitive, followed by AIFSN[AC] x aSlotTime + aSIFSTime – aRxTxTurnaroundTime of idle medium, and

   2) The end of the first AIFSN[AC] × aSlotTime – aRxTxTurnaroundTime of idle medium after SIFS (not necessarily medium idle during the SIFS duration, the start of the SIFS duration implied by the length in the PLCP header of the previous frame) when a PHY-RXEND.indication primitive occurs as specified in 9.2.8.

d) Following AIFSN[AC] × aSlotTime – aRxTxTurnaroundTime of idle medium after SIFS (not necessarily medium idle during the SIFS duration) after the last busy medium on the antenna that was the result of a transmission of a frame for any EDCAF and which did not require an acknowledgment.

e) Following AIFSN[AC] × aSlotTime + aSIFSTime – aRxTxTurnaroundTime of idle medium after the last indicated idle medium as indicated by the CS mechanism that is not covered by a) through d).

f) Following aSlotTime of idle medium, which occurs immediately after any of these conditions, a) through f), is met for the EDCAF.

At each of the above-described specific slot boundaries, each EDCAF shall initiate a transmission sequence if

— There is a frame available for transmission at that EDCAF, and

— The backoff timer for that EDCAF has a value of zero, and

— Initiation of a transmission sequence is not allowed to commence at this time for an EDCAF of higher UP.

At each of the above-described specific slot boundaries, each EDCAF shall decrement the backoff timer if the backoff timer for that EDCAF has a nonzero value.

At each of the above-described specific slot boundaries, each EDCAF shall invoke the backoff procedure due to an internal collision if

— There is a frame available for transmission at that EDCAF, and

— The backoff timer for that EDCAF has a value of zero, and

— Initiation of a transmission sequence is allowed to commence at this time for an EDCAF of higher UP.

At each of the above-described specific slot boundaries, an EDCAF shall do nothing if none of the above actions is taken.

An example showing the relationship between AIFS, AIFSN, DIFS, and slot times immediately following a medium busy condition (and assuming that medium busy condition was not caused by a frame in error) is shown in Figure 62b. In this case, with AIFSN = 2, the EDCAF may decrement the backoff counter for the first time at 2 × aSlotTime following the end of the medium busy condition (end of the medium busy condition happens at the end of M1 in Figure 62b). If, in this example, the backoff counter contained a value of 1 at the time the medium became idle, transmission would start as a result of an EDCA TXOP on-air at a time

aSIFSTime + 3 × aSlotTime

following the end of the medium busy condition.

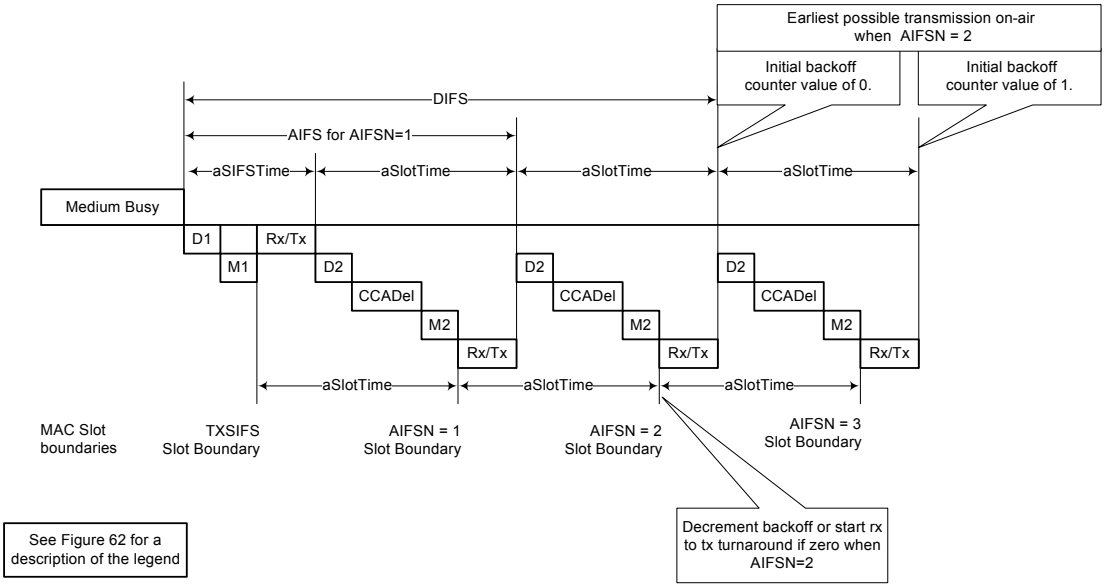


Figure 62b—EDCA mechanism timing relationships

9.9.1.4 Multiple frame transmission in an EDCA TXOP

Multiple frames may be transmitted in an acquired EDCA TXOP following the rules in 9.9.1.3 if there are more than one frame pending in the AC for which the channel has been acquired. However, those frames that are pending in other ACs shall not be transmitted in this EDCA TXOP. If a QSTA has in its transmit queue an additional frame of the same AC as the one just transmitted and the duration of transmission of that frame plus any expected acknowledgment for that frame is less than the remaining medium occupancy timer value, then the QSTA may commence transmission of that frame at SIFS after the completion of the immediately preceding frame exchange sequence. The intention of using the multiple frame transmission shall be indicated by the QSTA through the setting of the duration/ID values in one of the following two ways (see 7.1.4):

a) Long enough to cover the response frame, the next frame, and its response frame.

b) Long enough to cover the transmission of a burst of MPDUs subject to the limit set by dot11EDCATableTXOPLimit.

If the Duration/ID field is set for multiple frame transmission and there is a transmission failure, the corresponding channel access function may recover before the expiry of the NAV setting due to the setting of the Duration/ID field in the frame that resulted in a transmission failure. The backoff procedure is described in 9.9.1.5. However, at the expiry of the NAV set by the frame that resulted in a transmission failure, if the channel access function has not recovered, then the EDCAF shall invoke backoff procedure.

No other AC at the QSTA shall transmit before the expiry of the NAV set by the frame that resulted in a transmission failure. All other ACs at the QSTA shall treat the medium as busy until the expiry of the NAV set by the frame that resulted in a transmission failure, just as they would if they had received that transmission from another QSTA.

A frame exchange may be a multicast frame, a frame transmitted with No Ack policy (for which there is no expected acknowledgment), or a unicast frame followed by a correctly received ACK frame transmitted by either a non-AP QSTA or a QAP.

Note that, as for an EDCA TXOP, a multiple frame transmission is granted to an EDCAF, not to a non-AP QSTA or QAP, so that the multiple frame transmission is permitted only for the transmission of a frame of the same AC as the frame that was granted the EDCA TXOP.

9.9.1.5 EDCA backoff procedure

Each EDCAF shall maintain a state variable CW[AC], which shall be initialized to the value of the parameter CWmin[AC].

If a frame is successfully transmitted by a specific EDCAF, indicated by the successful reception of a CTS in response to an RTS, the successful reception of an ACK frame in response to a unicast MPDU or BlockAck, the successful reception of a BlockAck or ACK frame in response to a BlockAckReq frame, or the transmission of a multicast frame or a frame with No Ack policy, CW[AC] shall be reset to CWmin[AC].

The backoff procedure shall be invoked for an EDCAF when any of the following events occurs:

— A frame with that AC is requested to be transmitted, the medium is busy as indicated by either physical or virtual CS, and the backoff timer has a value of zero for that AC.

— The final transmission by the TXOP holder initiated during the TXOP for that AC was successful.

— The transmission of a frame of that AC fails, indicated by a failure to receive a CTS in response to an RTS, a failure to receive an ACK frame that was expected in response to a unicast MPDU, or a failure to receive a BlockAck or ACK frame in response to a BlockAckReq frame.

— The transmission attempt collides internally with another EDCAF of an AC that has higher priority, that is, two or more EDCAFs in the same QSTA are granted a TXOP at the same time.

If the backoff procedure is invoked for reason a) above, the value of CW[AC] shall be left unchanged. If the backoff procedure is invoked because of reason b) above, the value of CW[AC] shall be reset to CWmin[AC].

If the backoff procedure is invoked because of a failure event [either reason c) or d) above], the value of CW[AC] shall be updated as follows before invoking the backoff procedure:

a) If the QSRC[AC] or the QLRC[AC] for the QSTA has reached dot11ShortRetryLimit or dot11LongRetryLimit respectively, CW[AC] shall be reset to CWmin[AC].

b) Otherwise,

   1) If CW[AC] is less than CWmax[AC], CW[AC] shall be set to the value (CW[AC] + 1)*2 – 1.

   2) If CW[AC] is equal to CWmax[AC], CW[AC] shall remain unchanged for the remainder of any retries.

The backoff timer is set to an integer value chosen randomly with a uniform distribution taking values in the range [0,CW[AC]] inclusive.

All backoff slots occur following an AIFS[AC] period during which the medium is determined to be idle for the duration of the AIFS[AC] period, or following an EIFS – DIFS + AIFS[AC] period during which the medium is determined to be idle for the duration of the EIFS – DIFS + AIFS[AC] period following detection of a frame that was not received correctly.

9.9.1.6 Retransmit procedures

QSTAs shall maintain a short retry counter and a long retry counter for each MSDU or MMPDU that belongs to a TC requiring acknowledgment. The initial value for the short and long retry counters shall be zero. QSTAs also maintain a short retry counter and a long retry counter for each AC. They are defined as QSRC[AC] and QLRC[AC], respectively, and each is initialized to a value of zero.

After transmitting a frame that requires acknowledgment, the QSTA shall perform the acknowledgment procedure, as defined in 9.2.8. The short retry count for an MSDU or MMPDU and the QSRC[AC] shall be incremented every time transmission of a MAC frame of length less than or equal to dot11RTSThreshold fails for that MSDU or MMPDU. This short retry count and the QSTA QSRC[AC] shall be reset when a MAC frame of length less than or equal to dot11RTSThreshold succeeds for that MSDU or MMPDU. The long retry count for an MSDU or MMPDU and the QLRC[AC] shall be incremented every time transmission of a MAC frame of length greater than dot11RTSThreshold fails for that MSDU or MMPDU. This long retry count and the QLRC[AC] shall be reset when a MAC frame of length greater than dot11RTSThreshold succeeds for that MSDU or MMPDU. All retransmission attempts for an MPDU that has failed the acknowledgment procedure one or more times shall be made with the Retry field set to 1 in the data or management frame.

Retries for failed transmission attempts shall continue until the short retry count for the MSDU or MMPDU is equal to dot11ShortRetryLimit or until the long retry count for the MSDU or MMPDU is equal to dot11LongRetryLimit. When either of these limits is reached, retry attempts shall cease, and the MSDU or MMPDU shall be discarded.

For internal collisions occurring with the EDCA access method, the appropriate retry counters (short retry counter for MSDU or MMPDU and QSRC[AC] or long retry counter for MSDU or MMPDU and QLRC[AC]) are incremented. For transmissions that use Block Ack, the rules in 9.10.3 also apply. QSTAs shall retry failed transmissions until the transmission is successful or until the relevant retry limit is reached.

With the exception of a frame belonging to a TID for which Block Ack is set up, a QSTA shall not initiate the transmission of any management or data frame to a specific RA while the transmission of another management or data frame with the same RA and having been assigned its sequence number from the same sequence counter has not yet completed to the point of success, retry fail, or other MAC discard (e.g., lifetime expiry).

QSTAs shall maintain a transmit MSDU timer for each MSDU passed to the MAC. The MIB attribute dot11EDCATableMSDULifetime specifies the maximum amount of time allowed to transmit an MSDU for a given AC. The transmit MSDU timer shall be started when the MSDU is passed to the MAC. If the value of this timer exceeds the appropriate entry in dot11EDCATableMSDULifetime, then the MSDU, or any remaining, undelivered fragments of that MSDU, shall be discarded by the source QSTA without any further attempt to complete delivery of that MSDU.

9.9.2 HCCA

The HCCA mechanism manages access to the WM, using a HC that has higher medium access priority than non-AP STAs. This allows it to transfer MSDUs to non-AP QSTAs and to allocate TXOPs to non-AP QSTAs.

The HC is a type of centralized coordinator, but differs from the PC used in PCF in several significant ways, although it may optionally implement the functionality of a PC. Most important is that HCF frame exchange sequences may be used among QSTAs associated in a QBSS during both the CP and the CFP. Another significant difference is that the HC grants a non-AP QSTA a polled TXOP with duration specified in a QoS (+)CF-Poll frame. Non-AP QSTAs may transmit multiple frame exchange sequences within given polled TXOPs, subject to the limit on TXOP duration.

All STAs inherently obey the NAV rules of the HCF because each frame transmitted under HCF by the HC or by a non-AP QSTA contains a duration value chosen to cause STAs in the BSS to set their NAVs to protect the expected subsequent frames.

All non-AP QSTAs shall be able to respond to QoS (+)CF-Poll frames received from an HC with the Address 1 field matching their own addresses.

The HC shall perform delivery of buffered broadcast and multicast frames following DTIM beacons. The HC may also operate as a PC, providing (non-QoS) CF-Polls to associated CF-Pollable STAs using the frame formats, frame exchange sequences, and other applicable rules for PCF specified in 9.3.[10]

An HC may perform a backoff following an interruption of a frame exchange sequence due to lack of an expected response under the rules described in 9.9.2.1.3, using the parameters dot11HCCWmin, dot11HCCWmax, and dot11HCCAIFSN and the backoff rules in 9.1 and 9.9.1.5. The decision to perform a backoff by the HC is dependent on conditions such as interference from an overlapping BSS. The mechanism to detect the interference from an overlapping BSS and the decision to perform a backoff, DFS (such as in 11.6), or other techniques (such as inter-BSS scheduling) is beyond the scope of this amendment.

9.9.2.1 HCCA procedure

The HC gains control of the WM as needed to send QoS traffic to non-AP QSTAs and to issue QoS (+)CF-Poll frames to non-AP QSTAs by waiting a shorter time between transmissions than the STAs using the EDCA procedures. The duration values used in QoS frame exchange sequences reserve the medium to permit completion of the current sequence.

The HC may include a CF Parameter Set element in the Beacon frames it generates. This causes the BSS to appear to be a point-coordinated BSS to STAs. This causes STAs to set their NAVs to the CFPDurRemaining value in the CF Parameter Set element value at TBTT, as specified in 9.3.3.2. This prevents most contention in the CFP by preventing nonpolled transmissions by non-AP STAs whether or not they are CF-Pollable.

9.9.2.1.1 CFP generation

The HC may function as a PC that uses the CFP for delivery, generating a CFP as shown in Figure 59, with the restriction that the CFP initiated by an HC shall always end with a CF-End frame. The HC may also issue QoS (+)CF-Poll frames to associated non-AP QSTAs during the CFP. However, because the HC can also grant polled TXOPs, by sending QoS (+)CF-Poll frames, during the CP, it is not mandatory for the HC to use the CFP for QoS data transfers.

Only a QAP that also issues non-QoS CF-Poll frames to associated CF-Pollable STAs may end a CFP with a CF-End+CF-Ack frame and only when the CF-End+CF-Ack is acknowledging a reception from a CF-Pollable nQSTA. The use of a non-QoS CF-Poll frame by a QAP to a non-AP QSTA is deprecated (for further discussion, see 7.3.1.4).

9.9.2.1.2 CAP generation

When the HC needs access to the WM to start a CFP or a TXOP in CP, the HC shall sense the WM. When the WM is determined to be idle for one PIFS period, the HC shall transmit the first frame of any permitted frame exchange sequence, with the duration value set to cover the CFP or the TXOP. The first permitted frame in a CFP after a TBTT is the Beacon frame. CAPs along with the CFPs and the CPs are illustrated in Figure 62c.

After the last frame of all other nonfinal frame exchange sequences (e.g., sequences that convey unicast QoS data or management frames) during a TXOP, the holder of the current TXOP shall wait for one SIFS period before transmitting the first frame of the next frame exchange sequence. The HC may sense the channel and reclaim the channel after a duration of PIFS after the TXOP, if the channel remains idle. A CAP ends when the HC does not reclaim the channel after a duration of PIFS after the end of a TXOP.

[10]Attempting to intersperse HCF frame exchange sequences and PCF frame exchange sequences in a single CFP can be extremely complex.

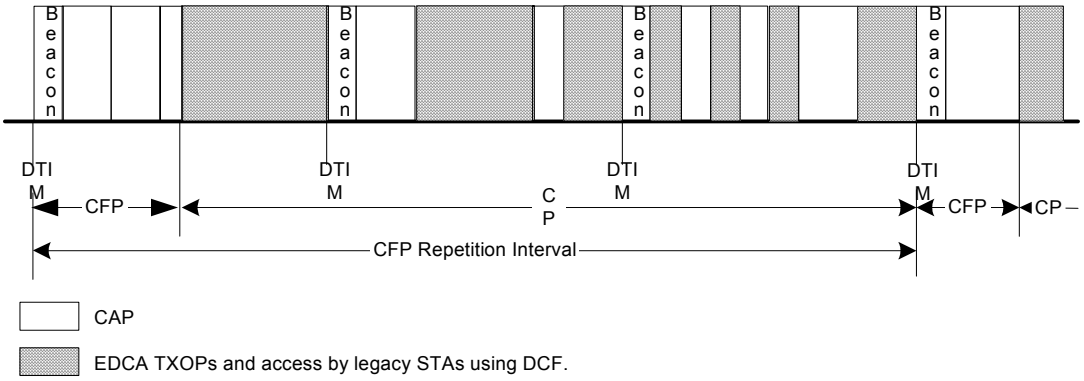


Figure 62c—CAP/CFP/CP periods

9.9.2.1.3 Recovery from the absence of an expected reception

This subclause describes recovery from the absence of an expected reception in a CAP. It should be noted that the recovery rules from the absence of an expected reception are different from EDCA because in this case the NAVs of all the STAs in the QBSS have already been set up by the transmissions by the HC. The recovery rules for the multiple frame transmission are different because a non-AP STA may always be hidden and may have not set its NAV due to the transmission by another non-AP STA. Finally, since an HC is collocated with the QAP, the QAP may recover using the rules described in this subclause even if the recovery is from the absence of an expected reception.

If the beginning of reception of an expected response, as detected by the occurrence of PHY-CCA.indication(busy) primitive at the QSTA that is expecting the response, does not occur during the first slot time following SIFS, then[11]

a) If the transmitting STA is the HC, it may initiate recovery by transmitting at a PIFS after the end of the HC's last transmission only if PHY-CCA.indication primitive is clear.

b) If the transmitting STA is a non-AP QSTA, it shall initiate recovery by transmitting at a PIFS after the end of the last transmission, if the polled TXOP limit is greater than 0 and at least one frame (re)transmissions can be completed within the remaining duration of a nonzero polled TXOP limit.

If the transmitted frame is not of type QoS (+)CF-Poll and the expected response frame is not received correctly, regardless of the occurrence of the PHY-RXSTART.indication primitive, the QSTA may initiate recovery following the occurrence of PHY-CCA.indication(idle) primitive so that a SIFS time interval occurs between the last energy on the air and the transmission of the recovery frame.

When there is a transmission failure within a polled TXOP, the frame retry counter corresponding to the AC of the failed frame shall be incremented. An MPDU belonging to a TC is subject to the respective retry limit as well as the dot11EDCATableMSDULifetime and is discarded when either of them is exceeded. An MPDU belonging to a TS with a specified delay bound is subject to delay bound and is discarded if the MPDU could not be transmitted successfully since it has been delivered to the MAC. An MPDU belonging to a TS with an unspecified delay is subject to dot11MaxTransmitMSDULifetime and is discarded when it is exceeded.

Non-AP QSTAs that receive a QoS (+)CF-Poll frame shall respond within a SIFS period, regardless of the NAV setting. If a response is not received but a PHY-CCA.indication(busy) primitive occurs during the slot following SIFS and is followed by a PHY-RXSTART.indication or PHY-RXEND.indication primitive prior

[11] This restriction is intended to avoid collisions due to inconsistent CCA reports in different QSTAs, not to optimize the bandwidth usage efficiency.

to a PHY-CCA.indication(idle) primitive, then the HC shall assume that the transmitted QoS (+)CF-Poll frame was successfully received by the polled non-AP QSTA. In the cases of QoS Data+CF-Poll, QoS Data+CF-Ack+CF-Poll, or QoS CF-Ack+CF-Poll, the PHY-CCA.indication(busy) primitive is used only to determine whether the transfer of control of the channel has been successful. The PHY-CCA.indication(busy) primitive is not used for determining the success or failure of the transmission. If the CF-Poll is piggybacked onto a QoS data frame, the HC may have to retransmit that QoS data frame subsequently.

If an HC receives a frame from a QSTA with a duration/ID covering only the response frame, the HC shall assume that the QSTA is terminating its TXOP, and the HC may initiate other transmissions or allow the channel to go into the CP.

If a polled non-AP QSTA has no queued traffic to send or if the MPDUs available to send are too long to be transmitted within the specified TXOP limit, the non-AP QSTA shall send a QoS (+)Null frame. In the case of no queued traffic, this QoS (+)Null frame shall have a QoS Control field that reports a queue size of 0 for any TID with the duration/ID set to the time required for the transmission of one ACK frame, plus one SIFS interval. In the case of insufficient TXOP size, such as when the maximum MSDU size is not specified, this QoS (+)Null frame shall have a QoS Control field that contains the TID and TXOP duration or a nonzero queue size needed to send the MPDU that is ready for transmission. When a queue size is transmitted, the HC shall combine the queue size information with the rate of the received QoS (+)Null frame to determine the required size of the requested TXOP.

Within a polled TXOP, the unused portion of TXOPs shall be returned back to the HC. The recipient of the final frame, with the Ack Policy subfield set to Normal Ack, shall be the HC if there will be time remaining in the TXOP after the transmission of the final frame and its expected ACK response frame. If there are no frames to be sent to the HC, then the non-AP QSTA shall send to the HC a QoS Null with the Queue Size subfield in the QoS Control field set to 0. If there is not enough time within the unused portion of the TXOP to transmit either the QoS Null frame or the frame with the Duration/ID field covering only the response frame, then the non-AP QSTA shall cease control of the channel.[12] If the beginning of the reception of an expected ACK response frame to the final frame does not occur, detected as the nonoccurrence of PHY-CCA.indication(busy) primitive at the non-AP QSTA that is expecting the response during the first slot time following SIFS, the non-AP QSTA shall retransmit the frame or transmit a QoS Null frame, with the Ack Policy subfield set to Normal Ack and the Queue Size subfield set to 0, after PIFS from the end of last transmission, until such time that it receives an acknowledgment or when there is not enough time remaining in the TXOP for sending such a frame. This is to avoid the situation where the HC may not receive the frame and may result in an inefficient use of the channel. If a PHY-CCA.indication(busy) primitive occurs at the non-AP QSTA that is expecting the ACK response frame during the first slot following SIFS after the end of the transmission of the final frame, it shall be interpreted as indicating that the channel control has been successfully transferred and no further frames shall be transmitted by the non-AP QSTA in the TXOP, even though the ACK frame from HC may be incorrectly received. Note that while PHY-CCA.indication(busy) primitive is used in this instance to determine the control of the channel, it is not used for determining the success or failure of the transmission.

9.9.2.2 TXOP structure and timing

Any QoS data frame of a subtype that includes CF-Poll contains a TXOP limit in its QoS Control field. The ensuing polled TXOP is protected by the NAV set by the Duration field of the frame that contained the QoS (+)CF-Poll function, as shown in Figure 62d. Within a polled TXOP, a QSTA may initiate the transmission of one or more frame exchange sequences, with all such sequences nominally separated by a SIFS interval. The QSTA shall not initiate transmission of a frame unless the transmission and any acknowledgment or other immediate response expected from the peer MAC entity are able to complete prior to the end of the remaining TXOP duration. All transmissions, including the response frames, within the polled TXOP are

[12]In this case the channel will not be accessed until the NAVs expire at all the STAs.

considered to be the part of the TXOP, and the HC shall account for these when setting the TXOP limit. If the TXOP Limit subfield in the QoS Control field of the QoS data frame that includes CF-Poll is set to 0, then the QSTA to which the frame is directed to shall respond with either one MPDU or one QoS Null frame.

A TXOP or transmission within a TXOP shall not extend across TBTT, dot11CFPMaxDuration (if during CFP), dot11MaxDwellTime (if using an FH PHY), or dot11CAPLimit. The HC shall ensure that the full duration of any granted TXOP meets these requirements so that non-AP QSTAs may use the time prior to the TXOP limit of a polled TXOP without checking for these constraints. Subject to these limitations, all decisions regarding what MSDUs and/or MMPDUs are transmitted during any given TXOP are made by the QSTA that holds the TXOP.[13, 14]

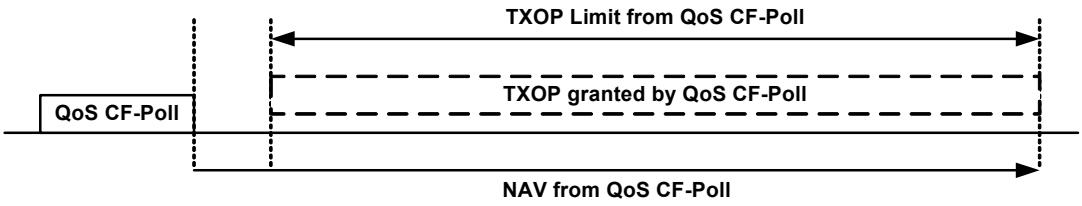


Figure 62d—Polled TXOP

9.9.2.2.1 NAV operation during a TXOP

A HC shall set its own NAV to prevent its transmitting during a TXOP that it has granted to a QSTA through an HCCA poll. However, the HC may reclaim the TXOP if a QSTA is not using it or ends the TXOP early (see 9.9.2.1.3).

In a CFP or CP, if the HC has no more QSTAs to poll and it has no more data, management, BlockAckReq, or BlockAck frames to send, it may reset the NAVs of all QSTAs in the QBSS by sending a QoS CF-Poll frame with the RA matching its own MAC address and with the Duration/ID field set to 0. When the QAP contains a PC, during the CFP, it may reset the NAVs of all STAs within the QBSS (i.e., corresponding to the same BSSID) by sending a CF-End frame, regardless of how the NAVs have been originally set.

A non-AP QSTA that receives a CF-End frame containing the BSSID of the QBSS, in which the non-AP QSTA is associated, shall reset the NAV value to 0. A non-AP QSTA that receives a QoS (+)CF-Poll frame with a MAC address in the Address 1 field that matches the HC's MAC address and the Duration/ID field value equal to zero shall reset the NAV value to 0.

When a QSTA receives a QoS (+)CF-Poll frame containing the BSSID of the QBSS in which the QSTA is associated, that QSTA shall update the NAV if necessary and shall save the TXOP holder address for that QBSS, which is the MAC address from the Address 1 field of the frame containing the QoS (+)CF-Poll. If an RTS frame is received with the RA address matching the MAC address of the QSTA and the MAC

[13]In certain regulatory domains, channel sensing must be done at periodic intervals (for example, in Japan, this period is 4 ms). This means that the duration of a TXOP in these regulatory domains might not be more than this periodic interval. If longer durations are desired, then the TXOP holder needs to sense the channel at least once in the limit imposed in the regulatory domain, by waiting for at least for the duration of one PIFS during which it senses the channel. If it does not detect any energy, it may continue by sending the next frame. In other words, the total TXOP size assigned should include an extra time allocated (i.e., $n \times$ aSlotTime, where n is the number of times the QSTA needs to sense the channel and is given by floor(TXOP limit/limit imposed in the regulatory domain).

[14]The TID value in the QoS Control field of a QoS Data+CF-Poll frame pertains only to the MSDU or fragment thereof in the Frame Body field of that frame. This TID value does not pertain to the TXOP limit value and does not place any constraints on what frame(s) the addressed QSTA may send in the granted TXOP.

address in the TA field in the RTS frame matches the saved TXOP holder address, then the QSTA shall send the CTS frame after SIFS, without regard for, and without resetting, its NAV. The saved TXOP holder address shall be cleared when the NAV is reset or when the NAV counts down to 0.

When a QSTA receives a frame addressed to it and requires an acknowledgment, it shall respond with an ACK or QoS +CF-Ack frame independent of its NAV. A non-AP QSTA shall accept a polled TXOP by initiating a frame exchange sequence independent of its NAV.

9.9.2.3 HCCA transfer rules

A TXOP obtained by receiving a QoS (+)CF-Poll frame uses the specified TXOP limit consisting of one or more frame exchange sequences with the sole time-related restriction being that the final sequence shall end not later than the TXOP limit. In QoS CF-Poll and QoS CF-Ack+CF-Poll frames, the TID subfield in the QoS Control field indicates the TS for which the poll is intended. The requirement to respond to that TID is nonbinding, and a QSTA may respond with any frame. Upon receiving a QoS (+)CF-Poll frame, a non-AP QSTA may send any frames, i.e., QoS data frames belonging to any TID as well as management frames in the obtained TXOP. MSDUs may be fragmented in order to fit within TXOPs.

The QoS CF-Poll frames shall be sent only by an HC. Non-AP QSTAs are not allowed to send QoS (+)CF-Poll frames. Non-AP QSTAs shall not send QoS (+)Data frames in response to any data frame other than the QoS (+)CF-Poll frames.

If a QSTA has set up at least one TS for which the Aggregation subfield in the associated TSPEC is set to 0, the QAP shall use only QoS CF-Poll or QoS CF-Ack+CF-Poll frames to poll the QSTA and shall never use QoS (+)Data+CF-Poll to poll the QSTA. It should be noted that although QoS (+)CF-Poll is a data frame, but it should be transmitted at one of the rates in the BSS basic rate set in order to set the NAV of all QSTA that are not being polled (see 9.6). If a CF-Poll is piggybacked with a QoS data frame, then the MSDU may be transmitted at the rate that is below the negotiated minimum PHY rate.

QSTAs shall use QoS data frames for all MSDU transfers to another QSTA. The TID in the QoS Control fields of these frames shall indicate the TC or TS to which the MPDU belongs. Furthermore, either the Queue Size subfield shall indicate the amount of queued traffic present in the output queue that the QSTA uses for traffic belonging to this TC or TS, or the TXOP Duration Requested subfield shall indicate the duration that the QSTA desires for the next TXOP for traffic belonging to this TC or TS. The queue size value reflects the amount on the appropriate queue not including the present MPDU. A non-AP QSTA that wishes to inform the HC of queue status may use the QoS Null frame indicating the TID and the queue size or TXOP duration request (also see 9.9.2.3.1).

QSTAs shall be able to receive QoS +CF-Ack frames. The HC may use QoS Data+CF-Ack frames to send frames to the same non-AP QSTA a SIFS after receiving the final transmission of the previous TXOP. The HC may also use QoS Data+CF-Ack frames to send frames to any other non-AP QSTA a SIFS after receiving the final transmission of the previous TXOP, if the non-AP QSTA that sent the final transmission of the previous TXOP has set the Q-Ack subfield in the QoS Capability information element in the (Re)Association Request frame to 1. In both CFP and CP, QSTAs shall always respond to QoS data frames having the Ack Policy subfield in the QoS Control field set to Normal Ack with an ACK frame, unless the acknowledgment is piggybacked in which case it shall use a QoS +CF-Ack frame. Piggybacked frames are allowed only in CFP or within TXOPs initiated by the HC. The HC shall not send a QoS data frame containing a +CF-Ack with an Address 1 that does not correspond to the address of the QSTA for which the +CF-Ack is intended, unless the QSTA to which the +CF-Ack is intended, sets the Q-Ack subfield in the QoS Capability information element in the (Re)Association Request frame. QSTAs are not required to be able to transmit QoS data frames with subtypes that include +CF-Ack.

A QAP that has dot11QACKOptionImplemented true may allow associations with STAs advertising support for the Q-Ack option. Such associations do not require the QAP to employ piggyback

acknowledgments directed toward that associated STA in frames that are not directed to that associated STA. QSTAs shall be able to process received QoS data frames with subtypes that include +CF-Ack when the QSTA to which the acknowledgment is directed is the same as the QSTA addressed by the Address 1 field of that QoS data frame. A QSTA that does not set the Q-Ack subfield to 1 in the QoS Capability information element in the (Re)Association Request frame is not required to handle the received QoS (+)Data+CF-Ack frames that are addressed to other QSTAs. The net effect of these restrictions on the use of QoS +CF-Ack frames is that the principal QoS +CF-Ack subtype that is useful is the QoS Data+CF-Ack frame, which can be sent by a non-AP QSTA as the first frame in a polled TXOP when that TXOP was conveyed in a QoS Data+CF-Poll(+CF-Ack) frame and the outgoing frame is directed to the HC's QSTA address. QoS (Data+)CF-Poll+CF-Ack frames are useful if the HC wants to grant another TXOP to the same non-AP QSTA a SIFS after receiving the final transmission of that non-AP QSTA's previous TXOP. QoS (Data+)CF-Poll+CF-Ack frames are also useful if the HC wants to grant another TXOP to a different non-AP QSTA a SIFS after receiving the final transmission of a non-AP QSTA's previous TXOP, if the non-AP QSTA that sent the final transmission of the previous TXOP has set the Q-Ack subfield in the QoS Capability information element in the (Re)Association Request frame to 1.

HCF contention-based channel access shall not be used to transmit MSDUs belonging to an established TS (with the HC's acceptance of the associated TSPEC), unless the granted TSPEC indicates it is permitted to do so when the Access Policy subfield of the TS Info field is set to "HCCA, EDCA mixed mode" (HEMM), the polled QSTA utilized the full TXOP provided by the HC, and it has to send more MPDUs. When this QSTA sends frames belonging to a TS using contention-based channel access, it shall encode the TID subfield in the QoS data frame with the TID associated with the TS. When the QAP grants a TSPEC with the Access Policy subfield set to HEMM and if the corresponding AC needs admission control, the QAP shall include the medium time that specifies the granted time for EDCA access in the ADDTS Response frame.

9.9.2.3.1 TXOP requests

Non-AP QSTAs may send TXOP requests during polled TXOPs or EDCA TXOPs using the QoS Control field in the QoS data or QoS Null frame directed to the HC, with the TXOP Duration Requested or Queue Size subfield value and TID subfield value indicated to the HC. QAPs indicate whether they process TXOP request or queue size in the QoS Info field in the Beacon, Probe Response, and (Re)Association Response frames. QAPs shall process requests in at least one format. The QAP may reallocate TXOPs if the request belongs to TS or update the EDCA parameter set if the above request belongs to TC. Non-AP QSTAs shall use only the request format that the QAP indicates it can process.

Even if the value of TXOP Duration Requested subfield or Queue Size subfield in a QoS data frame is zero, the HC shall continue to poll according to the negotiated schedule.

9.9.2.3.2 Use of RTS/CTS

QSTAs may send an RTS frame as the first frame of any frame exchange sequence for which improved NAV protection is desired, during either the CP or CFP, and without regard for dot11RTSThreshold.[15]

If a QSTA sends an RTS frame and does not receive an expected CTS frame, then the recovery rules are as specified in 9.9.2.1.3.

If NAV protection is desired for a transmission to the QAP in response to a QoS data frame with a subtype that includes CF-Poll, the polled non-AP QSTA is allowed to send a CTS frame (as a CTS frame is shorter

[15]The sending of an RTS frame during the CFP is usually unnecessary, but may be used to ensure that the addressed recipient QSTA is within range and awake and to elicit a CTS response that sets the NAV at STAs in the vicinity of the addressed recipient. This is useful when there are nearby STAs that are members of other BSSs and are out of range to receive Beacon frames from this BSS. Sending an RTS frame during the CFP is useful only when the recipient is a QSTA, because an nQSTA in the same BSS has its NAV set to protect the CFP, which renders those nQSTAs unable to respond. Using the same duration calculation during the CFP as specified for the CP is directly applicable for all cases except when the RTS frame is sent by the HC and the following frame includes a QoS (+)CF-Poll.

than a QoS data frame and can be sent with a higher probability that it will be received by other STAs) with the RA containing its own MAC address in order to set the NAV in its own vicinity without the extra time to send an RTS frame.[16]

9.9.3 Admission Control at the HC

An IEEE 802.11 network may use admission control to administer policy or regulate the available bandwidth resources. Admission control is also required when a QSTA desires guarantee on the amount of time that it can access the channel. The HC, which is in the QAP, is used to administer admission control in the network. As the QoS facility supports two access mechanisms, there are two distinct admission control mechanisms: one for contention-based access and another for controlled access.

Admission control, in general, depends on vendors' implementation of the scheduler, available channel capacity, link conditions, retransmission limits, and the scheduling requirements of a given stream. All of these criteria affect the admissibility of a given stream. If the HC has admitted no streams that require polling, it may not find it necessary to perform the scheduler or related HC functions.

9.9.3.1 Contention-based admission control procedures

A non-AP QSTA may support admission control procedures in 9.9.3.1.2 to send frames in the AC where admission control is mandated; but, if it does not support that procedure, it shall use EDCA parameters of a lower priority AC, as indicated in Table 20i, that does not require admission control. QAPs shall support admission control procedures, at least to the minimal extent of advertising that admission is not mandatory on its ACs.

The QAP uses the ACM (admission control mandatory) subfields advertised in the EDCA Parameter Set element to indicate whether admission control is required for each of the ACs. While the CWmin, CWmax, AIFS, TXOP limit parameters may be adjusted over time by the QAP, the ACM bit shall be static for the duration of the lifetime of the BSS. An ADDTS Request frame shall be transmitted by a non-AP QSTA to the HC in order to request admission of traffic in any direction (i.e., uplink, downlink, direct, or bidirectional) employing an AC that requires admission control. The ADDTS Request frame shall contain the UP associated with the traffic and shall indicate EDCA as the access policy. The QAP shall associate the received UP of the ADDTS Request frame with the appropriate AC as per the UP-to-AC mappings described in 9.1.3.1. The non-AP QSTA may transmit unadmitted traffic for the ACs for which the QAP does not require admission control. If a QSTA desires to send data without admission control using an AC that mandates admission control, the QSTA shall use EDCA parameters that correspond to a lower priority and do not require admission control. All ACs with priority higher than that of an AC with an ACM flag equal to 1 should have the ACM flag set to 1.

9.9.3.1.1 Procedures at the QAP

Regardless of the AC's ACM setting, the QAP shall respond to an ADDTS Request frame with an ADDTS Response frame that may be to accept or deny the request. On receipt of an ADDTS Request frame from a non-AP QSTA, the QAP shall make a determination about whether to

a) Accept the request, or
b) Deny the request.

The algorithm used by the QAP to make this determination is a local matter. If the QAP decides to accept the request, the QAP shall also derive the medium time from the information conveyed in the TSPEC element in the ADDTS Request frame. The QAP may use any algorithm in deriving the medium time, but K.2.2 provides a procedure that may be used. Having made such a determination, the QAP shall transmit a

[16]This is unnecessary because the NAVs in the vicinity of the QAP were set by the QoS (+)CF-Poll frame.

TSPEC element to the requesting non-AP QSTA contained in an ADDTS Response frame. If the QAP is accepting the request, the Medium Time field shall be specified.

9.9.3.1.2 Procedure at non-AP QSTAs

Each EDCAF shall maintain two variables: admitted_time and used_time.

The admitted_time and used_time shall be set to 0 at the time of (re)association. The non-AP QSTA may subsequently decide to explicitly request medium time for the AC that is associated with the specified priority.

In order to make such a request, the non-AP QSTA shall transmit a TSPEC element contained in an ADDTS Request frame with the following fields specified (i.e., nonzero): Nominal MSDU Size, Mean Data Rate, Minimum PHY Rate, Inactivity Interval, and Surplus Bandwidth Allowance. The Medium Time field is not used in the request frame and shall be set to 0.

On receipt of a TSPEC element contained in a ADDTS Response frame indicating that the request has been accepted, the non-AP QSTA shall recompute the admitted_time for the specified EDCAF as follows:

admitted_time = admitted_time + dot11EDCAAveragingPeriod * (medium time of TSPEC).

The non-AP QSTA may choose to tear down the explicit request at any time. For the teardown of an explicit admission, the non-AP QSTA shall transmit a DELTS frame containing the TSID and direction that specify the TSPEC to the QAP.

If the non-AP QSTA sends or receives a DELTS frame, it shall recompute the admitted_time for the specified EDCAF as follows:

admitted_time = admitted_time – dot11EDCAAveragingPeriod * (medium time of TSPEC).

To describe the behavior at the non-AP QSTA, two parameters are defined. The parameter used_time signifies the amount of time used, in units of 32 μs, by the non-AP QSTA in dot11EDCAAveragingPeriod. The parameter admitted_time is the medium time allowed by the QAP, in units of 32 μs, in dot11EDCA-AveragingPeriod. The non-AP QSTA shall update the value of used_time:

a) At dot11EDCAAveragingPeriod second intervals

used_time = max((used_time – admitted_time), 0)

b) After each successful or unsuccessful MPDU (re)transmission attempt,

used_time = used_time + MPDUExchangeTime

The MPDUExchangeTime equals the time required to transmit the MPDU sequence. For the case of an MPDU transmitted with Normal Ack policy and without RTS/CTS protection, this equals the time required to transmit the MPDU plus the time required to transmit the expected response frame plus one SIFS. If the used_time value reaches or exceeds the admitted_time value, the corresponding EDCAF shall no longer transmit using the EDCA parameters for that AC as specified in the QoS Parameter Set element. However, a non-AP QSTA may choose to temporarily replace the EDCA parameters for that EDCAF with those specified for an AC of lower priority, if no admission control is required for those ACs.

If, for example, a non-AP QSTA has made and had accepted an explicit admission for a TS and the channel conditions subsequently worsen, possibly including a change in PHY data rate so that it requires more time to send the same data, the non-AP QSTA may make a request for more admitted_time to the QAP and at the

same time downgrade the EDCA parameters for that AC for short intervals in order to send some of the traffic at the admitted priority and some at the unadmitted priority, while waiting for a response to the admission request.

9.9.3.2 Controlled-access admission control

This subclause describes the schedule management of the admitted HCCA streams by the HC. When the HC provides controlled channel access to non-AP QSTAs, it is responsible for granting or denying polling service to a TS based on the parameters in the associated TSPEC. If the TS is admitted, the HC is responsible for scheduling channel access to this TS based on the negotiated TSPEC parameters. The HC should not initiate a modification of TSPEC parameters of an admitted TS unless requested by the STA. The HC should not tear down a TS unless explicitly requested by the STA or at the expiry of the inactivity timer. The polling service based on admitted TS provides a "guaranteed channel access" from the scheduler in order to have its QoS requirements met. This is an achievable goal when the WM operates free of external interference (such as operation within the channel by other technologies and co-channel overlapping BSS interference). The nature of wireless communications may preclude absolute guarantees to satisfy QoS requirements. However, in a controlled environment (e.g., no interference), the behavior of the scheduler can be observed and verified to be compliant to meet the service schedule.

The normative behavior of the scheduler is as follows:

— The scheduler shall be implemented so that, under controlled operating conditions, all STAs with admitted TS are offered TXOPs that satisfy the service schedule.

— Specifically, if a TS is admitted by the HC, then the scheduler shall service the non-AP QSTA during an SP. An SP is a contiguous time during which a set of one or more downlink unicast frames and/or one or more polled TXOPs are granted to the QSTA. An SP starts at fixed intervals of time specified in Service Interval field. The first SP starts when the lower order 4 bytes of the TSF timer equals the value specified in Service Start Time field.[17] Additionally, the minimum TXOP duration shall be at least the time to transmit one maximum MSDU size successfully at the minimum PHY rate specified in the TSPEC. If maximum MSDU size is not specified in the TSPEC, then the minimum TXOP duration shall be at least the time to transmit one nominal MSDU size successfully at the minimum PHY rate. The vendors are free to implement any optimized algorithms, such as reducing the polling overheads, increasing the TXOP duration, etc., within the parameters of the transmitted schedule.

When the HC aggregates the admitted TS, it shall set the Aggregation field in the granted TSPEC to 1. A QAP shall schedule the transmissions in HCCA TXOPs and communicate the service schedule to the non-AP QSTA. The HC shall provide an aggregate service schedule if the non-AP QSTA sets the Aggregation field in its TSPEC request. If the QAP establishes an aggregate service schedule for a non-AP QSTA, it shall aggregate all HCCA streams for the QSTA. The service schedule is communicated to the non-AP QSTA in a Schedule element contained in an ADDTS Response frame. In the ADDTS Response frame, the modified service start time shall not exceed the requested service start time, if specified in ADDTS Request frame, by more than one maximum service interval (SI). The HC uses the maximum SI for the initial scheduling only as there may be situations that HC may not be able to service the TS at the scheduled timing, due to an EDCA or DCF transmission or other interferences interrupting the schedule. The Service Interval field value in the Schedule element shall be greater than the minimum SI. The service schedule could be subsequently updated by a QAP as long as it meets TSPEC requirements.

The HC may update the service schedule at any time by sending a Schedule element in a Schedule frame. The updated schedule is in effect when the HC receives the ACK frame for the Schedule frame. The service start time in the Schedule element in the Schedule frame shall not exceed the beginning of the immediately

[17]The lower order 4 bytes of the TSF timer cover a span of around 71 min. Due to TSF timer wrapover and due to the possibility of receiving the schedule frame after the indicated start time timer, ambiguity may occur. This ambiguity is resolved by using the nearest absolute TSF timer value in past or future when the lower order 4 bytes match the Start Time field in the Schedule element.

previous SP by more than the maximum SI. The service start time shall not precede the beginning of the immediately previous SP by more than the minimum SI.

A non-AP QSTA may affect the service schedule by modifying or deleting its existing TS as specified in 11.4.

K.3.1 provides guidelines for deriving an aggregate service schedule for a single non-AP QSTA from the non-AP QSTA's admitted TS. The schedule shall meet the QoS requirements specified in the TSPEC.

During any time interval [$t1$, $t2$] including the interval that is greater than the specification interval, the cumulative TXOP duration shall be greater than the time required to transmit all MSDUs (of nominal MSDU size) arriving at the mean data rate for the stream, over the period [$t1$, $t2 - D$]. The parameter D is set to the specified maximum SI in the TSPEC. If maximum SI is not specified, then D is set to the delay bound in the TSPEC.

The HC shall use the minimum PHY rate in calculating TXOPs if the minimum PHY rate is present in the TSPEC field in the ADDTS response. Otherwise, the HC may use an observed PHY rate in calculating TXOPs. Non-AP QSTAs may have an operational rate lower than the minimum PHY rate due to varying conditions on the channel for a short time and still may be able to sustain the TS without changing the minimum PHY rate in the TSPEC.

A minimum set of TSPEC parameters shall be specified during the TSPEC negotiation. The specification of a minimum set of parameters is required so that the scheduler can determine a schedule for the stream that is to be admitted. These parameters are Mean Data Rate, Nominal MSDU Size, Minimum PHY Rate, Surplus Bandwidth Allowance, and at least one of Maximum Service Interval and Delay Bound in the ADDTS Request frame. In the ADDTS Response frame, these parameters are Mean Data Rate, Nominal MSDU Size, Minimum PHY Rate, Surplus Bandwidth Allowance, and Maximum Service Interval and shall be nonzero when a stream is admitted.

If any of the elements in the minimum set of parameters does not have the required nonzero value, as specified above in this subclause, in the ADDTS Request frame, the HC may replace the unspecified parameters with nonzero values and admit the stream, or it may reject the stream. If the HC admits the stream with the alternative set of TSPEC parameters, these parameters are indicated to the non-AP QSTA through the ADDTS Response frame. If both maximum SI and delay bound are specified, the HC may use only the maximum SI. If any other parameter is specified in the TSPEC element, the scheduler may use it when calculating the schedule for the stream. The HC may also use the UP value in the TS Info field for admission control or scheduling purposes; however, this decision is outside the scope of this amendment. The mandatory set of parameters can be set by any higher layer entity or may be generated autonomously by the MAC.

If a QSTA specifies a nonzero minimum SI and if the TS is admitted, the HC shall generate a schedule that conforms to the specified minimum SI.

A reference design for a sample scheduler and admission control unit is provided in Annex K. A sample use of the TSPEC for admission control is also described in Annex K.

Insert after 9.9.3.2 the following new subclauses (9.10 through 9.10.5), including the new figures in those subclauses:

9.10 Block Acknowledgment (Block Ack)

9.10.1 Introduction

The Block Ack mechanism improves channel efficiency by aggregating several acknowledgments into one frame. There are two types of Block Ack mechanisms: immediate and delayed. Immediate Block Ack is suitable for high-bandwidth, low-latency traffic while the delayed Block Ack is suitable for applications that tolerate moderate latency.[18] In this clause, the QSTA with data to send using the Block Ack mechanism is referred to as the *originator*, and the receiver of that data as the *recipient*.

The Block Ack mechanism is initialized by an exchange of ADDBA Request/Response frames. After initialization, blocks of QoS data frames can be transmitted from the originator to the recipient. A block may be started within a polled TXOP or by winning EDCA contention. The number of frames in the block is limited, and the amount of state that is to be kept by the recipient is bounded. The MPDUs within the block of frames are acknowledged by a BlockAck control frame, which is requested by a BlockAckReq control frame.

The Block Ack mechanism does not require the setting up of a TS; however, QSTAs using the TS facility may choose to signal their intention to use Block Ack mechanism for the scheduler's consideration in assigning TXOPs. Acknowledgments of frames belonging to the same TID, but transmitted during multiple TXOPs, may also be combined into a single BlockAck frame. This mechanism allows the originator to have flexibility regarding the transmission of data MPDUs. The originator may split the block of frames across TXOPs, separate the data transfer and the Block Ack exchange, and interleave blocks of MSDUs for different TIDs or RAs.

Figure 62e illustrates the message sequence chart for the setup, data and Block Ack transfer, and the teardown of the Block Ack mechanism, which are discussed in detail in 9.10.2 through 9.10.5.

9.10.2 Setup and modification of the Block Ack parameters

A QSTA that intends to use the Block Ack mechanism for the transmission of QoS data frames to a peer should first check whether the intended peer QSTA is capable of participating in Block Ack mechanism by discovering and examining its Delayed Block Ack and Immediate Block Ack capability bits. If the intended peer QSTA is capable of participating, the originator sends an ADDBA Request frame indicating the TID for which the Block Ack is being set up. The Block Ack Policy and Buffer Size fields in the ADDBA Request frame are advisory and may be changed by the recipient. The receiving QSTA shall respond by an ADDBA Response frame. The receiving QSTA, which is the intended peer, has the option of accepting or rejecting the request. When the QSTA accepts, then a Block Ack agreement exists between the originator and recipient. When the QSTA accepts, it indicates the type of Block Ack and the number of buffers that it shall allocate for the support of this block. If the receiving QSTA rejects the request, then the originator shall not use the Block Ack mechanism.

If the Block Ack mechanism is being set up for a TS, bandwidth negotiation (using ADDTS Request and Response frames) should precede the setup of the Block Ack mechanism.

Once the Block Ack exchange has been set up, data and ACK frames are transferred using the procedure described in 9.10.3.

[18] The delayed Block Ack mechanism is primarily intended to allow existing implementations to use this feature with minimal hardware changes and also to allow inexpensive implementations that would use the processing power on the host.

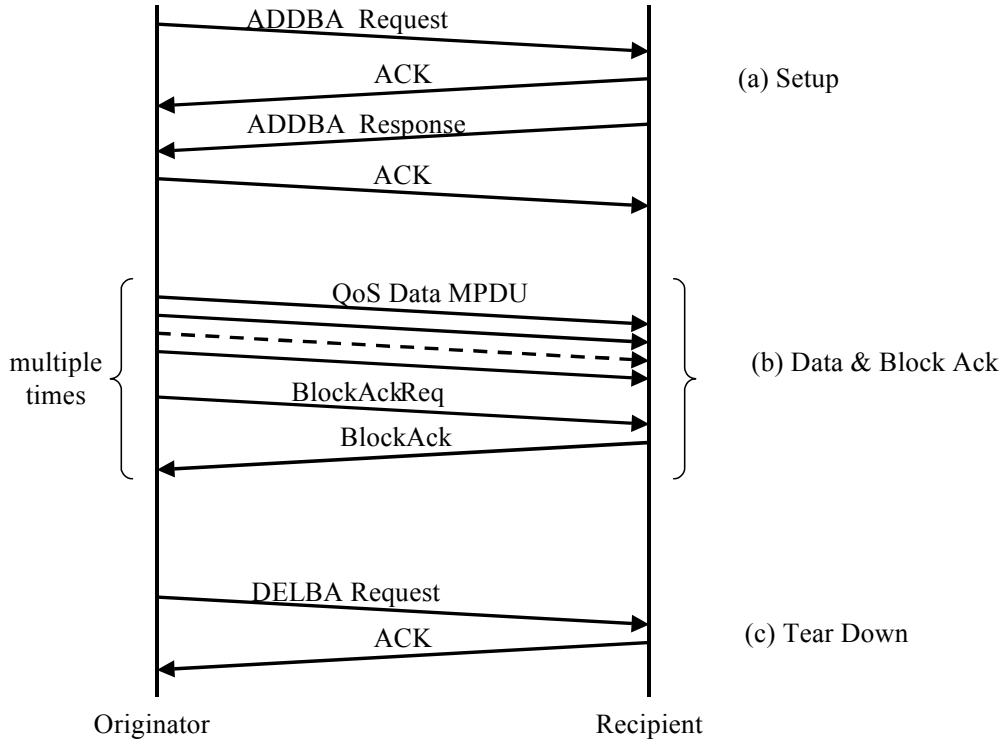


Figure 62e—Message sequence chart for Block Ack mechanism: (a) setup, (b) data and acknowledgment transfer and (c) tear down

9.10.3 Data and acknowledgment transfer

After setting up for the Block exchange following the procedure in 9.10.2, the originator may transmit a block of QoS data frames separated by SIFS period, with the total number of frames not exceeding the Buffer Size subfield value in the associated ADDBA Response frame. Each of the frames shall have the Ack Policy subfield in the QoS Control field set to Block Ack. The RA field of the frames shall be the recipient's *unicast* address. The originator requests acknowledgment of outstanding QoS data frames by sending a BlockAckReq frame. The recipient shall maintain a Block Ack record for the block.

Subject to any constraints in this subclause about permitted use of TXOP according to the channel access mechanism used, the originator may

— Separate the Block and BlockAckReq frames into separate TXOPs
— Split a Block frame across multiple TXOPs
— Sequence frames with different TIDs in the same TXOP
— Interleave MPDUs with different TIDs within the same TXOP
— Sequence or interleave MPDUs for different RAs within a TXOP

A protective mechanism (such as transmitting using HCCA, RTS/CTS, or the mechanism described in 9.13) should be used to reduce the probability of other STAs transmitting during the TXOP. If no protective mechanism is used, then the first frame that is sent as a block shall have a response frame and shall have the Duration field set so that the NAVs are set to appropriate values at all STAs in the QBSS.

The originator shall use the Block Ack starting sequence control to signal the first MPDU in the block for which an acknowledgment is expected. MPDUs in the recipient's buffer with a sequence control value that

precedes the starting sequence control value are called *preceding MPDUs*. The recipient shall reassemble any complete MSDUs from buffered preceding MPDUs and indicate these to its higher layer. The recipient shall then release any buffers held by preceding MPDUs. The range of the outstanding MPDUs (i.e., the reorder buffer) shall begin on an MSDU boundary. The total number of frames that can be sent depends on the total number of MPDUs in all the outstanding MSDUs. The total number of MPDUs in these MSDUs may not exceed the reorder buffer size in the receiver.

The recipient shall maintain a Block Ack record consisting of originator address, TID, and a record of reordering buffer size indexed by the received MPDU sequence control value. This record holds the acknowledgment state of the data frames received from the originator.

If the immediate Block Ack policy is used, the recipient shall respond to a BlockAckReq frame with a BlockAck frame. If the recipient sends the BlockAck frame, the originator updates its own record and retries any frames that are not acknowledged in the BlockAck frame, either in another block or individually.

If the delayed Block Ack policy is used, the recipient shall respond to a BlockAckReq frame with an ACK frame. The recipient shall then send its Block Ack response in a subsequently obtained TXOP. Once the contents of the BlockAck frame have been prepared, the recipient shall send this frame in the earliest possible TXOP using the highest priority AC. The originator shall respond with an ACK frame upon receipt of the BlockAck frame. If delayed Block Ack policy is used and if the HC is the recipient, then the HC may respond with a +CF-Ack frame if the BlockAckReq frame is the final frame of the polled TXOP's frame exchange. If delayed Block Ack policy is used and if the HC is the originator, then the HC may respond with a +CF-Ack frame if the BlockAck frame is the final frame of the TXOP's frame exchange.

The BlockAck frame contains acknowledgments for the MPDUs of up to 64 previous MSDUs. In the BlockAck frame, the QSTA acknowledges only the MPDUs starting from the starting sequence control until the MPDU with the highest sequence number (modulo 2^{12}) that has been received, and the QSTA shall set bits in the Block Ack bitmap corresponding to all other MPDUs to 0. The status of MPDUs that are considered "old" and prior to the sequence number range for which the receiver maintains status shall be reported as successfully received (i.e., the corresponding bit in the bitmap shall be set to 1). The sequence number space is considered divided into two parts, one of which is "old" and one of which is "new" by means of a boundary created by adding half the sequence number range to the current start of receive window (modulo 2^{12}). If the BlockAck frame indicates that an MPDU was not received correctly, the originator shall retry that MPDU subject to that MPDU's appropriate lifetime limit.

A typical BlockAck frame exchange sequence using the immediate Block Ack for a single TID is shown in Figure 62f.

A typical Block Ack sequence using the delayed Block Ack is shown in Figure 62g.

The subsequent Block Ack request starting sequence number shall be higher than or equal to the starting sequence number (modulo 2^{12}) of the immediately preceding BlockAckReq frame for the same TID.

The originator may continue to transmit MPDUs to the recipient after transmitting the BlockAckReq frame, but before receiving the BlockAck frame (applicable only to delayed Block Ack). The bitmap in the BlockAck frame shall include the status of frames received between the start sequence number and the transmission of the BlockAckReq frame. A recipient sending a delayed BlockAck frame may update the bitmap with information on QoS data frames received between the receipt of the BlockAckReq frame and the transmission of the BlockAck frame.

If there is no response (i.e., neither a BlockAck nor an ACK frame) to the BlockAckReq frame, the originator may retransmit the BlockAckReq frame within the current TXOP (if time permits) or within a subsequent TXOP. MSDUs that are sent using the Block Ack mechanism are not subject to retry limits but only to MSDU lifetime. The originator need not set the retry bit for any possible retransmissions of the MPDUs.

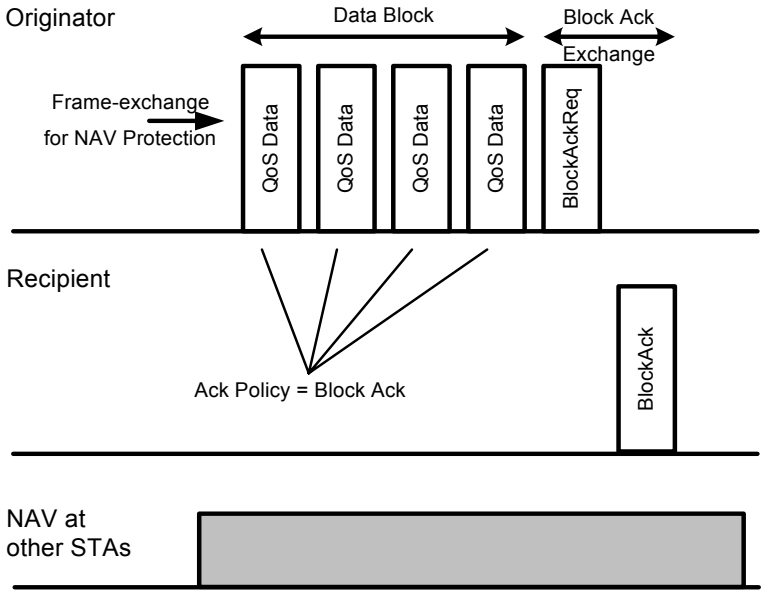


Figure 62f—A typical Block Ack sequence when immediate policy is used

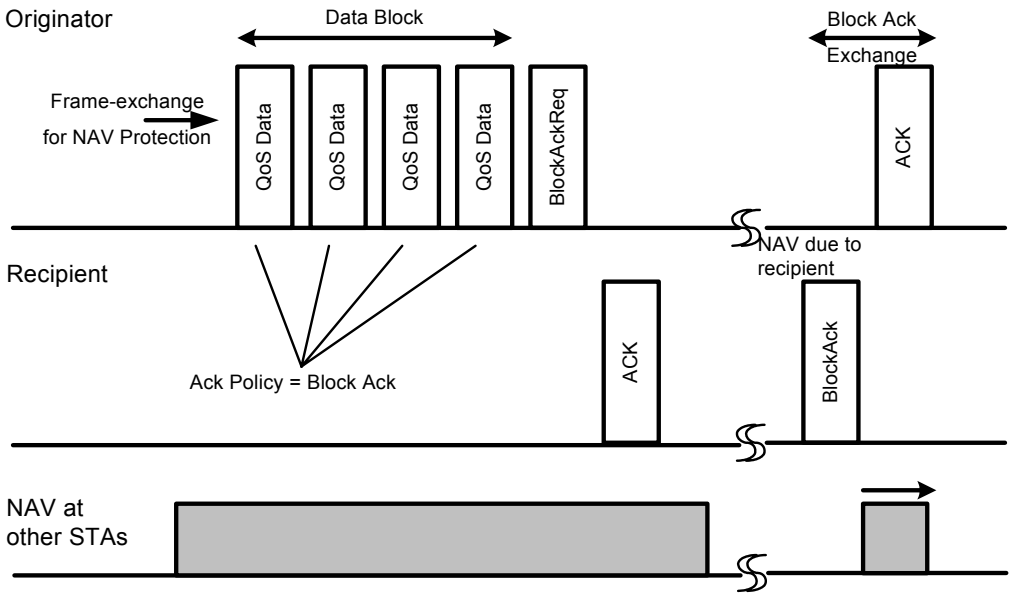


Figure 62g—A typical BlockAck sequence when delayed policy is used

The BlockAckReq frame shall be discarded if all MSDUs referenced by this BlockAckReq frame have been discarded from the transmit buffer due to expiry of their lifetime limit.

In order to improve efficiency, originators using the Block Ack facility may send MPDU frames with the Ack Policy subfield in QoS control frames set to Normal Ack if only a few MPDUs are available for transmission. The Block Ack record shall be updated irrespective of the Ack Policy subfield in the QoS data frame for the TID with an active Block Ack. When there are sufficient number of MPDUs, the originator may switch back to the use of Block Ack. The reception of QoS data frames using Normal Ack policy shall not be used by the recipient to reset the timer to detect Block Ack timeout (see 11.5.3). This allows the recipient to delete the Block Ack if the originator does not switch back to using Block Ack.

The frame exchange sequences are provided in 9.12.

9.10.4 Receive buffer operation

Upon the receipt of a QoS data frame from the originator for which the Block Ack agreement exists, the recipient shall buffer the MSDU regardless of the value of the Ack Policy subfield within the QoS Control field of the QoS data frame.

The recipient flushes received MSDUs from its receive buffer as described in this subclause.

If a BlockAckReq frame is received, all complete MSDUs with lower sequence numbers than the starting sequence number contained in the BlockAckReq frame shall be indicated to the MAC client using the MA-UNIDATA.indication primitive. Upon arrival of a BlockAckReq frame, the recipient shall indicate the MSDUs starting with the starting sequence number sequentially until there is an incomplete MSDU in the buffer.

If, after an MPDU is received, the receive buffer is full, the complete MSDU with the earliest sequence number shall be indicated to the MAC client using the MA-UNIDATA.indication primitive.

All comparisons of sequence numbers are performed circularly modulo 2^{12}.

The recipient shall always indicate the reception of MSDU to its MAC client in order of increasing sequence number.

9.10.5 Teardown of the Block Ack mechanism

When the originator has no data to send and the final Block Ack exchange has completed, it shall signal the end of its use of the Block Ack mechanism by sending the DELBA frame to its recipient. There is no management response frame from the recipient.[19] The recipient of the DELBA frame shall release all resources allocated for the Block Ack transfer.

The Block Ack agreement may be torn down if there are no BlockAck, BlockAckReq, or QoS data frames (sent under Block Ack policy) for the Block Ack's TID received from the peer within a duration of Block Ack timeout value (see 11.5.3).

[19]Normal Ack rules apply.

Insert after 9.10.5 the following new subclause (9.11):

9.11 No Acknowledgment (No Ack)

The usage of No Ack is determined by the policy at the QSTA. When No Ack policy is used, there is no MAC-level recovery, and the reliability of this traffic is reduced, relative to the reliability of traffic with other acknowledgment policies, due to the increased probability of lost frames from interference, collisions, or time-varying channel parameters. A protective mechanism (such as transmitting using HCCA, RTS/CTS, or the mechanism described in 9.13) should be used to reduce the probability of other STAs transmitting during the TXOP.

Change the following subclause number due to the change above in subclause order (see page 78) and change the first paragraph of the subclause as shown:

9.12 ~~9.7~~ Frame exchange sequences

The allowable frame exchange sequences are summarized in Table 21 ~~and Table 22~~through Table 22g. A legend applicable to ~~both~~all the tables follows ~~Table 22~~Table 22g. The frame exchange sequences are also illustrated in Figure 62h through Figure 62m in the form of high-order message sequence charts (HMSCs).

Insert the following entries at the end of Table 21 in 9.12:

Table 21—Frame sequences)

Sequence	Frames in sequence	Usage
<HCF-Sequence>	1 or more	Start of frame sequences initiated by a QSTA

Insert the following entries at the end of Table 22 in 9.12:

Table 22—CF frame sequences

CF frame sequence	Frames in sequence	Usage
<HCF-Sequence>	1 or more	Start of frame sequences initiated by a QSTA

Insert the following new tables (Table 22d through Table 22g) after Table 22 and move the legend to after Table 22g:

(NOTE: With the change in subclause order, Table 22a through Table 22c now come before Table 21.)

Table 22d—<HCF sequence>

HCF frame sequence	Frames in sequence	Usage
{CTS-}[Data (bc/mc) =] Data(bc/mc)	1 or more	Broadcast or multicast MSDU sent by a QAP
{CTS-} [QoS Data (bc/mc) =] QoS Data (bc/mc)	1 or more	Broadcast or multicast MSDU sent by a QAP as QoS data frames
{CTS-} [Mgmt (bc) =] Mgmt(bc)	1 or more	Broadcast MMPDU sent by a QAP
{CTS-} Mgmt(bc)	1 or 2	Broadcast MMPDU sent by a non-AP QSTA.
{CTS-}<TXOP-Sequence> [- <TXOP-Sequence>]	1 or more	Start of a TXOP sequence using contention
QoS CF-Poll self	1	QoS CF-Poll frame sent by the HC with the DA same as the HC's MAC address.
<CF-Ack-Piggybacked-QoS-Poll-Sequence>	2 or more	Poll started by a QoS CF-Poll frame onto which an acknowledgment is piggybacked.
{RTS - CTS - } <Non-CF-Ack-Piggybacked-QoS-Poll-Sequence>	2 or more	Poll started by a QoS CF-Poll frame onto which an acknowledgment is not piggybacked.

Table 22e—<Poll sequence>

Poll sequence[a]	Frames in sequence	Usage
QoS CF-Poll{+CF-Ack }(no data) { - CTS self } - <TXOP-Sequence> [- <TXOP-Sequence>] { - QoS Null(dir, normal ack)(no data) - ACK}	2 or more	Start of a TXOP sequence as initiated with a separate poll from HC. (*1), (*3), (*4)
QoS CF-Poll{+CF-Ack}(no data) - QoS Null(dir, normal ack)(no data) - ACK	3	Separate QoS CF-Poll frame from an HC to a QSTA with empty queue, insufficient time for queued MPDU, or too little time remaining before a dwell or medium occupancy boundary to send a queued frame (*1)
QoS Data(dir,normal ack)+CF-Poll{+CF-Ack} - <CF-Ack-Piggybacked-QoS-Data-Sequence>	3 or more	Start of a TXOP sequence as initiated with a poll from HC that is piggybacked onto a data frame. The acknowledgment policy of the data frame is set to Normal Ack. The response frame is an acknowledgment piggybacked onto the data frame with acknowledgment policy set to Normal Ack. (*1), (*2), (*3)
QoS Data(dir,normal ack)+CF-Poll{+CF-Ack} - ACK - <TXOP-Sequence> [- <TXOP-Sequence>] { - QoS Null(dir, normal ack)(no data) - ACK}	2 or more	Start of a TXOP sequence as initiated with a poll from HC that is piggybacked onto a data frame. The acknowledgment policy of the data frame is set to Normal Ack. The response frame is an ACK frame. (*1), (*3)

Table 22e—<Poll sequence> *(continued)*

Poll sequence[a]	Frames in sequence	Usage
QoS Data(dir,normal ack)+CF-Poll{+CF-Ack} - ACK - QoS Null(dir, normal ack)(no data) - ACK	4	QoS Poll piggybacked onto a QoS data frame with acknowledgment policy set to Normal Ack, from the HC to a QSTA with empty queue, insufficient time for queued MPDU, or too little time remaining before a dwell or medium occupancy boundary to send a queued frame (*1)
QoS Data(dir,block ack | no ack)+CF-Poll{+CF-Ack} {- CTS self } - <TXOP-Sequence> [- <TXOP-Sequence>] { -QoS Null(dir, normal ack)(no data) - ACK}	2 or more	Start of a TXOP sequence as initiated with a poll from the HC with CF-Ack with acknowledgment policy set to Block Ack or No Ack. (*1), (*3), (*4)
QoS Data(dir,block ack | no ack)+CF-Poll{+CF-Ack} - QoS Null(dir, normal ack)(no data) - ACK	3	QoS Poll from the HC to a QSTA with empty queue, insufficient time for queued MPDU, or too little time remaining before a dwell or medium occupancy boundary to send a queued frame (*1)

[a]If Piggybacked CF-Ack is present, it is a CF-Ack-Piggybacked-QoS-Poll-Sequence. Otherwise, it is a Non-CF-Ack-Piggybacked-QoS-Poll-Sequence.

Table 22f—<TXOP sequence>

TXOP frame sequence	Frames in sequence	Usage
{RTS - CTS - } Mgmt(dir) - ACK	2	Directed MMPDU
{RTS - CTS - } Data(dir) - ACK	2	Directed MPDU, transmitted by the QAP only
{RTS - CTS - } QoS Data(dir, normal ack) - ACK	2	Directed MPDU with Normal Ack policy responded by an ACK
{RTS - CTS - } QoS Data(dir, normal ack) - <CF-Ack-Piggybacked-QoS-Data-Sequence>	2 or more	Directed MPDU with Normal Ack policy responded with a new TXOP
{RTS - CTS - } QoS Data(dir, normal ack) - <CF-Ack-Piggybacked-QoS-Poll-Sequence>	2 or more	Directed MPDU with Normal Ack policy responded with a new TXOP
{RTS - CTS - } QoS Data(dir, normal ack) - Data+CF-Ack	2	Directed MPDU with Normal Ack policy responded with a Data+CF-Ack
RTS - CTS - QoS Data(dir, no ack)	3	Directed MPDUs with No Ack policy when they are sent as the first in a burst of frames within EDCA.
{RTS - CTS - } QoS Data(dir, no ack)	3	Directed MPDUs with No Ack policy when they are sent within a polled TXOP.
RTS - CTS - QoS Data(dir, block ack)	3	Directed MPDUs with Block Ack policy when they are sent as the first in a burst of frames within EDCA.
{RTS - CTS - } QoS Data (dir, block ack)	3	Directed MPDUs with Block Ack policy when they are sent within a polled TXOP.

Table 22f—<TXOP sequence> *(continued)*

TXOP frame sequence	Frames in sequence	Usage
{RTS - CTS - } QoS Null(dir, normal ack) (no data) - ACK	2	QoS Null frame to indicate no data or queue size
{RTS - CTS - } QoS Null(dir, normal ack) (no data) - <CF-Ack-Piggybacked-QoS-Data-Sequence>	2 or more	QoS Null with Normal Ack policy responded with a new TXOP
{RTS - CTS - } QoS Null(dir, normal ack) (no data) - <CF-Ack-Piggybacked-QoS-Poll-Sequence>	2 or more	QoS Null with Normal Ack policy responded with a new TXOP
{RTS - CTS - } QoS Null(dir, normal ack) (no data) - Data+CF-Ack	2	QoS Null with Normal Ack policy responded with a Data+CF-Ack
{RTS - CTS - } BAR - BA	2	Immediate Block Ack Request and its response frame.
{RTS - CTS - } BAR - ACK	2	Delayed Block Ack Request and a response frame to it.
{RTS - CTS - } BAR - <CF-Ack-Piggybacked-QoS-Data-Sequence>	2 or more	Delayed Block Ack Request responded with a new TXOP
{RTS - CTS - } BAR - <CF-Ack-Piggybacked-QoS-Poll-Sequence>	2 or more	Delayed Block Ack Request responded with a new TXOP
{RTS - CTS - } BAR - Data+CF-Ack	2	Delayed Block Ack Request responded with a Data+CF-Ack
{RTS - CTS - } BA - ACK	2	Delayed Block Ack response
{RTS - CTS - } BA - <CF-Ack-Piggybacked-QoS-Data-Sequence>	2 or more	Delayed Block Ack responded with a new TXOP
{RTS - CTS - } BA - <CF-Ack-Piggybacked-QoS-Poll-Sequence>	2 or more	Delayed Block Ack responded with a new TXOP
{RTS - CTS - } BA - Data+CF-Ack	2	Delayed Block Ack responded with a Data+CF-Ack

Table 22g—<CF-Ack piggybacked QoS data sequence>

CF-Ack piggybacked QoS data sequence	Frames in sequence	Usage
QoS Data(dir, normal ack)+CF-Ack - ACK [- TXOP-Sequence] {QoS Null(dir, normal ack)(no data) - ACK}	2 or more	Start of a TXOP sequence with a QoS data frame piggybacked onto the acknowledgment to the data frame sent in the previous sequence. (*1), (*2), (*3)
QoS Data(dir,normal ack)+CF-Ack - <CF-Ack-QoS-Piggybacked Poll Sequence>	3 or more	Start of a TXOP sequence with a QoS data frame piggybacked onto the acknowledgment to the data frame sent in the previous sequence. The response frame would start a new piggybacked Poll Sequence. (*1), (*2), (*3) (*5)

Table 22g—<CF-Ack piggybacked QoS data sequence> *(continued)*

CF-Ack piggybacked QoS data sequence	Frames in sequence	Usage
QoS Data(dir,normal ack)+CF-Ack - <CF-Ack-Piggybacked-QoS-Data-Sequence>	3 or more	Start of a TXOP sequence with a QoS data frame piggybacked onto the acknowledgment to the data frame sent in the previous sequence. The response frame would start a piggybacked non-polled data frame Sequence. (*1), (*2), (*3) (*5)
QoS Data(dir,normal ack)+CF-Ack - Data+CF-Ack	2	Start of a TXOP sequence with a QoS data frame with acknowledgment policy set to Normal Ack and piggybacked onto the acknowledgment to the data frame sent in the previous sequence. The response frame is a Data+CF-Ack, addressed to an nQSTA. (*1), (*2), (*3) (*5)
QoS Data(dir,block ack | no ack)+CF-Ack [- <TXOP-Sequence>] { - QoS Null(dir, normal ack)(no data) - ACK}	1 or more	Start of a TXOP sequence with a QoS data frame with acknowledgment policy set to Block Ack or No Ack and piggybacked onto the acknowledgment to the data frame sent in the previous sequence. (*1), (*2), (*3)

Change the title of legend after Table 22g as follows:

LEGEND (For Table 21 ~~and Table 22~~through Table 22g)

Insert the following definitions at the end of the legend after Table 22g:

24—A vertical bar (\|) between two items implies that one or the other item, but not both, is used.

25—"QoS Data(dir,normal ack)" represents any QoS data frame that has no piggybacking of acknowledgment or polling, with an individual address in the Address1 field and the acknowledgment policy set to normal ack.

26—"QoS Data(dir,no ack)" represents any QoS data frame that has no piggybacking of acknowledgment or polling, with an individual address in the Address1 field and the acknowledgment policy set to no ack.

27—"QoS Data(dir,block ack)" represents any QoS data frame that has no piggybacking of acknowledgment or polling, with an individual address in the Address1 field and the acknowledgment policy set to block ack.

28—"BAR" represents any BlockAckReq frame.

29—"BA" represents any BlockAck frame.

30—"QoS CF-Poll(no data)" represents a data frame of subtype QoS CF-Poll(no data).

31—(*1) implies that the frame is transmitted within a SIFS of the previously transmitted MPDU.

32—(*2) implies that piggybacking of acknowledgment to data frame is allowed only in the first response frame.

33—(*3) implies that the final QoS Null sequence shall be inserted if the TXOP holder has insufficient time for queued MPDU, or too little time remaining before a dwell or medium occupancy boundary to send a queued frame, and if the remaining TXOP is big enough to send the QoS Null sequence, or the TXOP holder has no queued MPDU.

34—(*4) implies CTS to self is allowed if the non-AP QSTA intends to set the NAV at its vicinity. The frame that follows CTS is sent by the QSTA that transmits the CTS.

35—(*5) Piggybacked poll sequence or a new sequence is started only when the previously received frame has the Duration field set to cover only the response frame.

36—An isolated equal sign (=) represents a PIFS separating the pair of frames.

Insert the following new figures (Figure 62h through Figure 62m) after the legend after Table 22g:

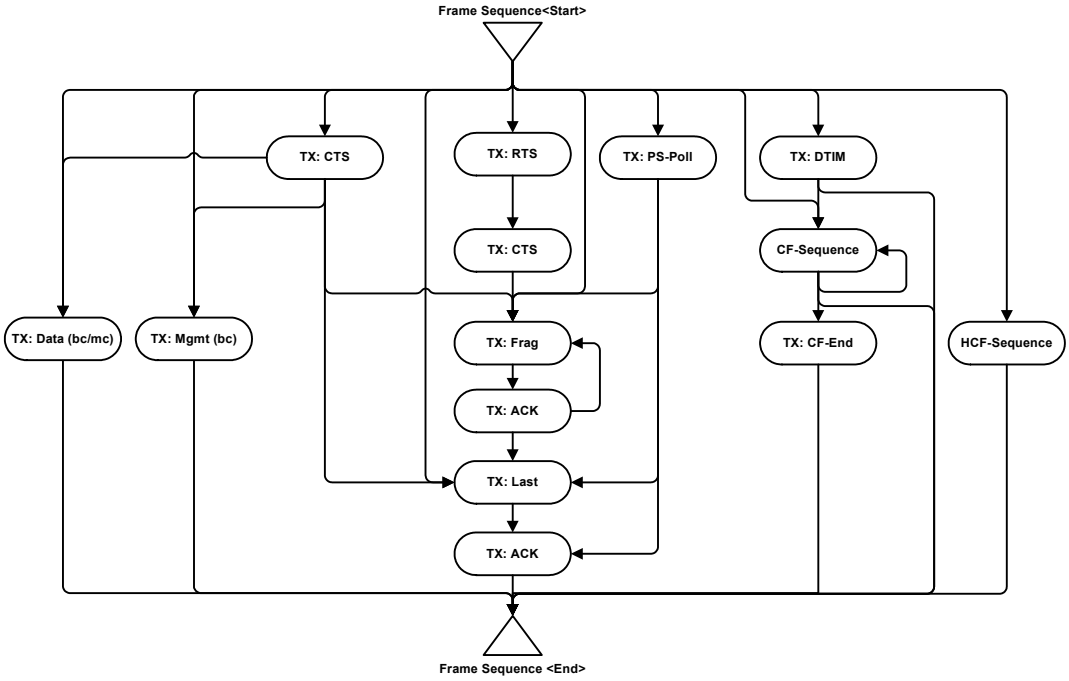


Figure 62h—Frame sequence

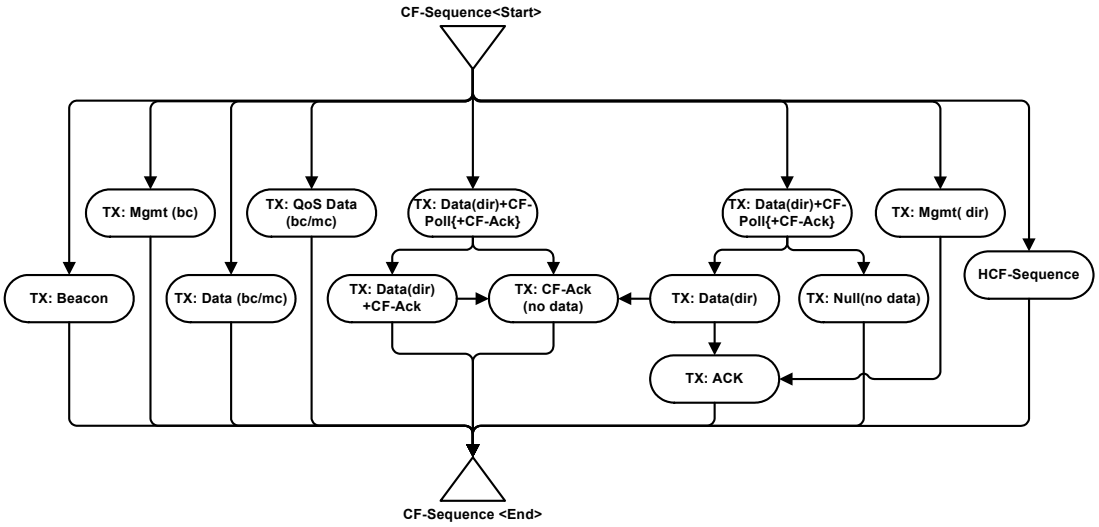


Figure 62i—CF frame sequence

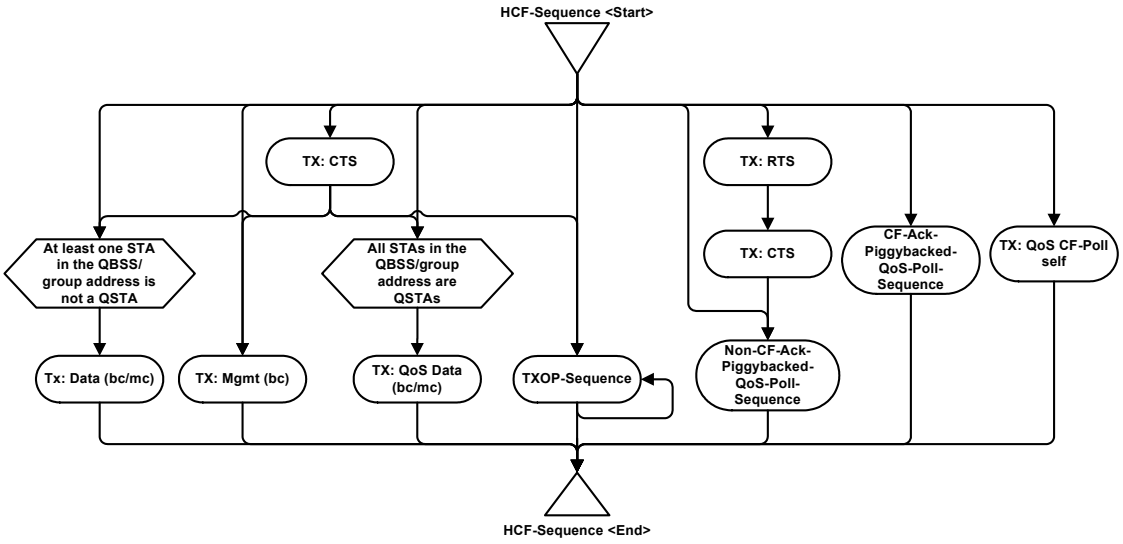


Figure 62j—HCF sequence

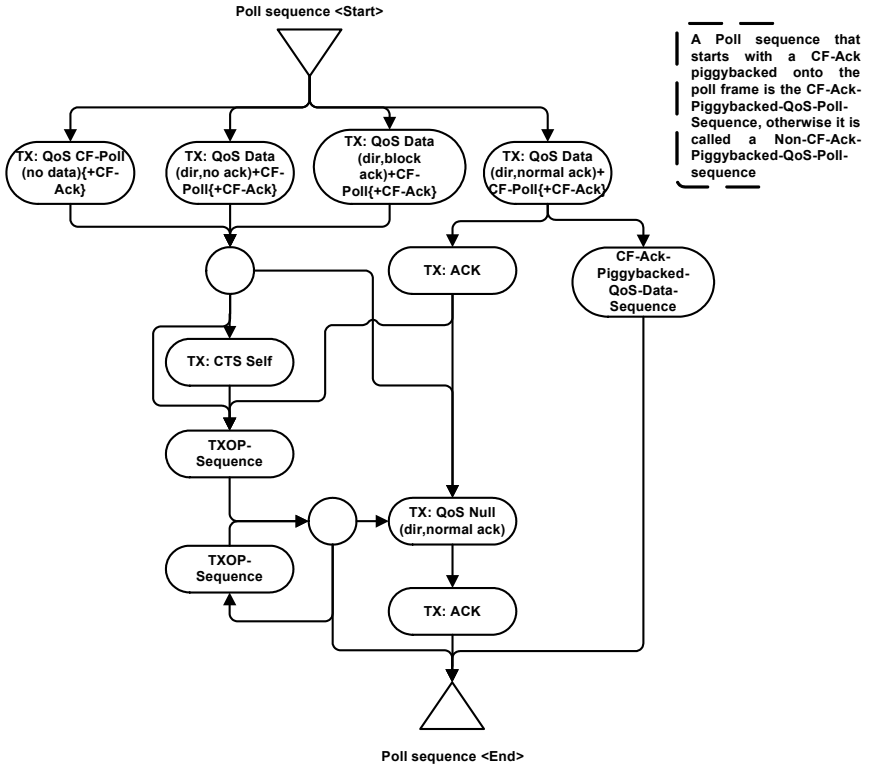


Figure 62k—Poll sequence

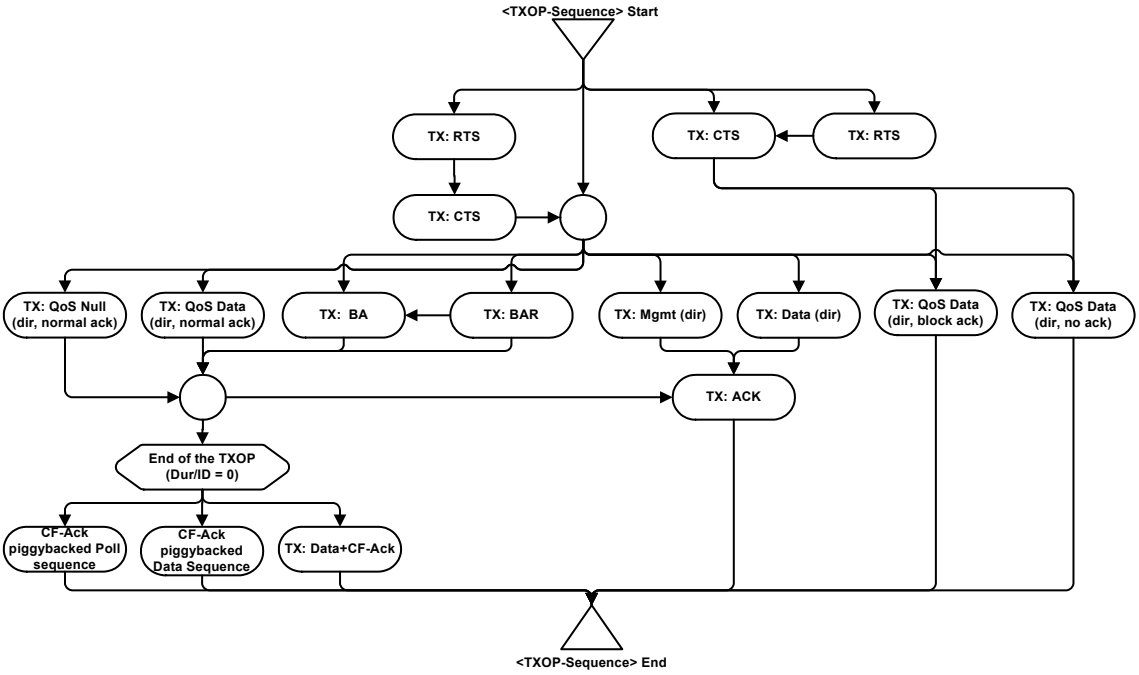


Figure 62l—TXOP sequence

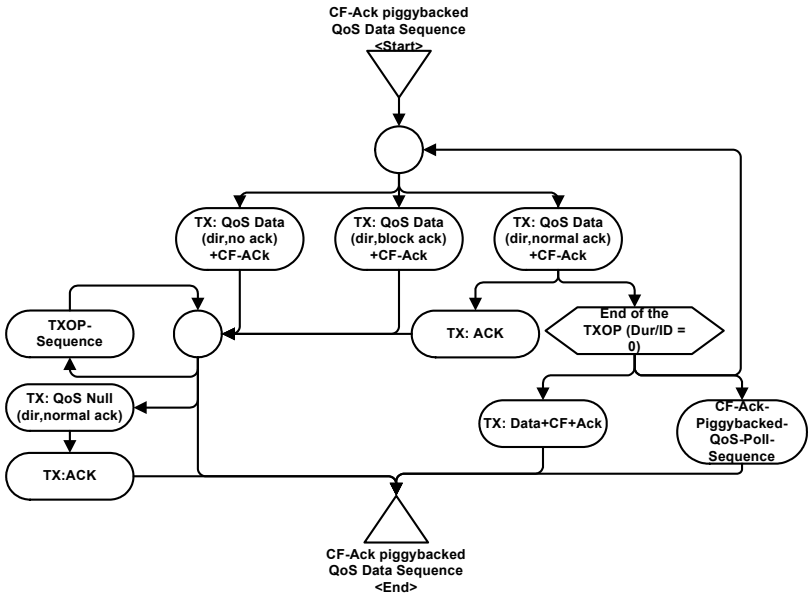


Figure 62m—CF-Ack piggybacked QoS data sequence

Change the following subclause number due to the changes above in subclause order (9.7 through 9.12):

9.13 ~~9.10~~ Protection mechanism

End of changes to Clause 9.

10. Layer management

10.3 MLME SAP interface

10.3.2 Scan

10.3.2.2 MLME-SCAN.confirm

10.3.2.2.2 Semantics of the service primitive

Insert the following entries at the end of the untitled table describing the BSSDescription parameter in 10.3.2.2.2:

Name	Type	Valid range	Description
QBSSLoad	As defined in frame format	As defined in frame format	The values from the QBSS Load information element if such an element was present in the probe response or beacon, else null.
EDCAParameterSet	As defined in frame format	As defined in frame format	The values from the EDCA Parameter Set information element if such an element was present in the probe response or beacon, else null.
QoSCapability	As defined in frame format	As defined in frame format	The values from the QoS Capability information element if such an element was present in the probe response or beacon, else null.

10.3.6 Associate

10.3.6.1 MLME-ASSOCIATE.request

10.3.6.1.2 Semantics of the service primitive

Change the primitive parameters in 10.3.6.1.2 as shown:

MLME-ASSOCIATE.request(
PeerSTAAddress,
AssociateFailureTimeout,
CapabilityInformation,
ListenInterval,
Supported Channels,
RSN,
QoSCapability
)

Insert the following entry at the end of the untitled table defining the primitive parameters in 10.3.6.1.2:

Name	Type	Valid range	Description
QoSCapability	As defined in frame format	As defined in frame format	Specifies the parameters within the QoS Capability information element that are supported by the MAC entity. The parameter shall be present only if the MIB attribute dot11QosOptionImplemented is true.

10.3.6.2 MLME-ASSOCIATE.confirm

10.3.6.2.2 Semantics of the service primitive

Change the primitive parameters in 10.3.6.2.2 as shown:

MLME-ASSOCIATE.confirm(
Resultcode,
CapabilityInformation,
SupportedRates,
EDCAParameterSet
)

Insert the following entries at the end of the untitled table defining the primitive parameters in 10.3.6.2.2:

Name	Type	Valid range	Description
CapabilityInformation	As defined in frame format	As defined in frame format	Specifies the operational capability definitions to be used by the peer MAC entity (AP).
SupportedRates	Set of integers	1–127 inclusive (for each integer in the set)	The set of data rates that are supported by the peer MAC entity (AP).
EDCAParameterSet	As defined in frame format	As defined in frame format	Specifies the EDCA parameter set that the QSTA should use. The parameter shall be present only if the MIB attribute dot11QosOptionImplemented is true.

10.3.6.3 MLME-ASSOCIATE.indication

10.3.6.3.2 Semantics of the service primitive

Change the primitive parameters in 10.3.6.3.2 as shown:

MLME-ASSOCIATE.indication(
PeerSTAAddress,
RSN,
CapabilityInformation,
SupportedRates,
QoSCapability
)

Insert the following entries at the end of the untitled table defining the primitive parameters in 10.3.6.3.2:

Name	Type	Valid range	Description
CapabilityInformation	As defined in frame format	As defined in frame format	Specifies the operational capability definitions to be used by the peer MAC entity.
SupportedRates	Set of integers	1–127 inclusive (for each integer in the set)	The set of data rates that are supported by the peer MAC entity.
QoSCapability	As defined in frame format	As defined in frame format	Specifies the parameters within the QoS Capability that are supported by the peer MAC entity. The parameter may be present only if the MIB attribute dot11QosOptionImplemented is true.

10.3.7 Reassociate

10.3.7.1 MLME-REASSOCIATE.request

10.3.7.1.2 Semantics of the service primitive

Change the primitive parameters in 10.3.7.1.2 as shown:

MLME-REASSOCIATE.request(
NewAPAddress,
ReassociateFailureTimeout,
CapabilityInformation,
ListenInterval,
Supported Channels,
RSN,
QoSCapability
)

Insert the following entry at the end of the untitled table defining the primitive parameters in 10.3.7.1.2:

Name	Type	Valid range	Description
QoSCapability	As defined in frame format	As defined in frame format	Specifies the parameters within the QoS Capability information element that are supported by the MAC entity. The parameter shall be present only if the MIB attribute dot11QosOptionImplemented is true.

10.3.7.2 MLME-REASSOCIATIE.confirm

10.3.7.2.2 Semantics of the service primitive

Change the primitive parameters in 10.3.7.2.2 as shown:

MLME-REASSOCIATE.confirm(
Resultcode,
CapabilityInformation,
SupportedRates,
EDCAParameterSet
)

Insert the following entries at the end of the untitled table defining the primitive parameters in 10.3.7.2.2:

Name	Type	Valid range	Description
CapabilityInformation	As defined in frame format	As defined in frame format	Specifies the operational capability definitions to be used by the peer MAC entity (AP).
SupportedRates	Set of integers	1–127 inclusive (for each integer in the set)	The set of data rates that are supported by the peer MAC entity (AP).
EDCAParameterSet	As defined in frame format	As defined in frame format	Specifies the EDCA parameter set that the QSTA should use. The parameter shall be present only if the MIB attribute dot11QosOptionImplemented is true.

10.3.7.3 MLME-REASSOCIATE.indication

10.3.7.3.2 Semantics of the service primitive

Change the primitive parameters in 10.3.7.3.2 as shown:

MLME-REASSOCIATE.indication(
PeerSTAAddress,
RSN,
CapabilityInformation,
SupportedRates,
QoSCapability
)

Insert the following entries at the end of the untitled table defining the primitive parameters in 10.3.7.3.2:

Name	Type	Valid range	Description
CapabilityInformation	As defined in frame format	As defined in frame format	Specifies the operational capability definitions to be used by the peer MAC entity.
SupportedRates	Set of integers	1–127 inclusive (for each integer in the set)	The set of data rates that are supported by the peer MAC entity.
QoSCapability	As defined in frame format	As defined in frame format	Specifies the parameters within the QoS Capability that are supported by the peer MAC entity. The parameter shall be present only if the MIB attribute dot11QosOptionImplemented is true.

10.3.10 Start

10.3.10.1 MLME-START.request

10.3.10.1.2 Semantics of the service primitive

Change the primitive parameters in 10.3.10.1.2 as shown:

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
)

Insert the following entry at the end of the untitled table defining the primitive parameters in 10.3.10.1.2:

Name	Type	Valid range	Description
EDCAParameterSet (QoS only)	As defined in frame format	As defined in frame format	The initial EDCA parameter set values to be used in the QBSS. The parameter shall be present only if the MIB attribute dot11QosOptionImplemented is true.

Insert after 10.3.23.1.4 the following new subclauses (10.3.24 through 10.3.28.3.4), including the tables in those subclauses:

10.3.24 TS management interface

This mechanism supports the process of adding, modifying, or deleting a TS in a QBSS using the procedures defined in 11.4.

The primitives used for this mechanism are called *TS Management primitives*, which include MLME-ADDTS.xxx and MLME-DELTS.xxx primitives, where xxx denotes request, confirm, indication, or response. Each primitive contains parameters that correspond to a QoS Action frame. Requests and responses may cause the transmission of the corresponding QoS Action frames. Confirms and indications are issued upon the receipt of the appropriate QoS Action frame.

Table 22h defines which primitives are supported by which type of STA.

Table 22h—Supported TS Management primitives

Primitive	Request	Confirm	Indication	Response
ADDTS	non-AP QSTA	non-AP QSTA	HC	HC
DELTS	non-AP QSTA and HC	non-AP QSTA and HC	non-AP QSTA and HC	—

10.3.24.1 MLME-ADDTS.request

10.3.24.1.1 Function

This primitive requests addition (or modification) of a TS. It is valid at the non-AP QSTA and requests the HC to admit the new or changed TS.

10.3.24.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-ADDTS.request (
                        DialogToken,
                        TSPEC,
                        TCLAS,
                        TCLASProcessing,
                        ADDTSFailureTimeout
                        )
```

Name	Type	Valid range	Description
DialogToken	Integer	0–127	Specifies a number unique to the QoS Action primitives and frames used in adding (or modifying) the TS of concern.
TSPEC	As defined in frame format with the exception of the surplus bandwidth allowance, minimum PHY rate, and maximum and minimum SIs, which are optionally specified	As defined in frame format with the exception of the surplus bandwidth allowance, minimum PHY rate, and maximum and minimum SIs, which are optionally specified	Specifies the TSID, traffic characteristics, and QoS requirements of the TS of concern.
TCLAS (0 or more)	As defined in frame format	As defined in frame format	Specifies the rules and parameters by which an MSDU may be classified to the specified TS.
TCLASProcessing	As defined in frame format	As defined in frame format	Specifies how the TCLAS elements are to be processed when there are multiple TCLAS elements.
ADDTSFailure-Timeout	Integer	Greater than or equal to 1	Specifies a time limit (in TU) after which the TS setup procedure will be is terminated.

10.3.24.1.3 When generated

This primitive is generated by the SME at a non-AP QSTA to request the addition of a new (or modification of an existing) TS in order to support parameterized QoS transport of the MSDUs belonging to this TS when a higher layer protocol or mechanism signals the QSTA to initiate such an addition (or modification).

10.3.24.1.4 Effect of receipt

The non-AP QSTA operates according to the procedures defined in 11.4.

10.3.24.2 MLME-ADDTS.confirm

10.3.24.2.1 Function

This primitive reports the results of a TS addition (or modification) attempt.

10.3.24.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-ADDTS.confirm(
                        ResultCode,
                        DialogToken,
                        TSDelay,
                        TSPEC,
                        Schedule,
                        TCLAS,
                        TCLASProcessing
                        )
```

Name	Type	Valid range	Description
ResultCode	Enumeration	SUCCESS, INVALID_ PARAMETERS, REJECTED_ WITH_ SUGGESTED_ CHANGES, REJECTED_ FOR_DELAY_ PERIOD, TIMEOUT, TRANSMISSION _FAILURE	Indicates the results of the corresponding MLME-ADDTS.request primitive.
DialogToken	Integer	As defined in the corresponding MLME-ADDTS.request primitive	Specifies a number unique to the QoS Action primitives and frames used in adding (or modifying) the TS.
TSDelay	Integer	≥ 0	When the result code is REJECTED_FOR_DELAY_ PERIOD, provides the amount of time a QSTA should wait before attempting another ADDTS request frame.
TSPEC	As defined in frame format	As defined in frame format	Specifies the TS information, traffic characteristics, and QoS requirements of the TS.
Schedule	As defined in frame format	As defined in frame format	Specifies the schedule information, service start time, SI, and the specification interval.
TCLAS (0 or more)	As defined in frame format	As defined in frame format	Specifies the rules and parameters by which an MSDU may be classified to the specified TS.
TCLASProcessing	As defined in frame format	As defined in frame format	Specifies how the TCLAS elements are to be processed when there are multiple TCLAS elements.

For the ResultCode value of SUCCESS, the TSPEC and the optional TCLAS parameters describe the characteristics of the TS that has been created (or modified); and the specified (nonzero) parameters [with the exception of Service Start Time, Medium Time, and any possibly unspecified minimum set of parameters (see 9.9.3.2) in the TSPEC in ADDTS Request frame] exactly match those of the matching MLME-ADDTS.request primitive.

For other values of ResultCode, no new TS has been created. In the case of REJECTED_WITH_ SUGGESTED_CHANGES, the TSPEC represents an alternative proposal by the HC. A TS is not created with this definition. If the suggested changes are acceptable to the non-AP QSTA, it is the responsibility of the non-AP QSTA to set up the TS with the suggested changes.

If this is the result of a modification of an existing TS, the status of that TS remains unchanged.

10.3.24.2.3 When generated

This primitive is generated by the MLME as a result of an MLME-ADDTS.request primitive indicating the results of that request.

This primitive is generated when that MLME-ADDTS.request primitive is found to contain invalid parameters, when a timeout occurs, or when the non-AP QSTA receives a response in the form of an ADDTS Response frame in the corresponding QoS Action frame from the HC.

10.3.24.2.4 Effect of receipt

The SME is notified of the results of the TS addition (or modification) procedure.

The SME should operate according to the procedures defined in 11.4.

10.3.24.3 MLME-ADDTS.indication

10.3.24.3.1 Function

This primitive reports to the HC SME the request for adding (or modifying) a TS.

10.3.24.3.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-ADDTS.indication (
DialogToken,
Non-APQSTAAddress
TSPEC,
TCLAS,
TCLASProcessing
)

Name	Type	Valid range	Description
DialogToken	Integer	As defined in the received ADDTS request frame	Specifies a number unique to the QoS Action primitives and frames used in adding (or modifying) the TS.
Non-APQSTA-Address	MACAddress		Contains the MAC address of the non-AP QSTA that initiated the MLME-ADDTS.request primitive.
TSPEC	As defined in frame format with the exception of the surplus bandwidth allowance, minimum PHY rate, and maximum and minimum SIs, which are optionally specified	As defined in the received ADDTS Request frame with the exception of the surplus bandwidth allowance, minimum PHY rate, and maximum and minimum SIs, which are optionally specified	Specifies the TSID, traffic characteristics, and QoS requirements of the TS.
TCLAS (0 or more)	As defined in frame format	As defined in frame format	Specifies the rules and parameters by which an MSDU may be classified to the specified TS.
TCLASProcessing	As defined in frame format	As defined in frame format	Specifies how the TCLAS elements are to be processed when there are multiple TCLAS elements.

The TCLAS is optional at the discretion of the non-AP QSTA that originated the request. An HC shall be capable of receiving an ADDTS request frame that contains a TCLAS element and capable of generating an indication that contains this as a parameter.

10.3.24.3.3 When generated

This primitive is generated by the MLME as a result of receipt of a request to add (or modify) a TS by a specified non-AP QSTA in the form of an Add TS request in the corresponding QoS Action frame.

10.3.24.3.4 Effect of receipt

The SME is notified of the request for a TS addition (or modification) by a specified non-AP QSTA.

This primitive solicits an MLME-ADDTS.response primitive from the SME that reflects the results of admission control at the HC on the TS requested to be added (or modified).

The SME should operate according to the procedures defined in 11.4.

The SME generates an MLME-ADDTS.response primitive within a dot11ADDTSResponseTimeout.

10.3.24.4 MLME-ADDTS.response

10.3.24.4.1 Function

This primitive responds to the request for a TS addition (or modification) by a specified non-AP QSTA MAC entity.

10.3.24.4.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-ADDTS.response (
ResultCode,
DialogToken,
Non-APQSTAAddress,
TSDelay,
TSPEC,
Schedule,
TCLAS,
TCLASProcessing
)

Name	Type	Valid range	Description
ResultCode	Enumeration	SUCCESS, INVALID_ PARAMETERS, REJECTED_ WITH_ SUGGESTED_ CHANGES, REJECTED_ FOR_DELAY_ PERIOD	Indicates the results of the corresponding MLME-ADDTS.indication primitive.
DialogToken	Integer	As defined in the corresponding MLME-ADDTS. indication	DialogToken of the matching MLME-ADDTS.indication primitive.
Non-APQSTA-Address	MAC-Address		Contains the non-AP QSTA address of the matching MLME-ADDTS.indication primitive.

Name	Type	Valid range	Description
TSDelay	Integer	≥ 0	When the result code is REJECTED_FOR_DELAY_PERIOD, provides the amount of time a QSTA should wait before attempting another ADDTS request.
TSPEC	As defined in frame format	As defined in frame format	Specifies the QoS parameters of the TS.
Schedule	As defined in frame format	As defined in frame format	Specifies the schedule information, service start time, SI, and the specification interval.
TCLAS (0 or more)	As defined in frame format	As defined in frame format	Specifies the rules and parameters by which an MSDU may be classified to the specified TS.
TCLASProcessing	As defined in frame format	As defined in frame format	Specifies how the TCLAS elements are to be processed when there are multiple TCLAS elements.

The DialogToken and non-APQSTAAddress parameters contain the values from the matching MLME-ADDTS.indication primitive.

If the result code is SUCCESS, the TSPEC and (optional) TCLAS parameters contain the values from the matching MLME-ADDTS-indication.

If the result code is REJECTED_WITH_SUGGESTED_CHANGES, the TSPEC and TCLAS parameters represent an alternative proposed TS. The TS, however, is not created. The TSID and direction values within the TSPEC are as in the matching MLME-ADDTS.indication primitive. The difference may lie in the QoS (e.g., minimum data rate, mean data rate, and delay bound) values, as a result of admission control performed at the SME of the HC on the TS requested to be added (or modified) by the non-AP QSTA. If sufficient bandwidth is not available, the QoS values may be reduced. In one extreme, the minimum data rate, mean data rate, and delay bound may be all set to 0, indicating that no QoS is to be provided to this TS.

10.3.24.4.3 When generated

This primitive is generated by the SME at the HC as a result of an MLME-ADDTS.indication primitive to initiate addition (or modification) of a TS with a specified peer MAC entity or entities.

10.3.24.4.4 Effect of receipt

This primitive approves addition (or modification) of a TS requested by a specified non-AP QSTA MAC entity, with or without altering the TSPEC.

This primitive causes the MAC entity at the HC to send an ADDTS Response frame in the corresponding QoS Action management frame to the requesting non-AP QSTA containing the specified parameters.

10.3.24.5 MLME-DELTS.request

10.3.24.5.1 Function

This primitive requests deletion of a TS with a specified peer MAC.

This primitive may be generated at either a non-AP QSTA or the HC.

10.3.24.5.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-DELTS.request(
Non-APQSTAAddress,
TSInfo,
ReasonCode
)

Name	Type	Valid range	Description
Non-APQSTA-Address (HC only)	MACAddress		(At the HC only) Specifies the MAC address of the non-AP QSTA that initiated this TS.
TSInfo	As defined in frame format	As defined in frame format	Specifies the TS to be deleted.
ReasonCode	Enumeration	QSTA_ LEAVING, END_TS, UNKNOWN_ TS	Indicates the reason why the TS is being deleted.

10.3.24.5.3 When generated

This primitive is generated by the SME at a QSTA to initiate deletion of a TS when a higher layer protocol or mechanism signals the QSTA to initiate such a deletion.

10.3.24.5.4 Effect of receipt

This primitive initiates a TS deletion procedure. The MLME subsequently issues an MLME-DELTS.confirm primitive that reflects the results.

This primitive causes the local MAC entity to send out a DELTS frame containing the specified parameters. If this primitive was generated at the HC, the frame is sent to the specified non-AP QSTA MAC address. If this primitive was generated at the non-AP QSTA, the frame is sent to its HC. In either case, the DELTS frame does not solicit a response from the recipient frame other than an acknowledgment to receipt of the frame.

10.3.24.6 MLME-DELTS.confirm

10.3.24.6.1 Function

This primitive reports the transmission status of a TS deletion attempt with a specified peer MAC entity.

10.3.24.6.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-DELTS.confirm (
ResultCode,
Non-APQSTAAddress,
TSInfo
)

Name	Type	Valid range	Description
ResultCode	Enumeration	SUCCESS, INVALID_ PARAMETERS, FAILURE	Indicates the results of the corresponding MLME-DELTS.request primitive.
Non-APQSTA-Address (HC only)	MAC-Address		(HC only) Contains the non-AP QSTA MAC address of the matching request.
TSInfo	As defined in frame format	As defined in frame format	Contains the TS information of the matching request.

10.3.24.6.3 When generated

This primitive is generated by the MLME as a result of an MLME-DELTS.request primitive after the DELTS frame has been sent (or attempts to send it have failed) and any internal state regarding the use of the TS has been destroyed.

10.3.24.6.4 Effect of receipt

The SME is notified of the results of the TS deletion procedure.

10.3.24.7 MLME-DELTS.indication

10.3.24.7.1 Function

This primitive reports the deletion of a TS by a specified peer MAC entity or deletion of the TS due to an inactivity timeout (HC only).

10.3.24.7.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-DELTS.indication (
                        Non-APQSTAAddress,
                        TSInfo,
                        ReasonCode
                        )
```

Name	Type	Valid range	Description
Non-APQSTA-Address (HC only)	MAC address		(HC only) Specifies the MAC address of the non-AP QSTA for which the TS is being deleted.
TSInfo	As defined in frame format	As defined in the received DELTS frame	Specifies the TS information of the TS of concern.
ReasonCode	Enumeration	QSTA_LEAVING, END_TS, UNKNOWN_TS, TIMEOUT	Indicates the reason why the TS is being deleted.

10.3.24.7.3 When generated

This primitive is generated by the MLME as a result of receipt of an initiation to delete a TS by a specified peer MAC entity.

This primitive may also be generated by the MLME at the HC as a result of inactivity of a particular TS. Inactivity results when a period equal to the inactivity interval in the TSPEC for the TS elapses

— Without arrival of an MSDU belonging to that TS at the MAC entity of the HC via an MA-UNITDATA.request primitive when the HC is the source STA of that TS or

— Without reception of an MSDU belonging to that TS by the MAC entity of the HC when a non-AP QSTA is the source STA of that TS.

This primitive is generated after any other state concerning the TSID/direction within the MAC has been destroyed.

10.3.24.7.4 Effect of receipt

The SME is notified of the initiation of a TS deletion by a specified peer MAC entity.

10.3.25 Management of direct links

This clause describes the management procedures associated with direct links.

10.3.25.1 MLME-DLS.request

10.3.25.1.1 Function

This primitive requests the setup of a direct link with a specified peer MAC entity.

10.3.25.1.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-DLS.request (
PeerMACAddress,
DLSTimeoutValue,
DLSResponseTimeout
)

Name	Type	Valid range	Description
PeerMACAddress	MAC-Address	Any valid individual MAC address	Specifies the MAC address of the non-AP QSTA that is the intended immediate recipient of the data flow.
DLSTimeoutValue	Integer	≥ 0	Specifies a time limit (in seconds) after which the direct link is terminated if there are no frame exchange sequences with the peer. A value of 0 implies that the direct link is never to be terminated based on a timeout.
DLSResponse-Timeout	Integer	≥ 1	Specifies a time limit (in TU) after which the DLS procedure is terminated.

10.3.25.1.3 When generated

This primitive is generated by the SME at a non-AP QSTA to set up a direct link with another non-AP QSTA.

10.3.25.1.4 Effect of receipt

This primitive initiates a DLS procedure. The MLME subsequently issues an MLME-DLS.confirm primitive that reflects the results.

10.3.25.2 MLME-DLS.confirm

10.3.25.2.1 Function

This primitive reports the results of a DLS attempt with a specified peer MAC entity.

10.3.25.2.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-DLS.confirm (
PeerMACAddress,
ResultCode,
CapabilityInformation,
DLSTimeoutValue,
SupportedRates
)

Name	Type	Valid range	Description
PeerMACAddress	MAC-Address	Any valid individual MAC address	Specifies the MAC address of the non-AP QSTA that is the intended immediate recipient of the data flow.
ResultCode	Enumeration	SUCCESS, INVALID_PARAMETERS, NOT_ALLOWED, NOT_PRESENT, NOT_QSTA, REFUSED, TIMEOUT	Indicates the results of the corresponding MLME-DLS.request primitive.
Capability-Information	As defined in frame format	As defined in frame format	Specifies the operational capability definitions to be used by the peer MAC entity.
DLSTimeoutValue	Integer	≥ 0	Specifies a time limit (in seconds) after which the direct link is terminated if there are no frame exchange sequences with the peer. A value of 0 implies that the direct link is never to be terminated based on a timeout.
SupportedRates	Set of integers	1–127 inclusive (for each integer in the set)	The set of data rates that are supported by the peer MAC entity.

10.3.25.2.3 When generated

This primitive is generated by the MLME as a result of an MLME-DLS.request primitive to establish a direct link with a specified peer MAC entity.

10.3.25.2.4 Effect of receipt

The SME is notified of the results of the DLS procedure.

10.3.25.3 MLME-DLS.indication

10.3.25.3.1 Function

This primitive reports the establishment of a direct link with a specified peer MAC entity.

10.3.25.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-DLS.indication(
                        PeerMACAddress,
                        CapabilityInformation,
                        DLSTimeoutValue,
                        SupportedRates
                        )
```

Name	Type	Valid range	Description
PeerMACAddress	MAC-Address	Any valid individual MAC address	Specifies the MAC address of the non-AP QSTA that is the sender of the data flow.
Capability-Information	As defined in frame format	As defined in frame format	Specifies the operational capability definitions to be used by the peer MAC entity.
DLSTimeoutValue	Integer	≥ 0	Specifies a time limit (in seconds) after which the direct link is terminated if there are no frame exchange sequences with the peer. A value of 0 implies that the direct link is never to be terminated based on a timeout.
SupportedRates	Set of integers	1–127 inclusive (for each integer in the set)	The set of data rates that are supported by the peer MAC entity.

10.3.25.3.3 When generated

This primitive is generated by the MLME as result of the establishment of a direct link with a specific peer MAC entity that resulted from a DLS procedure that was initiated by that specific peer MAC entity.

10.3.25.3.4 Effect of receipt

The SME is notified of the establishment of the DLS.

10.3.25.4 MLME-DLSTeardown.request

10.3.25.4.1 Function

When initiated by a non-AP QSTA, this primitive requests the teardown of the direct link with a specified peer MAC entity. When initiated by a QAP, this primitive requests the teardown of direct link between two specified MAC entities.

10.3.25.4.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-DLSTeardown.request(
                        PeerMACAddress1,
```

PeerMACAddress2,
ReasonCode
)

Name	Type	Valid range	Description
PeerMACAddress1	MAC-Address	Any valid individual MAC address	Specifies the MAC address of the non-AP QSTA that is the intended immediate recipient of the teardown.
PeerMACAddress2	MAC-Address	Any valid individual MAC address	Specifies the MAC address of the non-AP QSTA with which the DLS is being torn down. This parameter is applicable only at the QAP.
ReasonCode	Enumeration	QSTA_LEAVING, END_DLS, STAKEY_MISMATCH, UNKNOWN_DLS	Indicates the reason why the direct link is being torn down.

10.3.25.4.3 When generated

This primitive is generated by the SME at a QSTA for tearing down a direct link with another non-AP QSTA.

10.3.25.4.4 Effect of receipt

This primitive initiates a direct-link teardown procedure. The MLME subsequently issues an MLME-DLSTeardown.confirm primitive that reflects the results.

10.3.25.5 MLME-DLSTeardown.confirm

10.3.25.5.1 Function

This primitive reports the results of a direct-link teardown attempt with a specified peer MAC entity.

10.3.25.5.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-DLSTeardown.confirm(
PeerMACAddress1,
PeerMACAddress2,
ResultCode
)

Name	Type	Valid range	Description
PeerMACAddress1	MAC-Address	Any valid individual MAC address	Specifies the MAC address of the non-AP QSTA that is the intended immediate recipient of the teardown.
PeerMACAddress2	MAC-Address	Any valid individual MAC address	Specifies the MAC address of the non-AP QSTA with which the direct link is being torn down. This parameter is applicable only at the QAP.
ResultCode	Enumeration	SUCCESS, FAILURE	Indicates the results of the corresponding MLME-DLSTeardown.request primitive.

10.3.25.5.3 When generated

This primitive is generated by the MLME as a result of an MLME-DLSTeardown.request primitive to tear down an established direct link with a specified peer MAC entity.

10.3.25.5.4 Effect of receipt

The SME is notified of the results of the direct-link teardown procedure.

10.3.25.6 MLME-DLSTeardown.indication

10.3.25.6.1 Function

This primitive indicates the teardown of an already established direct link with a specific peer MAC entity.

10.3.25.6.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-DLSTeardown.indication(
                                    PeerMACAddress,
                                    ReasonCode
                                    )
```

Name	Type	Valid range	Description
PeerMACAddress	MAC-Address	Any valid individual MAC address	Specifies the MAC address of the non-AP QSTA that is the sender of the data flow.
ReasonCode	Enumeration	QSTA_LEAVING, END_DLS, STAKEY_MISMATCH, UNKNOWN_DLS, TIMEOUT	Indicates the reason why the direct link is being torn down.

10.3.25.6.3 When generated

This primitive is generated by the MLME as result of the teardown of a direct link with a specific peer MAC entity that resulted from a direct link teardown procedure that was initiated either by that specific peer MAC entity or by the local MAC entity.

10.3.25.6.4 Effect of receipt

The SME is notified of the teardown of the DLS.

10.3.26 Higher layer synchronization support

This mechanism supports the process of synchronization among higher layer protocol entities residing within different wireless STAs. The actual synchronization mechanism in the higher layer is out of the scope of this amendment. In principle, the MLME indicates the transmission/reception of frames with a specific multicast address in the Address 1 field of a data MPDU.

10.3.26.1 MLME-HL-SYNC.request

10.3.26.1.1 Function

This primitive requests activation of the synchronization support mechanism.

10.3.26.1.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-HL-SYNC.request(
GroupAddress
)

Name	Type	Valid range	Description
GroupAddress	MAC-Address	A multicast MAC address	Specifies the multicast MAC address to which the synchronization frames are addressed. A synchronization frame is a data frame with higher layer synchronization information.

10.3.26.1.3 When generated

This primitive is generated by the SME when a higher layer protocol initiates a synchronization process.

10.3.26.1.4 Effect of Receipt

This request activates the synchronization support mechanism at the STA. The MLME subsequently issues an MLME-HL-SYNC.confirm primitive that reflects the results of the higher layer synchronization support request. If the request has been successful and the higher layer synchronization support mechanism has been activated, the MLME issues an MLME-HL-SYNC.indication primitive when a higher layer synchronization frame, which is a data frame with the specified multicast address in Address 1 field, is received or transmitted.

10.3.26.2 MLME-HL-SYNC.confirm

10.3.26.2.1 Function

This primitive confirms the activation of the higher layer synchronization support mechanism.

10.3.26.2.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-HL-SYNC.confirm(
ResultCode,
)

Name	Type	Valid range	Description
ResultCode	Enumeration	SUCCESS, NOT_SUPPORTED	Indicates the result of the MLME-HL-SYNC.request primitive.

10.3.26.2.3 When generated

This primitive is generated by the MLME as a result of an MLME-HL-SYNC.request primitive to activate the higher layer synchronization support mechanism for a particular multicast address.

10.3.26.2.4 Effect of Receipt

The SME is notified of the activation of the higher layer synchronization support mechanism. The result code of NOT_SUPPORTED is issued if the MAC/MLME does not support the higher layer synchronization support mechanism or if the address provided by the MLME-HL-SYNC.request primitive is not a multicast address.

10.3.26.3 MLME-HL-SYNC.indication

10.3.26.3.1 Function

This primitive indicates the last symbol on air of a higher layer synchronization frame, whether transmitted or received by the MAC.

10.3.26.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-HL-SYNC.indication(
                        SourceAddress,
                        SequenceNumber
                        )
```

Name	Type	Valid range	Description
SourceAddress	MAC-Address	Any valid individual MAC address	Specifies the SA of the STA that transmitted the higher layer synchronization frame.
SequenceNumber	Integer	As defined in frame format	Specifies the sequence number of the higher layer synchronization frame received or transmitted.

10.3.26.3.3 When generated

This primitive is generated by the MLME when the successful reception or transmission of a higher layer synchronization frame is detected, as indicated by the PHY_RXEND.indication or PHY_TXEND.confirm primitives generated by the PHY. The higher layer synchronization frame is identified by the multicast MAC address registered by an earlier MLME-HL-SYNC.request primitive in the Address 1 field of a data frame.

10.3.26.3.4 Effect of Receipt

The SME is notified of the reception or transmission of a higher layer synchronization frame.

10.3.27 Block Ack

This mechanism supports the initiation (or modification) and termination of Block Ack.

The primitives used for this mechanism are called *Block Ack primitives*, which include MLME-ADDBA.xxx and MLME-DELBA.xxx primitives, where xxx denotes request, confirm, indication, or response. Each primitive contains parameters that correspond to a Block Ack Action frame body. Requests and responses

may cause these frames to be sent. Confirms and indications are emitted when an appropriate Block Ack Action frame is received.

10.3.27.1 MLME-ADDBA.request

10.3.27.1.1 Function

This primitive requests the initiation (or modification) of Block Ack with a peer MAC entity.

10.3.27.1.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-ADDBA.request(
PeerQSTAAddress,
DialogToken,
TID,
BlockAckPolicy,
BufferSize,
BlockAckTimeout,
ADDBAFailureTimeout,
BlockAckStartingSequenceControl
)

Name	Type	Valid range	Description
PeerQSTAAddress	MAC-Address	N/A	Specifies the address of the peer MAC entity with which to perform the Block Ack initiation (or modification).
DialogToken	Integer	0–255	Specifies a number unique to the Block Ack Action primitives and frames used in defining the Block Ack.
TID	Integer	0–15	Specifies the TID of the data.
BlockAckPolicy	Enumeration	Immediate, Delayed	Specifies the Block Ack policy.
BufferSize	Integer	0–127	Specifies the number of MPDUs that can be held in its buffer.
BlockAckTimeout	Integer	0–65 535	Specifies the number of TUs without a frame exchange between peers after which the Block Ack is considered to be torn down.
ADDBAFailure-Timeout	Integer	Greater than or equal to 1	Specifies a time limit (in TU) after which the Block Ack setup procedure is terminated.
BlockAckStarting-SequenceControl	As defined in frame format	As defined in frame format	Specifies the value of Block Ack starting sequence control.

10.3.27.1.3 When generated

This primitive is generated by the SME at a QSTA to request initiation (or modification) of Block Ack with the specified peer MAC entity.

10.3.27.1.4 Effect of receipt

The QSTA sends the ADDBA Request frame to the specified peer MAC entity.

10.3.27.2 MLME-ADDBA.confirm

10.3.27.2.1 Function

The primitive reports the results of initiation (or modification) of the Block Ack attempt with the specified peer MAC entity.

10.3.27.2.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-ADDBA.confirm(
PeerQSTAAddress,
DialogToken,
TID,
ResultCode,
BlockAckPolicy,
BufferSize,
BlockAckTimeout
)

Name	Type	Valid range	Description
PeerQSTAAddress	MAC-Address	N/A	Specifies the address of the peer MAC entity with which the Block Ack initiation (or modification) was attempted. This value must match the PeerQSTA-Address parameter specified in MLME-ADDBA.request primitive.
DialogToken	Integer	0–255	Specifies a number unique to the Block Ack Action primitives and frames used in defining the Block Ack. This value must match the DialogToken parameter specified in MLME-ADDBA.request primitive.
TID	Integer	0–15	Specifies the TID of the data. This value must match the TID specified in MLME-ADDBA.request primitive.
ResultCode	Enumeration	SUCCESS, INVALID_ PARAMETERS, REFUSED, TIMEOUT	Indicates the result of the corresponding MLME-ADDBA.request primitive.
BlockAckPolicy	Enumeration	Immediate, Delayed	Specifies the Block Ack policy.
BufferSize	Integer	0–127	Specifies the maximum number of MPDUs in the block for the specified TID.
BlockAckTimeout	Integer	0–65 535	Specifies the number of TUs without a frame exchange between peers after which the Block Ack is considered to be torn down.

10.3.27.2.3 When generated

This primitive is generated by the MLME as a result of an MLME-ADDBA.request primitive to indicate the results of that request.

10.3.27.2.4 Effect of receipt

The SME is notified of the results of the Block Ack initiation (or modification).

10.3.27.3 MLME-ADDBA.indication

10.3.27.3.1 Function

This primitive reports the initiation (or modification) of Block Ack by a peer MAC entity.

10.3.27.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-ADDBA.indication(
                        PeerQSTAAddress,
                        DialogToken,
                        TID,
                        BlockAckPolicy,
                        BufferSize,
                        BlockAckTimeout
                        )
```

Name	Type	Valid range	Description
PeerQSTAAddress	MAC-Address	N/A	Specifies the address of the peer MAC entity that requested the Block Ack initiation (or modification).
DialogToken	Integer	0–255	Specifies a number unique to the Block Ack Action primitives and frames used in defining the Block Ack.
TID	Integer	0–15	Specifies the TID of the data.
BlockAckPolicy	Enumeration	Immediate, Delayed	Specifies the Block Ack policy.
BufferSize	Integer	0–127	Specifies the number of MPDUs that can be held in peerQSTAAddress buffer.
BlockAckTimeout	Integer	0–65 535	Specifies the number of TUs without a frame exchange between peers after which the Block Ack is considered to be torn down.

10.3.27.3.3 When generated

This primitive is generated by the MLME as a result of receipt of a Block Ack initiation (or modification) by the specified peer MAC entity in the form of an ADDBA Request frame.

10.3.27.3.4 Effect of receipt

The SME is notified of the initiation (or modification) of the Block Ack by the specified peer MAC entity.

10.3.27.4 MLME-ADDBA.response

10.3.27.4.1 Function

The primitive responds to the initiation (or modification) by a specified peer MAC entity.

10.3.27.4.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-ADDBA.response(
PeerQSTAAddress,
DialogToken,
TID,
ResultCode,
BlockAckPolicy,
BufferSize,
BlockAckTimeout
)

Name	Type	Valid range	Description
PeerQSTAAddress	MAC-Address	N/A	Specifies the address of the peer MAC entity that attempted the Block Ack initiation (or modification). This value must match the PeerQSTAAddress parameter specified in MLME-ADDBA.indication primitive.
DialogToken	Integer	0–255	Specifies a number unique to the Block Ack Action primitives and frames used in defining the Block Ack. This value must match the DialogToken parameter specified in MLME-ADDBA.indication primitive.
TID	Integer	0–15	Specifies the TID of the data. This value must match the TID specified in MLME-ADDBA.indication primitive.
ResultCode	Enumeration	SUCCESS, REFUSED, INVALID_PARAMETERS, TIMEOUT	Indicates the result of the corresponding MLME-ADDBA.indication primitive.
BlockAckPolicy	Enumeration	Immediate, Delayed	Specifies the Block Ack policy. Undefined when the result code is REFUSED.
BufferSize	Integer	0–127	Specifies the maximum number of MPDUs in the block for the specified TID. Undefined when the result code is REFUSED.
BlockAckTimeout	Integer	0–65 535	Specifies the number of TUs without a frame exchange between peers after which the Block Ack is considered to be torn down.

10.3.27.4.3 When generated

This primitive is generated by the MLME as a result of an MLME-ADDBA.indication primitive to initiate Block Ack by the specified peer MAC entity.

10.3.27.4.4 Effect of receipt

The primitive causes the MAC entity to send an ADDBA Response frame to the specified peer MAC entity.

10.3.27.5 MLME-DELBA.request

10.3.27.5.1 Function

This primitive requests the deletion of Block Ack with a peer MAC entity.

10.3.27.5.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-DELBA.request(
PeerQSTAAddress,
Direction,
TID,
ReasonCode
)

Name	Type	Valid range	Description
PeerQSTAAddress	MAC-Address	N/A	Specifies the address of the peer MAC entity with which to perform the Block Ack deletion.
Direction	Enumeration	Originator, Recipient	Specifies if the MAC entity initiating the MLME-DELBA.request primitive is the originator or the recipient of the data stream that uses the Block Ack.
TID	Integer	0–15	Specifies the TID of the MSDUs for which this Block Ack has been set up.
ReasonCode	Enumeration	QSTA_LEAVING, END_BA, UNKNOWN_BA	Indicates the reason why the Block Ack is being deleted.

10.3.27.5.3 When generated

This primitive is generated by the SME at a QSTA to request deletion of Block Ack with the specified peer MAC entity.

10.3.27.5.4 Effect of receipt

The QSTA sends the DELBA frame to the specified peer MAC entity.

10.3.27.6 MLME-DELBA.confirm

10.3.27.6.1 Function

The primitive reports the transmission status of the Block Ack deletion attempt with a specified peer MAC entity.

10.3.27.6.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-DELBA.confirm(
PeerQSTAAddress,
Direction,
TID,
ResultCode,
)

Name	Type	Valid range	Description
PeerQSTAAddress	MAC-Address	N/A	Specifies the address of the peer MAC entity with which the Block Ack deletion was attempted. This value must match the PeerQSTAAddress parameter specified in MLME-DELBA.request primitive.
Direction	Enumeration	Originator, Recipient	Specifies if the MAC entity initiating the MLME-DELBA.request primitive is the originator or the recipient of the data stream that uses the Block Ack.
TID	Integer	0–15	Specifies the TID of the MSDUs for which this Block Ack has been set up.
ResultCode	Enumeration	SUCCESS, INVALID_ PARAMETERS, FAILURE	Indicates the result of the corresponding MLME-DELBA.request primitive.

10.3.27.6.3 When generated

This primitive is generated by the MLME as a result of an MLME-DELBA.request primitive to indicate the results of that request.

10.3.27.6.4 Effect of receipt

The SME is notified of the results of the Block Ack deletion.

10.3.27.7 MLME-DELBA.indication

10.3.27.7.1 Function

This primitive reports the deletion of Block Ack by a peer MAC entity.

10.3.27.7.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-DELBA.indication(
                        PeerQSTAAddress,
                        Direction,
                        TID,
                        ReasonCode
                        )
```

Name	Type	Valid range	Description
PeerQSTAAddress	MAC-Address	N/A	Specifies the address of the peer MAC entity with which to perform the Block Ack deletion.
Direction	Enumeration	Originator, Recipient	Specifies if the MAC entity initiating the MLME-DELBA.request primitive is the originator or the recipient of the data stream that uses the Block Ack.

Name	Type	Valid range	Description
TID	Integer	0–15	Specifies the TID of the MSDUs for which this Block Ack has been set up.
ReasonCode	Enumeration	QSTA_LEAVING, END_BA, UNKNOWN_BA, TIMEOUT	Indicates the reason why the Block Ack is being deleted.

10.3.27.7.3 When generated

This primitive is generated by the MLME as a result of receipt of a Block Ack deletion by the specified peer MAC entity in the form of a DELBA frame.

10.3.27.7.4 Effect of receipt

The SME is notified of the deletion of the Block Ack by the specified peer MAC entity.

10.3.28 Schedule element management

This clause describes the management procedures associated with the QoS Schedule element.

The primitives defined are MLME-SCHEDULE.request, MLME-SCHEDULE.confirm, and MLME-SCHEDULE.indication.

10.3.28.1 MLME-SCHEDULE.request

10.3.28.1.1 Function

This primitive requests transmission of a Schedule frame. It is valid at the HC.

10.3.28.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-SCHEDULE.request(
                        Non-APQSTAAddress,
                        Schedule
                        )
```

Name	Type	Valid range	Description
Non-APQSTA-Address	MAC-Address	Any valid individual address	MAC address of the non-AP QSTA to which the Schedule frame shall be sent.
Schedule	As defined in frame format	As defined in frame format	Specifies the schedule for the non-AP QSTA, including the SI (minimum and maximum), TXOP duration (minimum and maximum), and specification interval.

10.3.28.1.3 When generated

This primitive is generated by the SME at the HC to send the schedule information, in the form of a Schedule frame, to a specified non-AP QSTA when the schedule information for the non-AP QSTA is changed.

10.3.28.1.4 Effect of receipt

This primitive causes the MAC entity at the HC to send a Schedule frame to the non-AP QSTA specified in the primitive containing the specified Schedule parameters.

10.3.28.2 MLME-SCHEDULE.confirm

10.3.28.2.1 Function

This primitive reports the transmission status of a MLME-SCHEDULE.request primitive.

10.3.28.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-SCHEDULE.confirm(
                                    ResultCode,
                                    )
```

Name	Type	Valid range	Description
ResultCode	Enumeration	SUCCESS, INVALID_ PARAMETERS, UNSPECIFIED_ FAILURE	Indicates the results of the corresponding MLME-SCHEDULE.request primitive.

10.3.28.2.3 When generated

This primitive is generated by the MLME as a result of an MLME-SCHEDULE.request primitive when the action completes.

10.3.28.2.4 Effect of receipt

The SME is notified of the result of the MLME-SCHEDULE.request primitive. If the result code is SUCCESS, the Schedule element has been correctly sent by the HC to the non-AP QSTA in the Schedule frame.

10.3.28.3 MLME-SCHEDULE.indication

10.3.28.3.1 Function

This primitive reports the reception of a new schedule by the non-AP QSTA in the form of a Schedule frame. It is valid at the non-AP QSTA.

10.3.28.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-SCHEDULE.indication(
                                    Schedule
                                    )
```

Name	Type	Valid range	Description
Schedule	As defined in frame format	As defined in frame format	Specifies the schedule for the non-AP QSTA, including the SI (minimum and maximum), TXOP duration (minimum and maximum), and specification interval.

10.3.28.3.3 When generated

This primitive is generated by the MLME as a result of receipt of a new schedule in the form of a Schedule frame.

10.3.28.3.4 Effect of receipt

The SME is notified of the receipt of QoS schedule in the form of a Schedule frame. The new Schedule parameters overwrite the previously stored values.

End of changes to Clause 10.

11. MAC sublayer management entity (MLME)

11.2 Power management

11.2.1 Power management in an infrastructure network

Insert the following paragraph at the end of 11.2.1:

A non-AP QSTA may be in PS mode before the setup of DLS or Block Ack. Once DLS is set up with another non-AP QSTA, the non-AP QSTA suspends the PS mode and shall always be awake. When a STA enters normal (non-APSD) PS mode, any downlink Block Ack agreement without an associated schedule is suspended for the duration of this PS mode. MSDUs for TID without a schedule are sent using Normal Ack following a PS-poll as described in rest of this clause. Uplink Block Ack, Block Acks for any TID with a schedule, and any Block Acks to APSD QSTA continue to operate normally.

Insert after 11.2.1.3 the following new subclause as 11.2.1.4 (instructions for renumbering the existing 11.2.1.4 are given after this new text):

11.2.1.4 Power management with APSD

QAPs capable of supporting automatic power save delivery (APSD) shall signal this capability through the use of the APSD subfield in the Capability Information field in Beacon, Probe Response, and (Re)Association Response management frames.

Non-AP QSTAs use the Power Management field in the Frame Control field of a frame to indicate whether it is in active or PS mode. As APSD is a mechanism for the delivery of downlink frames to power-saving STAs, the frames of a non-AP QSTA using APSD shall have the Power Management bit in the Frame Control field set to 1 for buffering to take place at the QAP.

APSD defines two delivery mechanisms, namely *unscheduled APSD* (U-APSD) and *scheduled APSD* (S-APSD). Non-AP QSTAs may use U-APSD to have some or all of their frames delivered during unscheduled SPs. Non-AP QSTAs may use S-APSD to schedule delivery of some or all of their frames during scheduled SPs.

If there is no unscheduled SP in progress, the unscheduled SP begins when the QAP receives a trigger frame from a non-AP QSTA, which is a QoS data or QoS Null frame associated with an AC the STA has configured to be trigger-enabled. An unscheduled SP ends after the QAP has attempted to transmit at least one MSDU or MMPDU associated with a delivery-enabled AC and destined for the non-AP QSTA, but no more than the number indicated in the Max SP Length field if the field has a nonzero value.

In order to configure a QAP to deliver frames during an unscheduled SP, the non-AP QSTA designates one or more of its ACs to be delivery-enabled and one or more of its AC to be trigger-enabled. A non-AP QSTA may configure a QAP to use U-APSD using two methods. First, a non-AP QSTA may set individual U-APSD Flag bits in the QoS Info subfield of the QoS Capability element carried in (Re)Association Request frames. When a U-APSD Flag bit is set, it indicates that the corresponding AC is both delivery- and trigger-enabled. When all four U-APSD Flag subfields are set to 1 in (Re)Association Request frames, all the ACs associated with the non-AP QSTA are trigger- and delivery-enabled during (re)association. When all four U-APSD Flag subfields are set to 0 in (Re)Association Request frames, none of the ACs associated with the non-AP QSTA is trigger- or delivery-enabled during (re)association.

Alternatively, a non-AP QSTA may designate one or more AC as trigger-enabled and one or more AC as delivery-enabled by sending an ADDTS Request frame per AC to the QAP with the APSD subfield set to 1 and the Schedule subfield set to 0 in the TS Info field in the TSPEC element. APSD settings in a TSPEC request take precedence over the static U-APSD settings carried in the QoS Capability element. In other words, a TSPEC request overwrites any previous U-APSD setting of an AC. The request may be sent for ACs for which the ACM subfield is set to 0.

A non-AP QSTA may set an AC to be trigger- or delivery-enabled for its own use by setting up TSPECs with the APSD subfield set to 1 and the Schedule subfield set to 0 in the uplink or downlink direction, respectively. An uplink TSPEC plus a downlink TSPEC, or a bi-directional TSPEC with the APSD subfield set to 1 and the Schedule subfield set to 0, makes an AC both trigger- and delivery-enabled. An uplink TSPEC plus a downlink TSPEC, or a bi-directional TSPEC with the APSD and the Schedule subfields both set to 0, makes an AC neither trigger- nor delivery-enabled.

A scheduled SP starts at fixed intervals of time specified in the Service Interval field. In order to use a scheduled SP for a TS when the access policy is controlled channel access or for a AC when the access policy is contention-based channel access, a non-AP QSTA shall send an ADDTS Request frame to the QAP with the APSD and Schedule subfields of the TS Info field in the TSPEC element both set to 1. If the APSD mechanism is supported by the QAP and the QAP accepts the corresponding ADDTS Request frame from the non-AP QSTA, the QAP shall respond to the ADDTS Request frame with a response containing the Schedule element indicating that the requested service can be accommodated by the QAP. The first scheduled SP starts when the lower order 4 bytes of the TSF timer equals the value specified in the Service Start Time field. A non-AP QSTA using scheduled SP shall first wake up to receive downlink unicast frames buffered and/or polls from the AP/HC. The STA shall wake up subsequently at a fixed time interval equal to the SI. The QAP may modify the service start time by indicating so in the Schedule element in ADDTS Response frame and in Schedule frames.

A scheduled SP begins at the scheduled wakeup time that corresponds to the SI and the service start time indicated in the Schedule element sent in response to a TSPEC. The STA shall wake up at a subsequent time when

(TSF – service start time) mod minimum SI = 0.

If scheduled services periods are supported in a QBSS, a QSTA may use both unscheduled and scheduled APSD on different ACs at the same time. When a non-AP QSTA establishes scheduled delivery for an AC, that AC shall be considered delivery-enabled. However, the QAP shall not transmit frames associated with that AC during an SP that is initiated by a trigger frame, and it shall not treat frames associated with the AC

that are received from the STA as trigger frames. The QAP shall decline any ADDTS Request frame that indicates the use of both scheduled and unscheduled APSD to be used on the same AC at the same time.

APSD shall be used only to deliver unicast frames to a QSTA. Broadcast/multicast frame delivery shall follow the frame delivery rules defined for broadcast/multicast frames as defined in 11.2.1.5.

Change the following subclause number due to the insertion above of the new subclause, 11.2.1.4, and change the text as shown:

11.2.1.5 ~~11.2.1.4~~ AP operation during the CP

APs shall maintain a Power Management status for each currently associated STA that indicates in which Power Management mode the STA is currently operating. QAPs that implement and signal their support of APSD shall maintain an APSD and an access policy status for each currently associated non-AP QSTA that indicates whether the non-AP QSTA is presently using APSD and shall maintain the schedule (if any) for the non-AP QSTA. An AP shall, depending on the Power Management mode of the STA, temporarily buffer the MSDU or management frame destined to the STA. A QAP implementing APSD shall, if a non-AP QSTA is using APSD and is in PS mode, temporarily buffer the MSDU or management frames destined to that non-AP QSTA. No MSDUs or management frames received for STAs operating in the Active mode shall be buffered for power management reasons.

a) MSDUs, or management frames destined for PS STAs, shall be temporarily buffered in the AP. MSDUs, or management frames, destined for PS QSTAs using APSD shall be temporarily buffered in the APSD-capable QAP. The algorithm to manage this buffering is beyond the scope of this standard, with the exception that if the AP is QoS-enabled, it shall preserve the order of arrival of frames on a per-TID, per-STA basis.

b) MSDUs, or management frames destined for STAs in the active mode, shall be directly transmitted to those STAs.

c) At every beacon interval, the AP shall assemble the partial virtual bitmap containing the buffer status per destination for STAs in the PS mode, and shall send this out in the TIM field of the beacon. At every beacon interval, the APSD-capable QAP shall assemble the partial virtual bitmap containing the buffer status of nondelivery-enabled ACs (if there exists at least one nondelivery-enabled AC) per destination for non-AP QSTAs in PS mode and shall send this out in the TIM field of the Beacon frame. When all ACs are delivery-enabled, the APSD-capable QAP shall assemble the partial virtual bitmap containing the buffer status for all ACs per destination for non-AP QSTAs.

d) If a non-AP QSTA has set up a scheduled SP, it shall automatically wake up at each SP. Therefore, the APSD-capable QAP shall transmit frames associated with admitted traffic with the APSD subfield set to 1 in the TSPECs buffered for the non-AP QSTA during a scheduled SP. If the non-AP QSTA has set up to use unscheduled SPs, the QAP shall buffer frames belonging to delivery-enabled ACs until it has received a trigger frame associated with a trigger-enabled AC from the non-AP QSTA, which indicates the start of an unscheduled SP. A trigger frame received by the QAP from a non-AP QSTA that already has an unscheduled SP underway shall not trigger the start of a new unscheduled SP. The QAP transmits frames destined for the non-AP QSTA and associated with delivery-enabled ACs during an unscheduled SP. The bit for AID 0 (zero) shall be set to 1 when broadcast or multicast traffic is buffered.

e) ~~d)~~ All broadcast/multicast MSDUs, with the Order bit in the Frame Control field clear, shall be buffered if any associated STAs are in PS mode.

f) ~~e)~~ Immediately after every DTIM, the AP shall transmit all buffered broadcast/multicast MSDUs. The More Data field of each broadcast/multicast frame shall be set to indicate the presence of further buffered broadcast/multicast MSDUs. If the AP is unable to transmit all of the buffered broadcast/multicast MSDUs before the TBTT following the DTIM, the AP shall indicate that it will continue to deliver the broadcast/multicast MSDUs by setting the bit for AID 0 (zero) of the TIM element of every Beacon frame, until all buffered broadcast/multicast frames have been transmitted.

g) ~~f)~~ A single buffered MSDU or management frame for a STA in the PS mode shall be forwarded to the STA after a PS-Poll has been received from that STA. For a non-AP QSTAs using U-APSD, the QAP transmits one frame destined for the non-AP QSTA from any AC that is not delivery-enabled in response to PS-Poll from the non-AP QSTA. When all ACs associated with the non-AP QSTA are delivery-enabled, QAP transmits one frame from the highest priority AC. For a STA in PS mode and not using U-APSD, t~~T~~he More Data field shall be set to indicate the presence of further buffered MSDUs or management frames for the polling STA. For a non-AP QSTA using U-APSD, the More Data field shall be set to indicate the presence of further buffered MSDUs or management frames that do not belong to delivery-enabled ACs. When all ACs associated with the non-AP QSTA are delivery-enabled, the More Data field shall be set to indicate the presence of further buffered MSDUs or management frames belonging to delivery-enabled ACs. If there are buffered frames to transmit to the STA, the QAP may set the More Data bit in a QoS +CF-Ack frame to 1, in response to a QoS data frame to indicate that it has one or more pending frames buffered for the PS STA identified by the RA address in the QoS +CF-Ack frame. A QAP may also set the More Data bit in an ACK frame in response to a QoS data frame to indicate that it has one or more pending frames buffered for the PS STA identified by the RA address in the ACK frame, if that PS STA has set the More Data Ack subfield in the QoS Capability information element to 1. Further PS-Poll frames from the same STA shall be acknowledged and ignored until the MSDU or management frame has either been successfully delivered or presumed failed due to maximum retries being exceeded. This prevents a retried PS-Poll from being treated as a new request to deliver a buffered frame.

h) At each scheduled APSD SP for a non-AP QSTA, the APSD-capable QAP shall attempt to transmit at least one MSDU or MMPDU, associated with admitted TSPECs with the APSD and Schedule subfields both set to 1, that are destined for the non-AP QSTA. At each unscheduled SP for a non-AP QSTA, the QAP shall attempt to transmit at least one MSDU or MMPDU, but no more than the value specified in the Max SP Length field in the QoS Capability element from delivery-enabled ACs, that are destined for the non-AP QSTA. The More Data bit of the directed data or management frame associated with delivery-enabled ACs and destined for that non-AP QSTA indicates that more frames are buffered for the delivery-enabled ACs. The More Data bit set in MSDUs or management frames associated with nondelivery-enabled ACs and destined for that non-AP QSTA indicates that more frames are buffered for the nondelivery-enabled ACs. For all frames except for the final frame of the SP, the EOSP subfield of the QoS Control field of the QoS data frame shall be set to 0 to indicate the continuation of the SP. A QAP may also set the More Data bit to 1 in a QoS +CF-Ack frame in response to a QoS data frame to indicate that it has one or more pending frames buffered for the target STA identified by the RA address in the QoS +CF-Ack frame. If the QoS data frame is associated with a delivery-enabled AC, the More Data bit in the QoS +CF-Ack frame indicates more frames for all delivery-enabled ACs. If the QoS data frame is not associated with a delivery-enabled AC, the More Data bit in the QoS +CF-Ack frame indicates more frames for all ACs that are not delivery-enabled. The QAP considers APSD QSTA to be in awake state after it has sent a QoS +CF-Ack frame, with the EOSP subfield in the QoS Control field set to 0, to the APSD QSTA. If necessary, the QAP may generate an extra QoS Null frame, with the EOSP set to 1. When the QAP has transmitted a directed frame to the non-AP QSTA with the EOSP subfield set to 1 during the SP except for retransmissions of that frame, the QAP shall not transmit any more frames using this mechanism until the next SP. The QAP shall set the EOSP subfield to 1 to indicate the end of the SP in APSD.

i) If the QAP does not receive an acknowledgment to a directed MSDU or management frame sent to a non-AP QSTA in PS mode following receipt of a PS-Poll from that non-AP QSTA, it may retransmit the frame for at most the lesser of the maximum retry limit and the MIB attribute dot11QAPMissingAckRetryLimit times before the next beacon, but it shall retransmit that frame at least once before the next beacon, time permitting and subject to its appropriate lifetime limit. If an acknowledgment to the retransmission is not received, it may wait until after the next Beacon frame to further retransmit that frame subject to its appropriate lifetime limit.

j) If the QAP does not receive an acknowledgment to a directed MSDU sent with the EOSP subfield set to 1, it shall retransmit that frame at least once within the same SP, subject to applicable retry or

lifetime limit. The maximum number of retransmissions within the same SP is the lesser of the maximum retry limit and the MIB attribute dot11QAPMissingAckRetryLimit. If an acknowledgment to the retransmission of this last frame in the same SP is not received, it may wait until the next SP to further retransmit that frame, subject to its applicable retry or lifetime limit.

k) ~~g)~~An AP shall have an aging function to delete pending traffic when it is buffered for an excessive time period. The QAP may base the aging function on the listen interval specified by the non-AP QSTA in the (Re)Association Request frame.

l) ~~h)~~When an AP is informed that a STA changes to the Active mode, then the AP shall send buffered MSDUs and management frames (if any exist) to that STA without waiting for a PS-Poll. When a QAP is informed that an APSD-capable non-AP QSTA is not using APSD, then the QAP shall send buffered MSDUs and management frames (if any exist) to that non-AP QSTA according to the rules corresponding to the current PS mode of the non-AP QSTA.

Change the following subclause number due to the insertion above of the new subclause, 11.2.1.4:

11.2.1.6 ~~11.2.1.5~~ AP operation during the CFP

Change the following subclause number due to the insertion above of the new subclause, 11.2.1.4:

11.2.1.7 ~~11.2.1.6~~ Receive operation for STAs in PS mode during the CP

Change the text in item c) of the lettered list in 11.2.1.7 as shown:

c) The STA shall remain in the awake state until it receives the response to its poll or it receives another beacon whose TIM indicates that the AP does not have any MSDUs or management frames buffered for this STA. If the bit corresponding to the STA's AID is set in the subsequent TIM, the STA shall issue another PS-Poll to retrieve the buffered MSDU or management frame(s). When a non-AP QSTA that is using U-APSD and has all ACs delivery-enabled detects that the bit corresponding to its AID is set in the TIM, the non-AP QSTA shall issue a trigger frame or a PS-Poll frame to retrieve the buffered MSDU or management frames.

Change the text in item e) of the lettered list in 11.2.1.7 as shown:

e) When ReceiveDTIMs is true, the STA shall wake up early enough to be able to receive every DTIM. A STA receiving broadcast/multicast MSDUs shall remain awake until the More Data field of the broadcast/multicast MSDUs indicates there are no further buffered broadcast/multicast MSDUs or until a TIM is received indicating there are no more buffered broadcast/multicast MSDUs. If a non-AP QSTA receives a QoS +CF-Ack frame from its QAP with the More Data bit set to 1, then the QSTA shall operate exactly as if it received a TIM with its AID bit set. If a non-AP QSTA has set the More Data Ack subfield in QoS Capability information element to 1, then if it receives an ACK frame from its QAP with the More Data bit set to 1, the QSTA shall operate exactly as if it received a TIM with its AID bit set. For example, a QSTA that is using the PS-Poll delivery method shall issue a PS-Poll frame to retrieve a buffered frame.

Change the following subclause number due to the insertion above of the new subclause, 11.2.1.4:

11.2.1.8 ~~11.2.1.7~~ Receive operation for STAs in PS mode during the CFP

Insert after 11.2.1.8 the following new subclause (11.2.1.9):

11.2.1.9 Receive operation for non-AP QSTAs using APSD

A non-AP QSTA using APSD shall operate as follows to receive an MSDU or management frame from the QAP:

a) If a scheduled SP has been set up, the non-AP QSTA wakes up at its scheduled start time. (The non-AP QSTA shall wake up early enough to receive transmissions at the scheduled SP.)

b) If the non-AP QSTA is initiating an unscheduled SP, the non-AP QSTA wakes up and transmits a trigger frame to the QAP. When one or more ACs are not delivery-enabled, the non-AP QSTA may retrieve MSDUs and MMPDUs belonging to those ACs by sending PS-Poll frames to the QAP.

c) The non-AP QSTA shall remain awake until it receives a QoS data frame addressed to it, with the EOSP subfield in the QoS Control field set to 1.

d) The non-AP QSTA may send additional PS-Poll frames if the More Data subfield is set to 1 in downlink unicast data or management frames that do not belong to any deliver-enabled ACs. The non-AP QSTA may send additional trigger frames if the More Data subfield is set to 1 in downlink unicast data or management frames that belong to delivery-enabled ACs.

Change the following subclause numbers due to the insertion above of the new subclauses, 11.2.1.4 and 11.2.1.9:

11.2.1.10 ~~11.2.1.8~~ STAs operating in the Active mode

11.2.1.11 ~~11.2.1.9~~ AP aging function

11.2.2 Power management in an IBSS

Change the text of 11.2.2 as follows:

This subclause specifies the power management mechanism for use within an IBSS.

In a QIBSS, a QSTA shall not operate in PS mode while it has Block Ack set up with another QSTA. A QSTA may be in PS mode before the setup of Block Ack. Once Block Ack is set up with another QSTA, the QSTA suspends the PS mode and shall always remain awake.

Insert after 11.3.4 the following new subclauses as 11.4 through 11.7.5, including the new figures and tables in those subclauses (instructions for renumbering the existing 11.4 through 11.6 are given after this new text):

11.4 TS operation

11.4.1 Introduction

A TSPEC describes the traffic characteristics and the QoS requirements of a TS. The main purpose of the TSPEC is to reserve resources within the HC and modify the HC's scheduling behavior. It also allows other parameters to be specified that are associated with the TS, such as a traffic classifier and acknowledgment policy.

A TS may have one or more TCLAS (within the discretion of the QSTA that sets up the stream) associated with it. The QAP uses the parameters in the TCLAS elements to filter the MSDUs belonging to this TS so that they can be delivered with the QoS parameters that have been set up for the TS.

TSPEC and the optional TCLAS elements are transported on the air by the ADDTS, in the corresponding QoS Action frame and across the MLME SAP by the MLME-ADDTS primitives.

Following a successful negotiation, a TS is created, identified within the non-AP QSTA by its TSID and direction, and identified within the HC by a combination of TSID, direction, and non-AP QSTA address.

It is always the responsibility of the non-AP QSTA to initiate the creation of a TS regardless of its direction.

In the direct-link case, it is the responsibility of the non-AP QSTA that is going to send the data to create the TS. In this case, the non-AP QSTA negotiates with the HC to gain TXOPs that it uses to send the data. There is no negotiation between the originator and recipient non-AP QSTAs concerning the TS: the originator can discover the capabilities of the recipient (rates, BlockAck) using the DLS.

In the case of traffic relayed by a QAP, the sending and receiving non-AP QSTAs may both create individual TS for the traffic. Any traffic classifier created for the downlink TS applies equally regardless of whether the source is in the same BSS or reached through the DS.

A non-AP QSTA may simultaneously support up to eight TSs from the HC to itself and up to eight TSs from itself to other QSTAs, including the HC. The actual number it supports may be less due to implementation restrictions.

A HC may simultaneously support up to eight downlink TSs and up to eight uplink TSs per associated non-AP QSTA. The actual number it supports may be less due to implementation restrictions.

The traffic admitted in context of a TSPEC can be sent using EDCA or HCCA or HEMM. This depends on the access policy set in the TS Info field in the TSPEC. A TSPEC request may be set so that both HCCA and EDCA mechanisms (i.e., HEMM) are used.

11.4.2 TSPEC construction

TSPECs are constructed at the SME, from application requirements supplied via the SME, and with information specific to the MAC layer. There are no normative requirements on how any TSPEC is to be generated. However, in K.3.2, a description is given on how and where certain parameters may be chosen.

11.4.3 TS lifecycle

Figure 68a summarizes the TS lifecycle (using the HMSC syntax defined in ITU-T Z.120 [B24]).

Initially a TS is inactive. A QSTA shall not transmit any QoS data frames using an inactive TS.

Following a successful TS setup initiated by the non-AP QSTA, the TS becomes active, and either the non-AP QSTA or the HC may transmit QoS data frames using this TSID (according to the Direction field).

While the TS is active, the parameters of the TSPEC characterizing the TS can be renegotiated, when the renegotiation is initiated by the non-AP QSTA. This negotiation can succeed, resulting in a change to the TSPEC, or can fail, resulting in no change to the TSPEC.

An active TS becomes inactive following a TS deletion process initiated at either non-AP QSTA or HC. It also becomes inactive following a TS timeout detected at the HC. When an active TS becomes inactive, all the resources allocated for the TS are released.

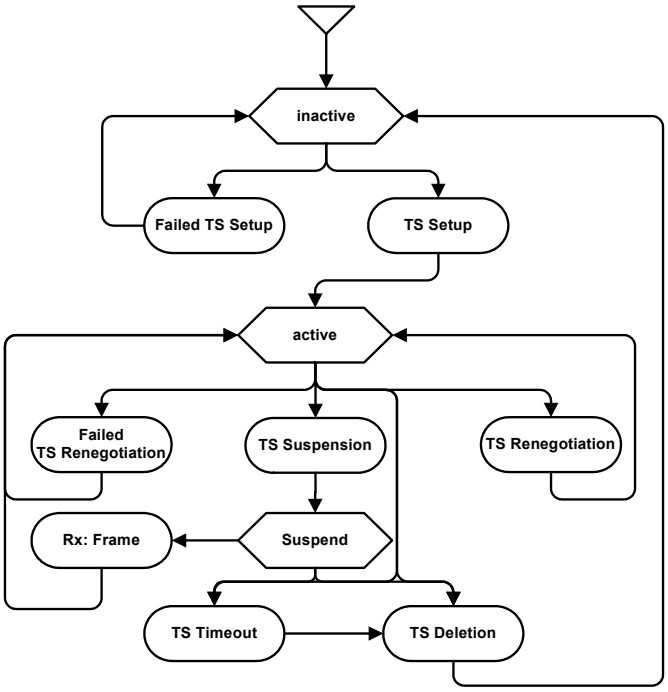


Figure 68a—TS lifecycle

An active TS may become suspended if no activity is detected for a duration of a suspension interval. Upon detection of activity, the TS may be reinstated. While the TS is in the suspended state, the HC shall not reclaim the resources assigned to the TS.

11.4.4 TS setup

Figure 68b shows the sequence of messages occurring at a TS setup. This message sequence in this figure and in the subsequent figures does not show the acknowledgment.

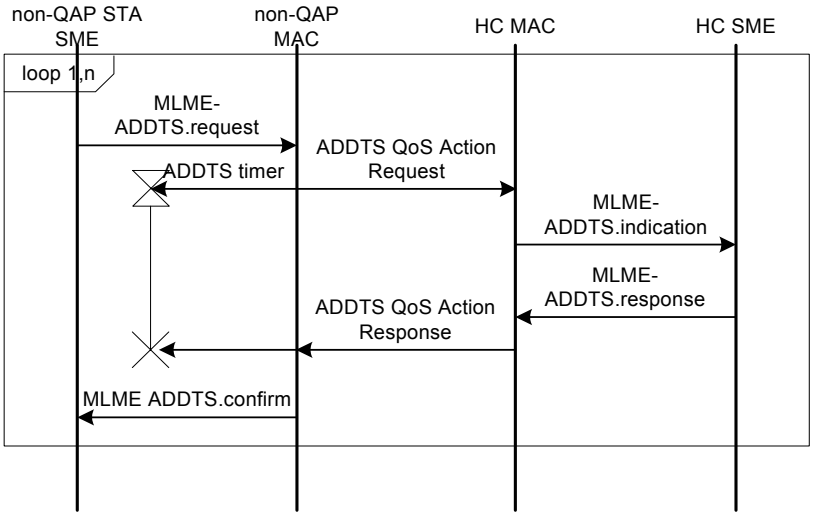


Figure 68b—TS setup

The non-AP QSTA SME decides that a TS needs to be created. How it does this, and how it selects the TSPEC parameters, is beyond the scope of this amendment. The SME generates an MLME-ADDTS.request primitive containing a TSPEC. A TSPEC may also be generated autonomously by the MAC without any initiation by the SME. However, if a TSPEC is generated subsequently by the SME, the TSPEC containing the same TSID generated autonomously by the MAC shall be overridden. If one or more TSPECs are initiated by the SME, the autonomous TSPEC shall be terminated.

The non-AP QSTA MAC transmits the TSPEC in an ADDTS Request frame to the HC and starts a response timer called *ADDTS timer* of duration dot11ADDTSResponseTimout.

The HC MAC receives this management frame and generates an MLME-ADDTS.indication primitive to its SME containing the TSPEC.

The SME in the HC decides whether to admit the TSPEC as specified, refuse the TSPEC, or not admit but suggest an alternative TSPEC. The SME then generates an MLME-ADDTS.response primitive containing the TSPEC and a ResultCode value. The contents of the TSPEC and Status fields contain values specified in 10.3.11.4.2.

The HC MAC transmits an ADDTS Response frame containing this TSPEC and status. The encoding of the ResultCode values to Status Code field values is defined in Table 23b.

Table 23b—Encoding of ResultCode to Status Code field value

ResultCode	Status code
SUCCESS	0
INVALID_PARAMETERS	38
REJECTED_WITH_SUGGESTED_CHANGES	39
REJECTED_FOR_DELAY_PERIOD	47

The non-AP QSTA MAC receives this management frame and cancels its ADDTS timer. It generates an MLME-ADDTS.confirm primitive to its SME containing the TSPEC and status.

The non-AP QSTA SME decides whether the response meets its needs. How it does this is beyond the scope of this amendment. If the result code is SUCCESS, the TS enters into an active state. Otherwise, the whole process can be repeated using the same TSID and direction and a modified TSPEC until the non-AP QSTA SME decides that the granted TXOP is adequate or inadequate and cannot be improved.

The parameters that are set for a TS can be renegotiated in a similar manner, when such a request is generated by the SME through ADDTS.request primitive. When a request for the modification of the TS parameters is accepted by the HC, it shall reset both the suspension interval and the inactivity interval timers.

If the AP/HC grants a TXOP for an ADDTS Request frame with the Ack Policy subfield set to Block Ack and the Direction field set to either downlink or bidirectional, then it shall initiate a Block Ack negotiation by sending an ADDBA Request frame to the non-AP QSTA that originated the ADDTS Request frame. If a non-AP QSTA is granted a TXOP for an ADDTS Request frame with the Ack Policy subfield set to Block Ack and the Direction field set to other than downlink, then it shall initiate a Block Ack negotiation by sending an ADDBA Request frame to the recipient of the TS.

11.4.5 Failed TS setup

There are two possible types of failed TS setup:

a) The transmission of ADDTS Request frame failed.

b) No ADDTS Response frame is received from the HC (e.g., because of delay due to congestion or because the response frame cannot be transmitted).

Figure 68c summarizes the remaining two cases. The MLME shall issue an MLME-ADDTS.confirm primitive, with a result code of TRANSMISSION_FAILURE in the former case and TIMEOUT in the latter case. In both cases, if the request is not for an existing TS, the non-AP QSTA MAC shall send a DELTS to the HC specifying the TSID and direction of the failed request just in case the HC had received the request and it was the response that was lost.

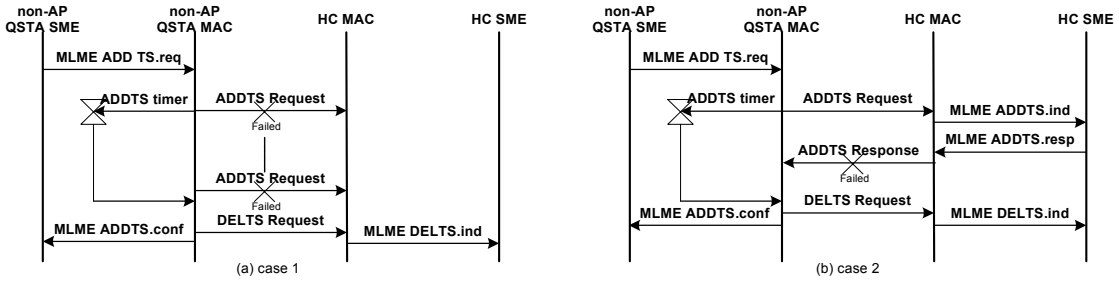


Figure 68c—Failed TS setup detected within non-AP STA MAC

11.4.6 Data transfer

After the setup of a TSPEC, MSDUs are classified above the MAC and are sent to the MAC through the MAC_SAP using the service primitive MA-UNITDATA.request with the priority parameter encoded to the TSID. The MAC delivers the MSDUs based on a schedule using QoS data frames. In the case of a non-AP QSTA, the MSDUs are transmitted using QoS data frames, when the non-AP QSTA is polled by the HC.

The generation of the associated TSID is done by a classifier above the MAC, and it may use the associated TCLAS elements if any are present. When there are multiple TCLASs and if the Processing subfield of TCLAS Processing information element is set to 0, the priority parameter in the associated MA-UNITDATA.request primitive is set to TSID if the classifier can match the parameters in all the TCLAS elements associated with the TS. When there are multiple TCLASs and if the Processing subfield of the TCLAS Processing information element is set to 1, the priority parameter in the associated MA-UNITDATA.request primitive is set to TSID if the classifier can match the parameters in at least one of the TCLAS elements associated with the TS. When there is no TCLAS element and if the Processing subfield of the TCLAS Processing information element is set to 2, the priority parameter in the associated MA-UNITDATA.request primitive is set to TSID if the classifier cannot match the parameters to any of the TCLAS elements.

When the Processing subfield of the TCLAS Processing information element is set to 1, then the receiver cannot recover the original UP unless it does a reverse mapping based on the TCLAS parameters. When the Processing subfield of the TCLAS Processing information element is set to 2, then the receiver cannot recover the original UP.

When an MSDU arrives from the MAC_SAP with a TSID for which there is no associated TSPEC, then the MSDUs shall be sent using EDCA using the access category AC_BE.

11.4.7 TS deletion

There are two types of TS deletion: non-AP QSTA-initiated and HC-initiated. In both cases, the SME entity above the MAC generates an MLME-DELTS.request primitive specifying the TSID and direction of the TS to be deleted and the reason for deleting the TS. This causes the MAC to send a DELTS Action frame. The encoding of ReasonCode values to Reason Code field (see 7.3.1.7) values is defined in Table 23c.

Table 23c—Encoding of ReasonCode to Reason Code field value for DELTS

ReasonCode	Reason Code field
QSTA_LEAVING	36
END_TS	37
UNKNOWN_TS	38
TIMEOUT	39

The TS is considered inactive within the initiating MAC when the ACK frame to the Action frame is received. No Action frame response is generated.

Figure 68d shows the case of TS deletion initiated by the non-AP QSTA and the case of TS deletion initiated by the HC.

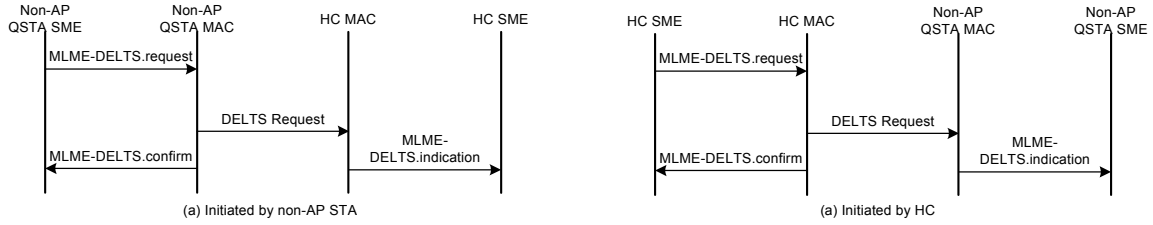


Figure 68d—TS deletion

An HC should not delete a TSPEC without a request from the SME except due to inactivity (see 11.4.8).

All TSPECs that have been set up shall be deleted upon disassociation and reassociation. Reassociation causes the non-AP QSTA and QAP to clear their state, and the non-AP QSTA will have to reinitiate the setup.

11.4.8 TS timeout

TS timeout is detected within the HC MAC when no traffic is detected on the TS within the inactivity time-out specified when the TS was created.

For an uplink TS, the timeout is based on the arrival of correctly received MSDUs that belong to the TS within the MAC after any decryption and reassembly.

For a downlink TS, the timeout is based on the following:

— Arrival of valid MA-UNITDATA.request primitives using this TS at the MAC_SAP when the QoS data frames are sent with the Ack Policy subfield set to No Ack.

— Confirmation of correctly sent MSDUs that belong to the TS within the MAC when the QoS data frames are sent with the Ack Policy subfield set other than to No Ack.

For a direct-link TS, inactivity is considered to have happened if one of the two following happens:

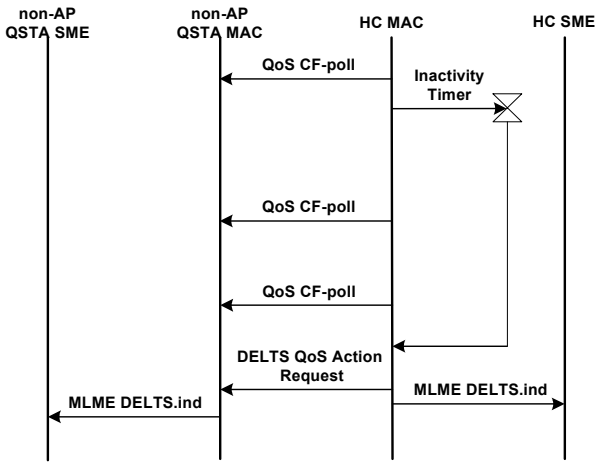
— Returning QoS Null immediately after SIFS interval that contains a zero Queue Size subfield in the QoS Control field in response to a QoS CF-Poll frame.

— No QoS Null frame indicating the queue size for related TSID within a TXOP. This is to ensure that the non-AP QSTA is actually using the assigned TXOP for the given TSID.

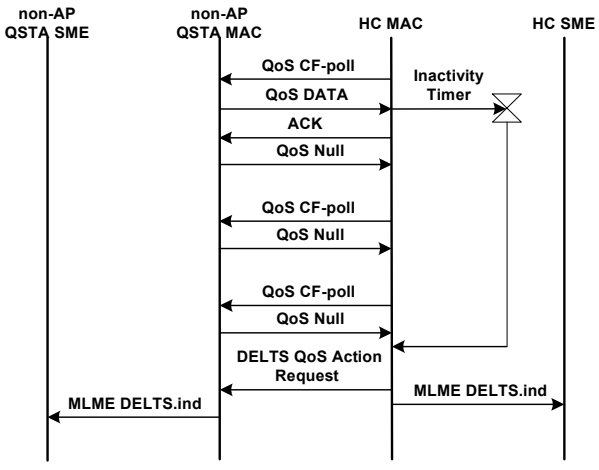
Any other use of a polled TXOP delivered to the non-AP QSTA is considered to be activity on all direct-link TS associated with that non-AP QSTA. Detection of inactivity of this type is optional.

In response to an inactivity timeout, the HC shall send a DELTS frame to the non-AP QSTA with the result code set to TIMEOUT and inform its SME using the MLME-DELTS.indication primitive.

The case of uplink TS timeout is shown in Figure 68e.



(a) No response from QSTA



(b) No Data frames from QSTA

Figure 68e—TS timeout

11.4.9 TS suspension

TS suspension occurs within the HC MAC when no traffic is detected on the TS within the suspension interval specified when the TS was created. In the suspended state, the generation of QoS (+)CF-Poll frames is stopped for the related TS.

11.4.10 TS Reinstatement

A suspended TS may be reinstated following activity for that TS.

For uplink and direct link, a suspended TS may be reinstated by a non-AP QSTA by sending a QoS data or QoS Null frame. This frame may be sent at the highest priority using EDCA. The generation of successive QoS (+)CF-Poll frames shall then be resumed by the HC.

For downlink, a suspended TS is reinstated by the HC when the QAP receives an MSDU from a higher layer.

11.5 Block Ack operation

Block Ack may be set up at the MAC (see 9.10.2) or by the initiation of SME. The setup and deletion of Block Ack at the initiation of the SME is described in this subclause.

11.5.1 Setup and modification of the Block Ack parameters

The procedures for setting up and modifying the Block Ack parameters are described in 11.5.1.1 and 11.5.1.2, respectively, and illustrated in Figure 68f.

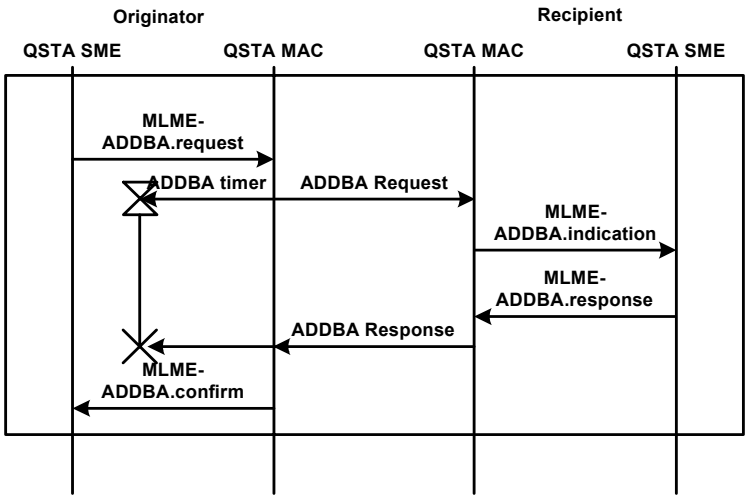


Figure 68f—Block Ack setup

11.5.1.1 Procedure at the originator

Upon receipt of an MLME-ADDBA.request primitive, an initiating QSTA that intends to send QoS data frames under the Block Ack mechanism shall set up the Block Ack using the following procedure:

a) Check whether the intended peer QSTA is capable of participating in the Block Ack mechanism by discovering and examining its "Delayed Block Ack" and "Immediate Block Ack" capability bits. If the recipient is capable of participating, the originator sends an ADDBA frame indicating the TID and the buffer size.

b) If an ADDBA Response frame is received with the matching dialog token and the TID and with a status code set to a value of 0, the QSTA has established a Block Ack mechanism with the recipient QSTA; and the MLME shall issue an MLME-ADDBA.confirm primitive indicating the successful completion of the Block Ack setup.

c) If an ADDBA Response frame is received with the matching dialog token and the TID and with a status code set to a value other than 0, the QSTA has not established a Block Ack mechanism with the recipient QSTA; and the MLME shall issue an MLME-ADDBA.confirm primitive indicating the failure of the Block Ack setup.

d) If there is no response from the recipient within dot11ADDBAFailureTimeout, the QSTA has not established a Block Ack mechanism with the recipient QSTA; and the MLME shall issue an MLME-ADDBA.confirm primitive with a result code of TIMEOUT.

11.5.1.2 Procedure at the recipient

A recipient shall operate as follows in order to support Block Ack initialization and modification:

a) When an ADDBA Request frame is received from another QSTA, the MLME shall issue an MLME-ADDBA.indication primitive.

b) Upon receipt of the MLME-ADDBA.response primitive, the QSTA shall respond by an ADDBA Response frame with a result code as defined in 7.4.4.2.

   1) If the result code is SUCCESS, the Block Ack is considered to be established with the originator. Contained in the frame are the type of Block Ack and the number of buffers that have been allocated for the support of this block.

   2) If the result code is REFUSED, the Block Ack is not considered to have been established.

The encoding of ResultCode values to Status Code field values is defined in Table 23d.

Table 23d—Encoding of ResultCode to Status Code field value

ResultCode	Status Code field
SUCCESS	0
REFUSED	37
INVALID_PARAMETERS	38

11.5.2 Teardown of the Block Ack mechanism

The procedure at the two QSTAs is described in 11.5.2.1 and 11.5.2.2 and illustrated in Figure 68g.

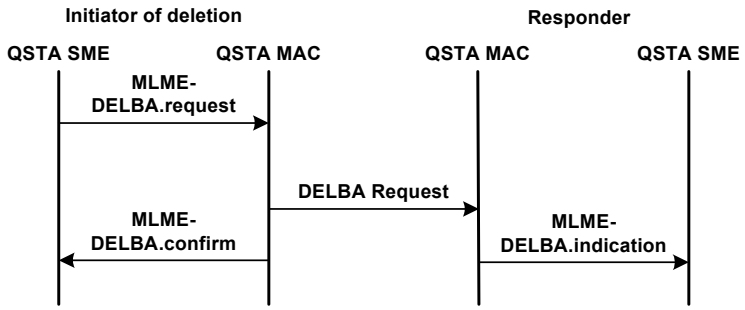


Figure 68g—Block Ack deletion

11.5.2.1 Procedure at the initiator of the Block Ack teardown

Upon receipt of an MLME-DELBA.request primitive, the QSTA MAC/MLME shall tear down the Block Ack using the following procedure:

a) The QSTA shall transmit a DELBA frame.

b) Upon the receipt of an acknowledgment to the DELBA frame, the MLME issues an MLME-DELBA.confirm primitive.

The encoding of ReasonCode values to Reason Code field (see 7.3.1.7) values is defined in Table 23e.

Table 23e—Encoding of ReasonCode to Reason Code field value for DELBA

ReasonCode	Reason Code field
QSTA_LEAVING	36
END_BA	37
UNKNOWN_BA	38
TIMEOUT	39

11.5.2.2 Procedure at the recipient of the DELBA frame

A QSTA shall issue a MLME-DELBA.indication primitive with the parameter ReasonCode having a value of REQUESTED when a DELBA frame is received.

11.5.3 Error recovery upon a peer failure

Every STA shall maintain an inactivity timer for every negotiated Block Ack setup. The inactivity timer at a recipient is reset when MPDUs corresponding to the TID for which the Block Ack policy is set are received and the Ack Policy subfield in the QoS Control field of that MPDU header is set to Block Ack. The inactivity timer is not reset when MPDUs corresponding to other TIDs are received. The inactivity timer at the recipient is also reset when a BlockAckReq frame corresponding to the TID for which the Block Ack policy is set is received. The inactivity timer at the originator is reset when a BlockAck frame corresponding to the TID for which the Block Ack policy is set is received. When a timeout of BlockAckTimeout is detected, the QSTA shall send a DELBA frame to the peer QSTA with the Reason Code field set to TIMEOUT and shall

issue a MLME-DELBA.indication primitive with the ReasonCode parameter having a value of TIMEOUT. The procedure is illustrated in Figure 68h.

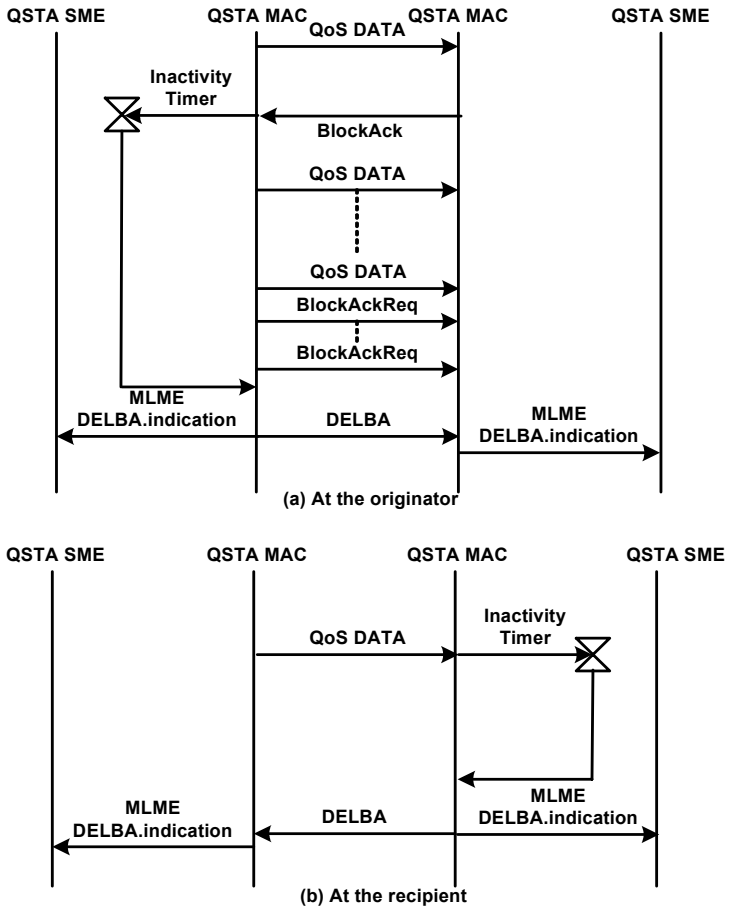


Figure 68h—Error recovery by the receiver upon a peer failure

When a recipient does not have an active Block ack for a TID, but receives data MPDUs with the Ack Policy subfield set to Block Ack, it shall discard them and shall send a DELBA frame using the normal access mechanisms. If such a QSTA receives a BlockAckReq frame, it may respond with an ACK frame and shall respond with a DELBA frame using the normal access mechanisms. The originator may attempt to set up the use of Block Ack or may send the MPDUs using an alternative acknowledgment mechanism. When the recipient transmits a DELBA frame, it shall set the last sequence number received value to the sequence number of the last received MPDU, regardless of the acknowledgment policy used in that frame. When the originator receives a DELBA frame, it shall

a) Discard any MPDU that has been transmitted and not acknowledged, with the possible exception if it was the last MPDU to be sent and it was not a retransmission, and

b) Set the sequence number to either that of the last MPDU that is sent if it intends to retransmit or one beyond the last MPDU sent.

The originator QSTA may send an ADDBA Request frame in order to update Block ACK Timeout Value. If the updated ADDBA Request frame is accepted, both QSTAs initialize the timer to detect Block ACK time-out. Even if the updated ADDBA Request frame is not accepted, the original Block ACK setup remains active.

11.6 Higher layer timer synchronization

11.6.1 Introduction

Higher layer timer synchronization is beyond the scope of this amendment. However, explanation on how the constructs in this amendment can be used to support such capabilities may be useful to the designer.

One way to accomplish synchronization across a BSS is by multicasting synchronization (Sync) packets from the higher layers containing a time stamp and a sequence number. These packets would be opaque to the MAC and would be transported in the same way as any other MSDU (most likely addressed to the multicast address). Sync packets would be treated as a type of management packet by the higher layers. The time stamp in the Sync packet would contain the higher layer clock value at the time when the previous Sync packet was transmitted. The sequence number parameter has a value equal to the sequence number of the MSDU in which the time stamp is provided.

The reason the packet would contain the time stamp for the previous Sync packet (rather than the current packet) is that hardware and layering constraints would prohibit the ability to provide a time stamp for the exact instant the current packet is transmitted within that packet. However, a MLME-HL-SYNC.indication primitive allows the transmitting QSTA to know exactly when each Sync packet is transmitted. A receiving QSTA can similarly note the time when each Sync packet is received as well as the sequence number for that frame. The receiving QSTA would save this receive time indication for each packet along with its sequence number and compare the indication of the previously received Sync packet to the time stamp in the most currently received packet. This comparison allows the STA to compute an offset, which can be used to adjust its time reference to match that of the synchronization source. The sequence number would ensure that the correct packet is being used for time stamp comparison. It is possible a packet is lost; in this case, the received time stamp for the lost packet should be discarded.

The last symbol of the Sync frame is indicated by the PHY using the PHY-TXEND.confirm and PHY-RXEND.indication primitives in the transmitter and receiver of the Sync frame, respectively. Practical limits on the coincidence of this indication and the last symbol of the Sync frame are implementation dependent. The accuracy of this technique also depends on the propagation delay between the source and receiving channel. However, both the time difference (between the PHY indication and the last symbol of the Sync frame) and the propagation delay can be considered as fixed-delay components. Therefore, they contribute only to the fixed time difference between the transmitter and receiver STAs' clocks and do not contribute to jitter. An implementation-dependent scheme can be used to cancel or minimize this fixed time difference.

The Sync frame may also be relayed through the AP. In this case, the non-AP QSTA that generates the time stamps notes the reception of the multicast Sync frame from the AP as the indication to save the higher clock value for the next Sync frame. Receiving QSTAs would also similarly note the time when each Sync packet is received from the AP. The sequence number would include a value corresponding to the frame received from the AP.

Other implementations (aside from what is described here) may also be supported.

11.6.2 Procedure at the QSTA

A MAC that supports the MLME-HL-SYNC primitives shall respond to a MLME-HL-SYNC.request primitive with an MLME-HL-SYNC.confirm primitive containing a result code of SUCCESS. This confirms to the requesting application that the specified group address has been entered into a MAC table of group addresses for which MLME-HL-SYNC.indication primitive shall be provided.

In order to determine whether to provide an MLME-HL-SYNC.indication primitive for a particular data frame, a MAC that supports MLME-HL-SYNC primitives compares the Address 1 field in a data frame's MAC header against a list of group addresses previously registered by an MLME-HL-SYNC.request

primitive. If the MAC and the transmitter of the Sync frame are collocated within the same STA, the MLME-HL-SYNC.indication primitive shall occur when the last symbol of a matching data frame is transmitted. Otherwise, the indication shall occur when the last symbol of the matching data frame is received. In both cases, the MLME-HL-SYNC.indication primitive provided is simultaneous (within implementation-dependent delay bounds) with the indication provided to other STAs within the BSS for the same data frame.

11.7 DLS operation

In general, STAs are not allowed to transmit frames directly to other STAs in a BSS and should always rely on the AP for the delivery of the frames. However, STAs with QoS facility (i.e., QSTAs) may transmit frames directly to another QSTA by setting up such data transfer using DLS. The need for this protocol is motivated by the fact that the intended recipient may be in PS mode, in which case it can be awakened only by the QAP. The second feature of DLS is to exchange rate set and other information between the sender and the receiver.

This protocol prohibits the STAs going into PS mode for the duration of the direct stream as long as there is an active DLS between the two STAs.

DLS does not apply in a QIBSS, where frames are always sent directly from one STA to another.

The handshake involved in the setup is illustrated in Figure 68i and involves the four steps listed after the figure.

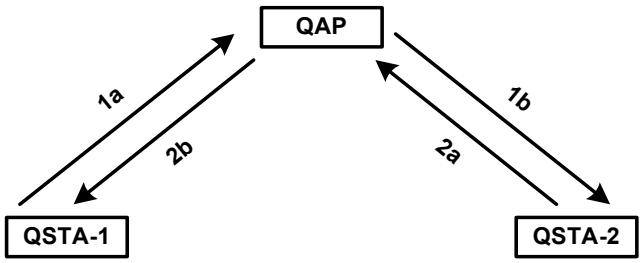


Figure 68i—The four steps involved in direct-link handshake

a) A STA, QSTA-1, that intends to exchange frames directly with another non-AP STA, QSTA-2, invokes DLS and sends a DLS Request frame to the QAP (step 1a in Figure 68i). This request contains the rate set, capabilities of QSTA-1, and the MAC addresses of QSTA-1 and QSTA-2.

b) If QSTA-2 is associated in the BSS, direct streams are allowed in the policy of the BSS, and QSTA-2 is indeed a QSTA, then the QAP forwards the DLS Request frame to the recipient, QSTA-2 (step 1b in Figure 68i).

c) If QSTA-2 accepts the direct stream, it sends a DLS Response frame to the QAP (step 2a in Figure 68i), which contains the rate set, (extended) capabilities of QSTA-2, and the MAC addresses of QSTA-1 and QSTA-2.

d) The QAP forwards the DLS Response frame to QSTA-1 (step 2b in Figure 68i), after which the direct link becomes active and frames can be sent from QSTA-1 to QSTA-2 and from QSTA-2 to QSTA-1.

If one of the QSTAs roams to a different QAP after a DLS is set up, then there are two possibilities:

— There is an implicit teardown (see 11.7.4).

— The QSTAs continue to be able to communicate, subject to the acceptable level of security between the QSTAs.

Note in this case the DLS cannot be torn down because a teardown message cannot be sent because the QSTAs are not on the same QAP.

11.7.1 DLS

The DLS message flow is illustrated in Figure 68j.

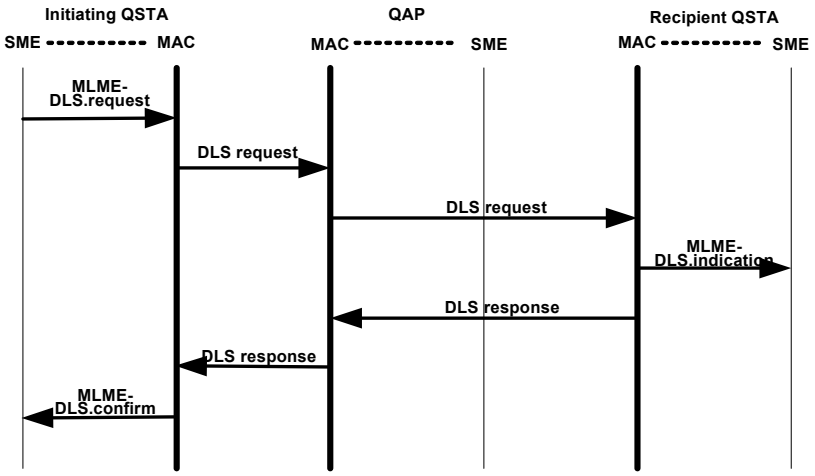


Figure 68j—DLS message flow

11.7.1.1 Setup procedure at the QSTA

All QSTAs shall maintain a list of QSTAs with which a direct link has been established.

Upon receipt of an MLME-DLS.request primitive from the SME, the QSTA shall do one of the following actions:

— Issue an MLME-DLS.confirm primitive with a result code of SUCCESS if the peer MAC address of MLME-DLS.request primitive is in the list of QSTAs with which direct link has been established; or

— Initiate the setup of the direct link by sending the DLS Request frame to the AP (step 1a in Figure 68i). If the QSTA does not receive a DLS Response frame within DLSResponseTimeout after sending the DLS Request frame, the QSTA shall issue an MLME-DLS.confirm primitive with the result code of TIMEOUT.

Upon receipt of the DLS Request frame from the QAP (step 1b in Figure 68i), the QSTA MAC shall send to the AP a DLS Response frame (step 2a in Figure 68i) with a result code of

— SUCCESS if the QSTA is willing to participate in the direct link with the source QSTA. The QSTA shall also issue an MLME-DLS.indication primitive to the SME and shall add the source QSTA to the list of the QSTAs, if not already present, with which direct link has been established.

— REFUSED if the QSTA is not willing to participate in the direct link.

Upon receipt of the DLS Response frame from the QAP (step 2b in Figure 68i), the QSTA shall issue an MLME-DLS.confirm primitive.

— If the result code is SUCCESS, the direct link is considered to be established with the destination QSTA in the DLS Response frame, and the QSTA MAC shall add the destination QSTA to the list of QSTAs with which direct link has been established.

— If the result code is REFUSED, the direct links is not considered to have been established.

11.7.1.2 Setup procedure at the QAP

Upon receipt of the DLS Request frame (step 1a in Figure 68i), the QAP shall

— Send DLS Response frame to the QSTA that sent the DLS Request frame with a result code of Not Allowed in the BSS, if direct links are not allowed in the QBSS (step 2b in Figure 68i).

— Send DLS Response frame to the QSTA that sent the DLS Request frame with a result code of Not Present, if the destination QSTA is not present in the QBSS (step 2b in Figure 68i).

— Send DLS Response frame to the QSTA that sent the DLS Request frame with a result code of Not a QSTA, if the destination STA does not have QoS facility (step 2b in Figure 68i).

— Send the DLS Request frame, with all fields having the same value as the DLS Request frame received by the AP, to the destination QSTA (step 1b in Figure 68i).

Upon receipt of the DLS Response frame from a QSTA (step 2a in Figure 68i), the QAP shall send DLS Response frame to the source QSTA (step 2b in Figure 68i).

The mapping of Status Code field values to ResultCode parameter values in an MLME-DLS.confirm primitive is defined in Table 23f.

Table 23f—Mapping of Status Code field value to ResultCode

ResultCode	Status Code field
SUCCESS	0
REFUSED	37
INVALID_PARAMETERS	38
NOT_ALLOWED	48
NOT_PRESENT	49
NOT_QSTA	50

11.7.2 Data transfer after setup

Both the QSTAs may use direct link for data transfers using any of the access mechanisms defined in this amendment. QSTAs may also set up Block Ack if needed. If needed, the QSTAs may set up TSs with the HC to ensure they have enough bandwidth or use polled TXOPs for data transfer. A protective mechanism (such as transmitting using HCCA, RTS/CTS, or the mechanism described in 9.13) should be used to reduce the probability of other STAs interfering with the direct-link transmissions.

11.7.3 DLS teardown

The DLS teardown message flow is illustrated in Figure 68k.

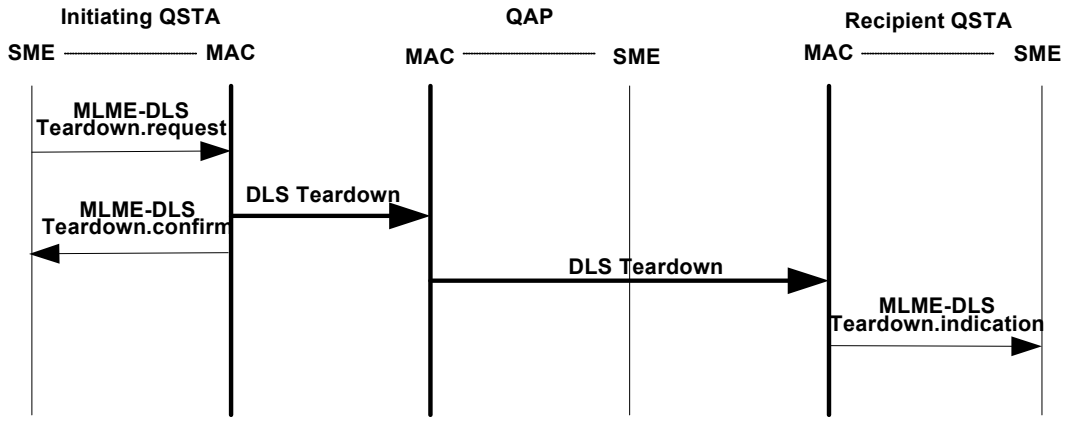


Figure 68k—DLS teardown message flow

11.7.3.1 Teardown procedure at the QSTA

Upon receipt of MLME-DLSTeardown.request primitive from the SME, the QSTA shall initiate the teardown of the direct link by sending the DLS Teardown frame to the QAP. The applicable values of ReasonCode and their encoding to the Reason Code field (see 7.3.1.7) values are defined in Table 23g.

Table 23g—Encoding of ReasonCode to Reason Code field value for DLS teardown

ReasonCode	Reason Code field	Applicable at
QSTA_LEAVING	36	Non-AP QSTA
END_DLS	37	Non-AP QSTA
UNKNOWN_DLS	38	Non-AP QSTA
TIMEOUT	39	Non-AP QSTA
STAKEY_MISMATCH	45	AP

If the transmission of the frame is successful, it shall issue an MLME-DLSTeardown.confirm primitive with a result code of SUCCESS. If the frame could not be transmitted, it shall issue an MLME-DLSTeardown.confirm primitive with a result code of FAILURE.

Upon receipt of the DLS Teardown frame (from the QAP), the QSTA shall issue an MLME-DLSTeardown.indication primitive to the SME and shall delete the QSTA from the list of the QSTAs with which direct link has been established.

11.7.3.2 Teardown procedure at the QAP

Upon receipt of the DLS Teardown frame from a QSTA, the QAP shall send a DLS Teardown frame to the destination QSTA.

Upon receipt of MLME-DLSTeardown.request primitive from the SME, the QAP shall announce the tearing down of the direct link by sending the DLS Teardown frame to the two QSTAs using the direct link. The

only applicable value of the ReasonCode is STAKEY_MISMATCH and its encoding to the Reason Code field value is defined in Table 23g.

If the transmission of the frame is successful, it shall issue an MLME-DLSTeardown.confirm primitive with a result code of SUCCESS. If the frame could not be transmitted, it shall issue an MLME-DLSTeardown.confirm primitive with a result code of FAILURE.

11.7.4 Error recovery upon a peer failure

Every STA shall maintain an inactivity timer for every negotiated direct link (i.e., STAs on both sides of the link maintain these timers). The DLS inactivity timer shall be reset for every successful frame transmission or reception for that direct link. The direct link becomes inactive when no frames have been exchanged as part of the direct link for the duration of DLS timeout value, if the DLS Timeout Value field is set to a non-zero value during the DLS. When the direct link becomes inactive due to the timeout, the MAC issues an MLME-DLSTeardown.indication primitive to the SME and sends a DLS Teardown frame to the QAP, with the peer MAC address as the destination MAC address and the reason code set to TIMEOUT. All frames shall henceforth be sent via the QAP. The procedure is illustrated in Figure 68l.

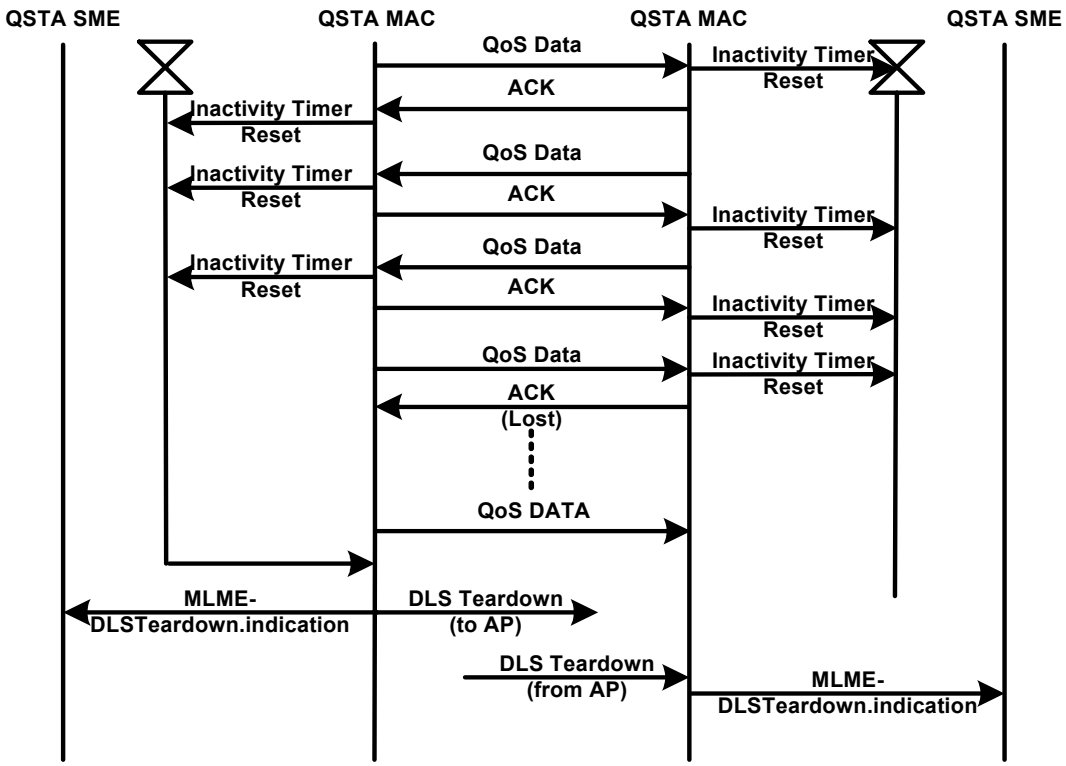


Figure 68l—Error recovery upon a peer failure

If there has been a TS setup for the data transfer, it is the responsibility of the QSTAs to renegotiate the parameters of the TSPEC with the HC.

When two QSTAs have set up a direct link, either QSTA may send DLS Request frames in order to update the DLS timeout value. If the updated DLS Request frame is accepted, both QSTAs initialize the timer to detect DLS timeout. Even if the updated DLS Request frame is not accepted, the original direct link remains active.

11.7.5 Secure DLS operation

STAKey EAPOL-Key frames, defined in 8.5.2, are used to establish the keys needed to enable secure DLS operation. The STAKey message exchange, described in 8.5.2.1, shall be commenced after the DLS establishment and completed prior to initiation of the DLS data frame exchange.

If the QAP does not receive STAKey Message 2 from a STA, it shall attempt dotRSNAConfigPairwiseUpdateCount transmits of STAKey Message 1, plus a final timeout. If it still has not received a response after these retries, then the initiating QAP shall send a DLS teardown message to the applicable STA. 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.

The initiating STA shall wait for the QAP to complete both timeouts before it may retransmit the STAKey request message. The initiating STA may send a DLS teardown message to the QAP at any time. The DLS STAKeys shall be deleted when the DLS teardown message is sent or received.

Change the following subclause numbers due to the insertion above of the new subclauses, 11.4 through 11.7.5:

11.8 ~~11.4~~ Association, reassociation, and disassociation

11.8.1 ~~11.4.1~~ STA association procedures

11.8.2 ~~11.4.2~~ AP association procedures

11.8.3 ~~11.4.3~~ STA reassociation procedures

11.8.4 ~~11.4.4~~ AP reassociation procedures

11.8.5 ~~11.4.5~~ STA disassociation procedures

11.8.6 ~~11.4.6~~ AP disassociation procedures

11.9 ~~11.5~~ TPC procedures

11.9.1 ~~11.5.1~~ Association based on transmit power capability

11.9.2 ~~11.5.2~~ Specification of regulatory and local maximum transmit power levels

11.9.3 ~~11.5.3~~ Selection of a transmit power

11.9.4 ~~11.5.4~~ Adaptation of the transmit power

11.10 ~~11.6~~ DFS procedures

11.10.1 ~~11.6.1~~ Association based on supported channels

11.10.2 ~~11.6.2~~ Quieting channels for testing

11.10.3 ~~11.6.3~~ Testing channels for radars

11.10.4 ~~11.6.4~~ Discontinuing operations after detecting radars

11.10.5 ~~11.6.5~~ Detecting radars

11.10.6 ~~11.6.6~~ Requesting and reporting of measurements

11.10.7 ~~11.6.7~~ Selecting and advertising a new channel

11.10.7.1 ~~11.6.7.1~~ Selecting and advertising a new channel in an infrastructure BSS

11.10.7.2 ~~11.6.7.2~~ Selecting and advertising a new channel in an IBSS

End of changes to Clause 11.

Annex A

(normative)

Protocol Implementation Conformance Statement (PICS) proforma

A.4 PICS proforma

Insert the following at the end of A.4.3

A.4.3 Implementation under test (IUT) configuration

Item	IUT configuration	References	Status	Support
CF12	Quality of service (QoS) supported	9.9, 9.10	O	Yes ☐ No ☐

Insert after A.4.13 the following new subclauses (A.4.14 through A.4.16):

A.4.14 QoS base functionality

Item	Protocol capability	References	Status	Support
QB1	QoS frame format	7.2.1.1 - 7.2.1.3, 7.2.2, 7.2.3.1, 7.2.3.4 - 7.2.3.7, 7.2.3.9, 7.2.3.12,	CF12:M	Yes ☐ No ☐ N/A ☐
QB2	Per traffic identifier (TID) duplicate detection	7.1.3.4, 7.1.3.5, 9.2.9	CF12:M	Yes ☐ No ☐ N/A ☐
QB3	Decode of no-acknowledgment policy in QoS data frames	7.1.3.5.3, 9.9.1.4, 9.9.1.5, 9.9.3.1, 9.9.3.2	CF12:M	Yes ☐ No ☐ N/A ☐
QB4	Block Acknowledgments (Block Acks)	7.2.1.7, 7.2.1.8, 7.4.4, 9.10, 11.5	CF12:O	Yes ☐ No ☐ N/A ☐
QB5	Automatic power-save delivery (APSD)	7.4.2, 11.2.3	CF12:O	Yes ☐ No ☐ N/A ☐
QB6	Direct-link setup (DLS)	7.3.2.20, 7.4.3, 10.3.12, 11.7	(CF1 AND CF12):M (CF2 AND CF12):O	Yes ☐ No ☐ N/A ☐

A.4.15 QoS enhanced distributed channel access (EDCA)

Item	Protocol capability	References	Status	Support
QD1	Support for four transmit queues with a separate channel access entity associated with each	9.1.3.1, 9.9.1.1	CF12:M	Yes ☐ No ☐ N/A ☐
QD2	Per-channel access function differentiated channel access	9.9.1.2, 9.9.1.3, 9.9.1.5	CF12:M	Yes ☐ No ☐ N/A ☐

Item	Protocol capability	References	Status	Support
QD3	Multiple frame transmission support	9.9.1.4	CF12:O	Yes ☐ No ☐ N/A ☐
QD4	Maintenance of within-queue ordering, exhaustive retransmission when sending non-QoS data frames	9.9.1.6	CF12:M	Yes ☐ No ☐ N/A ☐
QD5	Interpretation of admission control mandatory (ACM) bit in EDCA Parameter Set element	7.3.2.14, 9.9.3.1	(CF2 & CF12):M	Yes ☐ No ☐ N/A ☐
QD6	Contention-based admission control	9.9.3.1, 7.2.3,15, 7.2.3.16, 7.4.2.1 - 7.4.2.3, 11.4	(CF1 & CF12):O (CF2 & CF12):O	Yes ☐ No ☐ N/A ☐
QD7	Power management	11.2	CF12:O	Yes ☐ No ☐ N/A ☐

A.4.16 QoS hybrid coordination function (HCF) controlled channel access (HCCA)

Item	Protocol Capability	References	Status	Support
QP1	Traffic specification (TSPEC) and associated frame formats	7.4.2	CF12:M	Yes ☐ No ☐ N/A ☐
QP2	HCCA rules	9.1.3.2, 9.9.2, 9.9.2.1 - 9.9.2.3	CF12:M	Yes ☐ No ☐ N/A ☐
QP3	HCCA schedule generation and management	9.9.3	(CF1 & CF12):M	Yes ☐ No ☐ N/A ☐
QP4	HCF frame exchange sequences	9.12	CF12:M	Yes ☐ No ☐ N/A ☐
QP5	Traffic stream (TS) management	11.4	CF12:M	Yes ☐ No ☐ N/A ☐
QP6	Minimum TSPEC parameter set	9.9.3	CF12:M	Yes ☐ No ☐ N/A ☐
QP7	Power management	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	CF12:M	Yes ☐ No ☐ N/A ☐

Annex C

(normative)

Formal description of MAC operation

C.3 State machines for MAC

C.4 State machines for MAC AP

Insert the following text at the end of the text portions introducing C.3 and C.4:

This clause does not describe the behavior of a STA with QoS facility.

Annex D

(informative)

ASN.1 encoding of the MAC and PHY MIB

In "Major sections" of Annex D, change the list of MAC attributes as shown:

```
        --  dot11OperationTable             ::= { dot11mac 1 }
        --  dot11CountersTable              ::= { dot11mac 2 }
        --  dot11GroupAddressesTable        ::= { dot11mac 3 }
        --  dot11EDCATable                  ::= { dot11mac 4 }
        --  dot11QAPEDCATable               ::= { dot11mac 5 }
        --  dot11QosCountersTable           ::= { dot11mac 6 }
```

In "SMT Station Config Table" of Annex D, change Dot11StationConfigEntry as shown:

```
Dot11StationConfigEntry ::=
        SEQUENCE {
            dot11StationID                          MacAddress,
            dot11MediumOccupancyLimit               INTEGER,
            dot11CFPollable                         TruthValue,
            dot11CFPPeriod                          INTEGER,
            dot11CFPMaxDuration                     INTEGER,
            dot11AuthenticationResponseTimeOut      Unsigned32,
            dot11PrivacyOptionImplemented           TruthValue,
            dot11PowerManagementMode                INTEGER,
            dot11DesiredSSID                        OCTET STRING,
            dot11DesiredBSSType                     INTEGER,
            dot11OperationalRateSet                 OCTET STRING,
            dot11BeaconPeriod                       INTEGER,
            dot11DTIMPeriod                         INTEGER,
            dot11AssociationResponseTimeOut         Unsigned32,
            dot11DisassociateReason                 INTEGER,
            dot11DisassociateStation                MacAddress,
            dot11DeauthenticateReason               INTEGER,
            dot11DeauthenticateStation              MacAddress,
            dot11AuthenticateFailStatus             INTEGER,
            dot11AuthenticateFailStation            MacAddress,
            dot11MultiDomainCapabilityImplemented   TruthValue,
            dot11MultiDomainCapabilityEnabled       TruthValue,
            dot11CountryString                      OCTET STRING,
            dot11SpectrumManagementImplemented      TruthValue,
            dot11SpectrumManagementRequired         Truthvalue,
            dot11RSNAOptionImplemented              TruthValue,
            dot11RSNAPreauthenticationImplemented   TruthValue,
            dot11RegulatoryClassesImplemented       TruthValue,
            dot11RegulatoryClassesRequired          TruthValue,
            dot11QosOptionImplemented               TruthValue,
            dot11ImmediateBlockAckOptionImplemented TruthValue,
            dot11DelayedBlockAckOptionImplemented   TruthValue,
            dot11DirectOptionImplemented            TruthValue,
            dot11APSDOptionImplemented              TruthValue,
            dot11QAckOptionImplemented              TruthValue,
            dot11QBSSLoadOptionImplemented          TruthValue,
            dot11QueueRequestOptionImplemented      TruthValue,
```

```
        dot11TXOPRequestOptionImplemented       TruthValue,
        dot11MoreDataAckOptionImplemented       TruthValue,
        dot11AssociateinNQBSS                   TruthValue,
        dot11DLSAllowedInQBSS                   TruthValue,
        dot11DLSAllowed                         TruthValue }
```

Insert the following elements at the end of dot11StationConfigEntry element definitions after dot11RegulatoryClassesRequired { dot11StationConfigEntry 29 }:

```
dot11QosOptionImplemented OBJECT-TYPE
             SYNTAX TruthValue
             MAX-ACCESS read-only
             STATUS current
             DESCRIPTION
                    "This attribute, when TRUE, indicates that the station
                    implementation is capable of supporting QoS. The capability
                    is disabled, otherwise. The default value of this attribute
                    is FALSE."
      ::= { dot11StationConfigEntry 30}

dot11ImmediateBlockAckOptionImplemented OBJECT-TYPE
             SYNTAX TruthValue
             MAX-ACCESS read-only
             STATUS current
             DESCRIPTION
                    "This attribute, when TRUE, indicates that the station
                    implementation is capable of supporting Immediate Block Ack.
                    The capability is disabled, otherwise. The default value of
                    this attribute is FALSE."
      ::= { dot11StationConfigEntry 31}

dot11DelayedBlockAckOptionImplemented OBJECT-TYPE
             SYNTAX TruthValue
             MAX-ACCESS read-only
             STATUS current
             DESCRIPTION
                    "This attribute, when TRUE, indicates that the station
                    implementation is capable of supporting Delayed Block Ack.
                    The capability is disabled, otherwise. The default value of
                    this attribute is FALSE."
      ::= { dot11StationConfigEntry 32 }

dot11DirectOptionImplemented OBJECT-TYPE
             SYNTAX TruthValue
             MAX-ACCESS read-only
             STATUS current
             DESCRIPTION
                    "This attribute, when TRUE, indicates that the station
                    implementation is capable of sending and receiving frames
                    from a non-AP QSTA in the QBSS. The capability is disabled,
                    otherwise. The default value of this attribute is FALSE."
      ::= { dot11StationConfigEntry 33 }

dot11APSDOptionImplemented OBJECT-TYPE
             SYNTAX TruthValue
             MAX-ACCESS read-only
             STATUS current
             DESCRIPTION
```



```
                    "This attribute is available only at a QAP. When TRUE,
                    this attribute indicates that the QAP implementation is
                    capable of delivering data and polls to stations using
                    APSD."
       ::= { dot11StationConfigEntry 34 }

dot11QAckOptionImplemented OBJECT-TYPE
              SYNTAX TruthValue
              MAX-ACCESS read-only
              STATUS current
              DESCRIPTION
                    "This attribute, when TRUE, indicates that the station
                    implementation is capable of interpreting the CF-Ack bit
                    in a received frame of type data even if the frame is not
                    directed to the QoS station. The capability is disabled,
                    otherwise. A QSTA is capable of interpreting the CF-Ack bit
                    in a received data frame if that station is the recipient
                    of the data frame, regardless of the value of this MIB
                    variable. The default value of this attribute is FALSE."
       ::= { dot11StationConfigEntry 35 }

dot11QBSSLoadOptionImplemented OBJECT-TYPE
              SYNTAX TruthValue
              MAX-ACCESS read-only
              STATUS current
              DESCRIPTION
                    "This attribute is available only at a QAP. This attribute,
                    when TRUE, indicates that the QAP implementation is capable
                    of generating and transmitting the QBSS load element in the
                    Beacon and Probe Response frames. The capability is
                    disabled, otherwise. The default value of this attribute is
                    FALSE."
       ::= { dot11StationConfigEntry 36 }

dot11QueueRequestOptionImplemented OBJECT-TYPE
              SYNTAX TruthValue
              MAX-ACCESS read-only
              STATUS current
              DESCRIPTION
                    "This attribute is available only at a QAP. This attribute,
                    when TRUE, indicates that the QAP is capable of processing
                    the Queue Size field in QoS Control field of QoS Data type
                    frames. The capability is disabled, otherwise. The default
                    value of this attribute is FALSE."
       ::= { dot11StationConfigEntry 37 }

dot11TXOPRequestOptionImplemented OBJECT-TYPE
              SYNTAX TruthValue
              MAX-ACCESS read-only
              STATUS current
              DESCRIPTION
                    "This attribute is available only at a QAP. This attribute,
                    when TRUE, indicates that the QAP is capable of processing
                    the TXOP Duration requested field in QoS Control field of
                    QoS Data type frames. The capability is disabled, otherwise.
                    The default value of this attribute is FALSE."
       ::= { dot11StationConfigEntry 38 }

dot11MoreDataAckOptionImplemented OBJECT-TYPE
```

```
              SYNTAX TruthValue
              MAX-ACCESS read-only
              STATUS current
              DESCRIPTION
                     "This attribute, when TRUE, indicates that the station
                     implementation is capable of interpreting the More Data
                     bit in the ACK frames. The capability is disabled,
                     otherwise. The default value of this attribute is FALSE."
       ::= { dot11StationConfigEntry 39 }

dot11AssociateinNQBSS OBJECT-TYPE
              SYNTAX TruthValue
              MAX-ACCESS read-write
              STATUS current
              DESCRIPTION
                     "This attribute, when TRUE, indicates that the station may
                     associate in an nQBSS. When FALSE, this attribute indicates
                     that the station may only associate in a QBSS. The default
                     value of this attribute is TRUE."
       ::= { dot11StationConfigEntry 40 }

dot11DLSAllowedInQBSS OBJECT-TYPE
              SYNTAX TruthValue
              MAX-ACCESS read-write
              STATUS current
              DESCRIPTION
                     "This attribute available at the QAP, when TRUE, indicates
                     that non-AP QSTAs may set up DLS and communicate with other
                     non-AP QSTAs in the QBSS. When FALSE, this attribute
                     indicates that the stations shall not set up DLS nor
                     communicate with other non-AP QSTAs in the QBSS. The default
                     value of this attribute is TRUE."
       ::= { dot11StationConfigEntry 41 }

dot11DLSAllowed OBJECT-TYPE
              SYNTAX TruthValue
              MAX-ACCESS read-write
              STATUS current
              DESCRIPTION
                     "This attribute available at the non-AP QSTAs, when TRUE,
                     indicates that QSTA may set up DLS or accept DLS Requests,
                     and communicate with other non-AP QSTAs in the QBSS. When
                     FALSE, this attribute indicates that the QSTA shall not set
                     up DLS nor accept DLS, nor communicate with other non-AP
                     QSTAs in the QBSS. The default value of this attribute is
                     TRUE."
       ::= { dot11StationConfigEntry 42}
```

In "dot11OperationTable" of Annex D, change Dot11OperationEntry as shown:

```
Dot11OperationEntry ::=
        SEQUENCE { dot11MACAddress                    MacAddress,
                   dot11RTSThreshold                  INTEGER,
                   dot11ShortRetryLimit               INTEGER,
                   dot11LongRetryLimit                INTEGER,
                   dot11FragmentationThreshold        INTEGER,
                   dot11MaxTransmitMSDULifetime       Unsigned32,
                   dot11MaxReceiveLifetime            Unsigned32,
```



```
                dot11ManufacturerID                 DisplayString,
                dot11ProductID                      DisplayString,
                dot11CAPLimit                       INTEGER,
                dot11HCCWmin                        INTEGER,
                dot11HCCWmax                        INTEGER,
                dot11HCCAIFSN                       INTEGER,
                dot11ADDBAResponseTimeout           INTEGER,
                dot11ADDTSResponseTimeout           INTEGER,
                dot11ChannelUtilizationBeaconIntervalINTEGER,
                dot11ScheduleTimeout                INTEGER,
                dot11DLSResponseTimeout             INTEGER,
                dot11QAPMissingAckRetryLimit        INTEGER,
                dot11EDCAAveragingPeriod            INTEGER }
```

Change the last sentence of DESCRIPTION under dot11RTSThreshold as shown:

```
                The default value of this attribute shall be 23473000.
```

Change the fourth sentence of DESCRIPTION under dot11FragmentationThreshold as shown:

```
                The default value for this attribute shall be the lesser
                of 23463000 or the aMPDUMaxLength of the attached PHY and
                shall never exceed the lesser of 23463000 or the
                aMPDUMaxLength of the attached PHY.
```

Insert the following elements at the end of dot11OperationEntry element definitions after dot11ProductID { dot11OperationEntry 9 }:

```
dot11CAPLimit OBJECT-TYPE
        SYNTAX INTEGER
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
             "This attribute shall specify the maximum number of TUs a
             Controlled access phase(CAP) can last."
    ::= { dot11OperationEntry 10 }

dot11HCCWmin OBJECT-TYPE
        SYNTAX INTEGER (0..aCWmin)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
             "This attribute shall specify the value of the minimum size of the
             window that shall be used by the HC 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 2^X - 1, where X is an
             integer. The default value of this attribute shall be 0."
    ::= { dot11OperationEntry 11 }

dot11HCCWmax OBJECT-TYPE
        SYNTAX INTEGER (0..aCWmax)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
             "This attribute shall specify the value of the maximum size of the
             window that shall be used by the HC 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 2^X - 1, where X is an
```

```
              integer. The default value of this attribute shall be 0."
    ::= { dot11OperationEntry 12 }

dot11HCCAIFSN OBJECT-TYPE
        SYNTAX INTEGER (1..15)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
              "This attribute shall specify the number of slots, after a SIFS
              duration, that the HC shall sense the medium idle either before
              transmitting or executing a backoff. The default value of this
              attribute shall be 1."
    ::= { dot11OperationEntry 13 }

dot11ADDBAResponseTimeout OBJECT-TYPE
        SYNTAX INTEGER (1..65535)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
              "This attribute shall specify the maximum number of seconds a BA is
              to be responded. The default value of this attribute shall be 1."
    ::= { dot11OperationEntry 14 }

dot11ADDTSResponseTimeout OBJECT-TYPE
        SYNTAX INTEGER (1..65535)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
              "This attribute shall specify the maximum number of seconds an
              ADDTS request is to be responded. The default value of this
              attribute shall be 1."
    ::= { dot11OperationEntry 15 }

dot11ChannelUtilizationBeaconInterval OBJECT-TYPE
        SYNTAX INTEGER (1..100)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
              "This attribute shall indicate the number of beacon intervals over
              which the channel busy time should be averaged. The default value
              for this parameter shall be 50."
    ::= { dot11OperationEntry 16 }

dot11ScheduleTimeout OBJECT-TYPE
        SYNTAX INTEGER (1..100)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
              "This attribute shall indicate the duration in TUs after which a
              STA could go into power-save mode. The default value for this
       parameter shall be 10."
    ::= { dot11OperationEntry 17 }

dot11DLSResponseTimeout OBJECT-TYPE
        SYNTAX INTEGER (1..65535)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
              "This attribute shall specify the maximum number of seconds a
```



```
            direct link request is to be responded. The default value of this
            attribute shall be 10."
    ::= { dot11OperationEntry 18 }

dot11QAPMissingAckRetryLimit OBJECT-TYPE
        SYNTAX INTEGER (1..100)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "This attribute indicates the number of times the QAP may retry a
            frame for which it does not receive an ACK for a STA in power-save
            mode after receiving a PS-Poll and sending a directed response or
            after the QAP does not receive an ACK to a directed MPDU sent with
            the EOSP set to 1."
    ::= { dot11OperationEntry 19 }

dot11EDCAAveragingPeriod OBJECT-TYPE
        SYNTAX INTEGER (1..100)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "This attribute shall indicate the number of seconds over which
            the admitted_time and used_time are computed. The default value
            for this parameter shall be 5."
    ::= { dot11OperationEntry 20 }
```

In "dot11Counters Table" of Annex D, change Dot11CountersEntry as shown:

```
Dot11CountersEntry ::=
        SEQUENCE { dot11TransmittedFragmentCount       Counter32,
                   dot11MulticastTransmittedFrameCount Counter32,
                   dot11FailedCount                    Counter32,
                   dot11RetryCount                     Counter32,
                   dot11MultipleRetryCount             Counter32,
                   dot11FrameDuplicateCount            Counter32,
                   dot11RTSSuccessCount                Counter32,
                   dot11RTSFailureCount                Counter32,
                   dot11ACKFailureCount                Counter32,
                   dot11ReceivedFragmentCount          Counter32,
                   dot11MulticastReceivedFrameCount    Counter32,
                   dot11FCSErrorCount                  Counter32,
                   dot11TransmittedFrameCount          Counter32,
                   dot11WEPUndecryptableCount          Counter32,
                   dot11QosDiscardedFragmentCount      Counter32,
                   dot11AssociatedStationCount         Counter32,
                   dot11QosCFPollsReceivedCount        Counter32,
                   dot11QosCFPollsUnusedCount          Counter32,
                   dot11QosCFPollsUnusableCount        Counter32 }
```

Insert the following elements at the end of Dot11CountersEntry element definitions after dot11WEPUndecryptableCount { dot11CountersEntry 14 }:

```
dot11QosDiscardedFragmentCount OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
              "This counter shall increment for each QoS Data MPDU that has been
```

```
              discarded."
    ::= { dot11CountersEntry 15 }

dot11AssociatedStationCount OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
               "This counter, only available at QAP, shall increment when a
              station associates or reassociates. This counter shall decrement
              when a station disassociates."
    ::= { dot11CountersEntry 16 }

dot11QosCFPollsReceivedCount OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
               "This counter shall increment for each QoS (+)CF-Poll that has been
              received."
    ::= { dot11CountersEntry 17 }

dot11QosCFPollsUnusedCount OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
               "This counter shall increment for each QoS (+)CF-Poll that has been
              received but not used."
    ::= { dot11CountersEntry 18 }

dot11QosCFPollsUnusableCount OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
               "This counter shall increment for each QoS (+)CF-Poll that has been
              received but could not be used due to the TXOP size being smaller
              than the time that is required for one frame exchange sequence."
    ::= { dot11CountersEntry 19 }
```

After the end of the "Group Addresses TABLE" in Annex D, insert the following new tables (SMT EDCA Config TABLE, SMT QAP EDCA Config TABLE, and dot11QosCounters TABLE):

```
-- ************************************************************************
-- *    SMT EDCA Config TABLE
-- ************************************************************************

dot11EDCATable OBJECT-TYPE
        SYNTAX SEQUENCE OF Dot11EDCATableEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
              "Conceptual table for EDCA default parameter values at a non-AP
              QSTA. This table shall contain the four entries of the EDCA
              parameters corresponding to four possible ACs. Index 1 corresponds
              to AC_BK, index 2 to AC_BE, index 3 to AC_VI, and index 4 to AC_VO."
       REFERENCE
```



```
              "IEEE 802.11 Tge Draft Version, 9.1.3.1"
    ::= { dot11mac 4 }

dot11EDCATableEntry OBJECT-TYPE
        SYNTAX Dot11EDCATableEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
          "An Entry (conceptual row) in the EDCA Table.

          ifIndex - Each IEEE 802.11 interface is represented by an ifEntry.
          Interface tables in this MIB module are indexed by ifIndex."
        INDEX { ifIndex,
                dot11EDCATableIndex }
    ::= { dot11EDCATable  1 }

Dot11EDCATableEntry ::=
        SEQUENCE { dot11EDCATableIndex               INTEGER,
                   dot11EDCATableCWmin               INTEGER,
                   dot11EDCATableCWmax               INTEGER,
                   dot11EDCATableAIFSN               INTEGER,
                   dot11EDCATableTXOPLimit           INTEGER,
                   dot11EDCATableMSDULifetime        INTEGER,
                   dot11EDCATableMandatory           TruthValue }

dot11EDCATableIndex OBJECT-TYPE
        SYNTAX INTEGER (1..4)
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "The auxiliary variable used to identify instances of the columnar
             objects in the EDCA Table. The value of this variable is
             1)     1, if the value of the AC is AC_BK.
             2)     2, if the value of the AC is AC_BE.
             3)     3, if the value of the AC is AC_VI.
             4)     4, if the value of the AC is AC_VO."
    ::= { dot11EDCATableEntry 1 }

dot11EDCATableCWmin OBJECT-TYPE
        SYNTAX INTEGER (0..255)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
             "This attribute shall specify the value of the minimum size of the
             window that shall be used by a QSTA for a particular AC 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 2^X - 1, where X is an integer. The default value for this
             attribute is
             1)     aCWmin, if dot11EDCATableIndex is 1 or 2.
             2)     (aCWmin+1)/2 - 1, if dot11EDCATableIndex is 3.
             3)     (aCWmin+1)/4 - 1, if dot11EDCATableIndex is 4."
    ::= { dot11EDCATableEntry 2 }

dot11EDCATableCWmax OBJECT-TYPE
        SYNTAX INTEGER (0..65535)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
```

```
            "This attribute shall specify the value of the maximum size of the
            window that shall be used by a QSTA for a particular AC 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 2^X - 1, where X is an integer. The default value for this
            attribute is
            1)     aCWmax, if dot11EDCATableIndex is 1 or 2.
            2)     aCWmin, if dot11EDCATableIndex is 3.
            3)     (aCWmin+1)/2 - 1, if dot11EDCATableIndex is 4."
    ::= { dot11EDCATableEntry 3 }

dot11EDCATableAIFSN OBJECT-TYPE
        SYNTAX INTEGER (2..15)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "This attribute shall specify the number of slots, after a SIFS
            duration, that the QSTA, for a particular AC, shall sense the
            medium idle either before transmitting or executing a backoff. The
            default value for this attribute is
            1)     7, if dot11EDCATableIndex is 1,
            2)     3, if dot11EDCATableIndex is 2
            3)     2, otherwise."
    ::= { dot11EDCATableEntry 4 }

dot11EDCATableTXOPLimit OBJECT-TYPE
        SYNTAX INTEGER (0..65535)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "This attribute shall specify the maximum number of microseconds of
            an EDCA TXOP for a given AC at a non-AP QSTA. The default value for
            this attribute is
            1)     0 for all PHYs, if dot11EDCATableIndex is 1 or 2; this
            implies that the sender can send one MSDU in an EDCA TXOP,
            2)     3008 microseconds for IEEE 802.11a/g PHY and 6016
            microseconds for IEEE 802.11b PHY, if dot11EDCATableIndex is 3,
            3)     1504 microseconds for IEEE 802.11a/g PHY and 3264
            microseconds for IEEE 802.11b PHY, if dot11EDCATableIndex is 4."
    ::= { dot11EDCATableEntry 5 }

dot11EDCATableMSDULifetime OBJECT-TYPE
        SYNTAX INTEGER (0..500)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "This attribute shall specify (in TUs) the maximum duration an
            MSDU, for a given AC, would be retained by the MAC before it is
            discarded. The default value for this parameter shall be 500."
    ::= { dot11EDCATableEntry 6 }

dot11EDCATableMandatory OBJECT-TYPE
        SYNTAX TruthValue
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
           "This attribute, when TRUE, indicates that admission control is
            mandatory for the given AC. When False, this attribute indicates
            that the admission control is not mandatory for the given AC. The
```



```
             default value for this parameter shall be FALSE."
    ::= { dot11EDCATableEntry 7 }

-- ************************************************************************
-- *    End of SMT EDCA Config TABLE
-- ************************************************************************

-- ************************************************************************
-- *    SMT QAP EDCA Config TABLE
-- ************************************************************************

dot11QAPEDCATable OBJECT-TYPE
        SYNTAX SEQUENCE OF Dot11QAPEDCATableEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
             "Conceptual table for EDCA default parameter values at the QAP.
             This table shall contain the four entries of the EDCA parameters
             corresponding to four possible ACs. Index 1 corresponds to AC_BK,
             index 2 to AC_BE, index 3 to AC_VI, and index 4 to AC_VO."
       REFERENCE
             "IEEE 802.11 Tge Draft Version, 9.9.1"
    ::= { dot11mac 5 }

dot11QAPEDCATableEntry OBJECT-TYPE
        SYNTAX Dot11QAPEDCATableEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "An Entry (conceptual row) in the EDCA Table.

            ifIndex - Each IEEE 802.11 interface is represented by an
            ifEntry. Interface tables in this MIB module are indexed
            by ifIndex."
        INDEX { ifIndex,
                dot11QAPEDCATableIndex }
    ::= { dot11QAPEDCATable  1 }

Dot11QAPEDCATableEntry ::=
        SEQUENCE { dot11QAPEDCATableIndex            INTEGER,
                   dot11QAPEDCATableCWmin            INTEGER,
                   dot11QAPEDCATableCWmax            INTEGER,
                   dot11QAPEDCATableAIFSN            INTEGER,
                   dot11QAPEDCATableTXOPLimit        INTEGER,
                   dot11QAPEDCATableMSDULifetime     INTEGER,
                   dot11QAPEDCATableMandatory        TruthValue }

dot11QAPEDCATableIndex OBJECT-TYPE
        SYNTAX INTEGER (1..4)
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
             "The auxiliary variable used to identify instances of the columnar
             objects in the EDCA Table. The value of this variable is
             1)     1, if the value of the AC is AC_BK.
             2)     2, if the value of the AC is AC_BE.
             3)     3, if the value of the AC is AC_VI.
             4)     4, if the value of the AC is AC_VO."
    ::= { dot11QAPEDCATableEntry 1 }
```

```
dot11QAPEDCATableCWmin OBJECT-TYPE
        SYNTAX INTEGER (0..255)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
             "This attribute shall specify the value of the minimum size of the
             window that shall be used by a QAP for a particular AC 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 2^X - 1, where X is an integer. The default value for this
             attribute is
             1)     aCWmin, if dot11QAPEDCATableIndex is 1 or 2.
             2)     (aCWmin+1)/2 - 1, if dot11QAPEDCATableIndex is 3.
             3)     (aCWmin+1)/4 - 1, if dot11QAPEDCATableIndex is 4."
    ::= { dot11QAPEDCATableEntry 2 }

dot11QAPEDCATableCWmax OBJECT-TYPE
        SYNTAX INTEGER (0..65535)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
             "This attribute shall specify the value of the maximum size of the
             window that shall be used by a QAP for a particular AC 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 2^X - 1, where X is an integer. The default value for this
             attribute is
             1)     aCWmax, if dot11QAPEDCATableIndex is 1.
             2)     4*(aCWmin+1) - 1, if dot11QAPEDCATableIndex is 2.
             3)     aCWmin, if dot11QAPEDCATableIndex is 3.
             4)     (aCWmin+1)/2 - 1, if dot11QAPEDCATableIndex is 4."
    ::= { dot11QAPEDCATableEntry 3 }


dot11QAPEDCATableAIFSN OBJECT-TYPE
        SYNTAX INTEGER (1..15)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
             "This attribute shall specify the number of slots, after a SIFS
             duration, that the QAP, for a particular AC, shall sense the
             medium idle either before transmitting or executing a backoff. The
             default value for this attribute is
             1)     7, if dot11QAPEDCATableIndex is 1,
             2)     3, if dot11QAPEDCATableIndex is 2
             3)     1, otherwise."
    ::= { dot11QAPEDCATableEntry 4 }

dot11QAPEDCATableTXOPLimit OBJECT-TYPE
        SYNTAX INTEGER (0..65535)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
             "This attribute shall specify the maximum number of microseconds of
             an EDCA TXOP for a given AC at the QAP. The default value for
             this attribute is
             1)     0 for all PHYs, if dot11QAPEDCATableIndex is 1 or 2; this
             implies that the sender can send one MSDU in an EDCA TXOP,
```



```
              2)     3008 microseconds for IEEE 802.11a/g PHY and 6016
              microseconds for IEEE 802.11b PHY, if dot11QAPEDCATableIndex is 3,
              3)     1504 microseconds for IEEE 802.11a/g PHY and 3264
              microseconds for IEEE 802.11b PHY, if dot11QAPEDCATableIndex is 4."
    ::= { dot11QAPEDCATableEntry 5 }

dot11QAPEDCATableMSDULifetime OBJECT-TYPE
        SYNTAX INTEGER (0..500)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
              "This attribute shall specify (in TUs) the maximum duration an
              MSDU, for a given AC, would be retained by the MAC at the QAP before
              it is discarded. The default value for this parameter shall be
              500."
    ::= { dot11QAPEDCATableEntry 6 }

dot11QAPEDCATableMandatory OBJECT-TYPE
        SYNTAX TruthValue
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
              "This attribute, when TRUE, indicates that admission control is
              mandatory for the given AC. When False, this attribute indicates
              that the admission control is not mandatory for the given AC. The
              default value for this parameter shall be FALSE."
    ::= { dot11QAPEDCATableEntry 7 }

-- ****************************************************************
-- *    End of SMT QAP EDCA Config TABLE
-- ****************************************************************

-- ****************************************************************
-- *    dot11QosCounters TABLE
-- ****************************************************************

dot11QosCountersTable OBJECT-TYPE
        SYNTAX SEQUENCE OF Dot11QosCountersEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
              "Group containing attributes that are MAC counters implemented as a
              table to allow for multiple instantiations on an agent."
    ::= { dot11mac 6 }

dot11QosCountersEntry OBJECT-TYPE
        SYNTAX Dot11QosCountersEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "An Entry (conceptual row) in the EDCA Table.

            ifIndex - Each IEEE 802.11 interface is represented by an
            ifEntry. Interface tables in this MIB module are indexed
            by ifIndex."
        INDEX { ifIndex,
                dot11QosCountersIndex }
    ::= { dot11QosCountersTable 1 }
```

```
Dot11QosCountersEntry ::=
        SEQUENCE { dot11QosCountersIndex              INTEGER,
                   dot11QosTransmittedFragmentCount   Counter32,
                   dot11QosFailedCount                Counter32,
                   dot11QosRetryCount                 Counter32,
                   dot11QosMultipleRetryCount         Counter32,
                   dot11QosFrameDuplicateCount        Counter32,
                   dot11QosRTSSuccessCount            Counter32,
                   dot11QosRTSFailureCount            Counter32,
                   dot11QosACKFailureCount            Counter32,
                   dot11QosReceivedFragmentCount      Counter32,
                   dot11QosTransmittedFrameCount      Counter32,
                   dot11QosDiscardedFrameCount        Counter32,
                   dot11QosMPDUsReceivedCount         Counter32,
                   dot11QosRetriesReceivedCount       Counter32,}

dot11QosCountersIndex OBJECT-TYPE
        SYNTAX INTEGER (1..16)
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
             "The auxiliary variable used to identify instances of the columnar
             objects in the QoSCounter Table. The value of this variable is
             equal to TID + 1."
    ::= { dot11QosCountersEntry 1 }

dot11QosTransmittedFragmentCount OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
             "This counter shall be incremented for an acknowledged MPDU, for a
             particular UP, with an individual address in the address 1 field or
             an MPDU with a multicast address in the address 1 field, either
             belonging to a particular TID. This counter has relevance only for
             TIDs between 0 and 7."
    ::= { dot11QosCountersEntry 2 }

dot11QosFailedCount OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
             "This counter shall increment when an MSDU, for a particular UP, is
             not transmitted successfully due to the number of transmit attempts
             exceeding either the dot11ShortRetryLimit or dot11LongRetryLimit.
             This counter has relevance only for TIDs between 0 and 7."
    ::= { dot11QosCountersEntry 3 }

dot11QosRetryCount OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
             "This counter shall increment when an MSDU, of a particular UP, is
             successfully transmitted after one or more retransmissions. This
             counter has relevance only for TIDs between 0 and 7."
    ::= { dot11QosCountersEntry 4 }
```



```
dot11QosMultipleRetryCount OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
              "This counter shall increment when an MSDU, of a particular UP, is
              successfully transmitted after more than one retransmissions. This
              counter has relevance only for TIDs between 0 and 7."
    ::= { dot11QosCountersEntry 5 }

dot11QosFrameDuplicateCount OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
              "This counter shall increment when a frame, of a particular UP, is
              received that the Sequence Control field indicates is a duplicate.
              This counter has relevance only for TIDs between 0 and 7."
    ::= { dot11QosCountersEntry 6 }

dot11QosRTSSuccessCount OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
              "This counter shall increment when a CTS is received in response to
              an RTS that has been sent for the transmission of an MPDU of a
              particular UP. This counter has relevance only for TIDs between 0
              and 7."
    ::= { dot11QosCountersEntry 7 }

dot11QosRTSFailureCount OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
              "This counter shall increment when a CTS is not received in
              response to an RTS that has been sent for the transmission of an
              MPDU of a particular UP. This counter has relevance only for TIDs
              between 0 and 7."
    ::= { dot11QosCountersEntry 8 }

dot11QosACKFailureCount OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
              "This counter shall increment when an ACK is not received in
              response to an MPDU of a particular UP. This counter has relevance
              only for TIDs between 0 and 7."
    ::= { dot11QosCountersEntry 9 }

dot11QosReceivedFragmentCount OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
              "This counter shall be incremented for each successfully received
              MPDU of type Data of a particular UP. This counter has relevance
```

```
              only for TIDs between 0 and 7."
    ::= { dot11QosCountersEntry 10 }

dot11QosTransmittedFrameCount OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
              "This counter shall increment for each successfully transmitted
              MSDU of a particular UP. This counter has relevance only for TIDs
              between 0 and 7."
    ::= { dot11QosCountersEntry 11 }

dot11QosDiscardedFrameCount OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
              "This counter shall increment for each Discarded MSDU of a
              particular UP. This counter has relevance only for TIDs between 0
              and 7."
    ::= { dot11QosCountersEntry 12 }

dot11QosMPDUsReceivedCount OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
              "This counter shall increment for each received MPDU of a
              particular TID."
    ::= { dot11QosCountersEntry 13 }

dot11QosRetriesReceivedCount OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
              "This counter shall increment for each received MPDU of a
              particular TID with the retry bit set to 1."
    ::= { dot11QosCountersEntry 14 }

-- ****************************************************************
-- *    End of dot11QosCounters TABLE
-- ****************************************************************
```

In "Compliance Statements" of Annex D, change the dot11Compliance 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 {
              dot11SMTbase4 6,
              dot11MACbase 2, dot11CountersGroup 2,
              dot11SmtAuthenticationAlgorithms,
              dot11ResourceTypeID, dot11PhyOperationComplianceGroup }
```


In "Compliance Statements" of Annex D, change OPTIONAL-GROUPS as follows:

```
-- OPTIONAL-GROUPS { dot11SMTprivacy, dot11MACStatistics,
       -- dot11PhyAntennaComplianceGroup, dot11PhyTxPowerComplianceGroup,
       -- dot11PhyRegDomainsSupportGroup,
       -- dot11PhyAntennasListGroup, dot11PhyRateGroup,
       -- dot11SMTbase3, dot11MultiDomainCapabilityGroup,
       -- dot11PhyFHSSComplianceGroup2, dot11RSNAadditions,
       -- dot11RegulatoryClassesGroup, dot11Qosadditions }
```

In "Groups - units of conformance" of Annex D, change dot11MACbase as follows:

```
dot11MACbase OBJECT-GROUP
       OBJECTS { dot11MACAddress, dot11Address,
              dot11GroupAddressesStatus,
              dot11RTSThreshold, dot11ShortRetryLimit,
              dot11LongRetryLimit, dot11FragmentationThreshold,
              dot11MaxTransmitMSDULifetime,
              dot11MaxReceiveLifetime, dot11ManufacturerID,
              dot11ProductID }
       STATUS ~~current~~deprecated
       DESCRIPTION
              "The MAC object class provides the necessary support for the
              access control, generation, and verification of frame check
              sequences (FCSs), and proper delivery of valid data to upper
              layers."
::= { dot11Groups 3 }
```

In "Groups - units of conformance" of Annex D, change dot11CountersGroup as follows:

```
dot11CountersGroup OBJECT-GROUP
       OBJECTS { dot11TransmittedFragmentCount,
              dot11MulticastTransmittedFrameCount,
              dot11FailedCount, dot11ReceivedFragmentCount,
              dot11MulticastReceivedFrameCount,
              dot11FCSErrorCount,
              dot11WEPUndecryptableCount,
              dot11TransmittedFrameCount }
       STATUS ~~current~~deprecated
       DESCRIPTION
              "Attributes from the dot11CountersGroup that are not described
              in the dot11MACStatistics group. These objects are mandatory."
::= { dot11Groups 16 }
```

In "Groups - units of conformance" of Annex D, change dot11SMTbase4 as follows:

```
dot11SMTbase4 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,
              dot11RSNAOptionImplemented }
       STATUS ~~current~~deprecated
       DESCRIPTION
              "The SMTbase4 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."
::= { dot11Groups 26 }
```

In "Groups - units of conformance" in Annex D, insert the following objects after dot11SMTbase5 { dot11Groups 30 }:

```
dot11MACbase2 OBJECT-GROUP
       OBJECTS { dot11MACAddress, dot11Address,
              dot11GroupAddressesStatus,
              dot11RTSThreshold, dot11ShortRetryLimit,
              dot11LongRetryLimit, dot11FragmentationThreshold,
              dot11MaxTransmitMSDULifetime,
              dot11MaxReceiveLifetime, dot11ManufacturerID,
              dot11ProductID, dot11CAPLimit, dot11HCCWmin,
              dot11HCCWmax, dot11HCCAIFSN,
              dot11ADDBAResponseTimeout, dot11ADDTSResponseTimeout,
              dot11ChannelUtilizationBeaconInterval, dot11ScheduleTimeout,
              dot11DLSResponseTimeout, dot11QAPMissingAckRetryLimit,
              dot11EDCAAveragingPeriod }
       STATUS current
       DESCRIPTION
              "The MAC object class provides the necessary support for the
              access control, generation, and verification of frame check
              sequences (FCSs), and proper delivery of valid data to upper
              layers."
::= { dot11Groups 31 }

dot11CountersGroup2 OBJECT-GROUP
       OBJECTS { dot11TransmittedFragmentCount,
              dot11MulticastTransmittedFrameCount,
              dot11FailedCount, dot11ReceivedFragmentCount,
              dot11MulticastReceivedFrameCount,
              dot11FCSErrorCount,
              dot11WEPUndecryptableCount,
              dot11TransmittedFrameCount,
              dot11QosDiscardedFragmentCount,
              dot11AssociatedStationCount,
              dot11QosCFPollsReceivedCount,
              dot11QosCFPollsUnusedCount,
              dot11QosCFPollsUnusableCount }
       STATUS current
       DESCRIPTION
              "Attributes from the dot11CountersGroup that are not described
```



```
            in the dot11MACStatistics group. These objects are mandatory."
::= { dot11Groups 32 }

dot11Qosadditions OBJECT-GROUP
      OBJECTS { dot11EDCATable, dot11QAPEDCATable,
            dot11QosCounters }
      STATUS current
      DESCRIPTION
      "This object class provides the objects from the IEEE 802.11 MIB required
      to manage QoS functionality."
::= { dot11Groups 33 }

dot11SMTbase6 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,
            dot11RSNAOptionImplemented,
            dot11RegulatoryClassesImplemented,
            dot11RegulatoryClassesRequired,
            dot11QosOptionImplemented,
            dot11ImmediateBlockAckOptionImplemented,
            dot11DelayedBlockAckOptionImplemented,
            dot11DirectOptionImplemented,
            dot11APSDOptionImplemented,
            dot11QAckOptionImplemented,
            dot11QBSSLoadOptionImplemented,
            dot11QueueRequestOptionImplemented,
            dot11TXOPRequestOptionImplemented,
            dot11MoreDataAckOptionImplemented,
            dot11AssociateinNQBSS,
            dot11DLSAllowedInQBSS,
            dot11DLSAllowed }
      STATUS current
      DESCRIPTION
            "The SMTbase6 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 34 }
```

Annex E

(informative)

Bibliography

E.1 General

Insert the following references in alphanumeric order into E.1:

[B17] "Ethernet, a local area network: Data link layer and physical layer specifications version 1.0." Digital Equipment Corporation, Intel Corporation, Xerox Corporation, September 1980.

[B18] IEEE Std 802.1Q, 2003 Edition, IEEE Standards for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks.[20]

[B19] IETF RFC 2205-1997, Resource ReSerVation Protocol (RSVP)—Version 1 Functional Specification.[21]

[B20] IETF RFC 2212-1997, Specification of the Guaranteed Quality of Service.

[B21] IETF RFC 2215-1997, General Characterization Parameters for Integrated Service Network Elements.

[B22] IETF RFC 2474-1998, Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers.

[B23] IETF RFC 3290-2002, An Informal Management Model for Diffserv Routers.

[B24] ITU-T Z.120 (1999), Series Z: Programming Languages—Formal Description Techniques (FDT)—Message Sequence Chart (MSC).[22]

[20]IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, Piscataway, NJ 08854, USA (http://standards.ieee.org/).

[21]IETF RFCs are available from the IETF Secretariat, c/o Corporation for National Research Initiatives, 1895 Preston White Drive, Suite 100, Reston, VA 20191-5434, USA (http://www.ietf.org).

[22]ITU-T publications are available from the International Telecommunications Union, Place des Nations, CH-1211, Geneva 20, Switzerland/Suisse (http://www.itu.int/).

After Annex J, insert Annex K:

Annex K

(informative)

Admission control

K.1 Example use of TSPEC for admission control

Admission control, in general, depends on vendors' implementations of schedulers, available channel capacity, link conditions, retransmission limits, and the scheduling requirements of a given TSPEC. However, for any given channel capacity, link conditions, and retransmission limits, some TSPEC constructions might be categorically rejected because a scheduler cannot create a meaningful schedule for that TSPEC. There must, for example, be a minimum number of specified fields in the TSPEC in order for the admission control mechanism to create a valid TSPEC. Table K.1 below lists the valid TSPEC parameters that must be present for all admission control algorithms to admit a TSPEC. This represents a set of necessary parameters in order for TSPEC to be admitted; it is not sufficient in and of itself to guarantee TSPEC admittance, which depends upon channel conditions and other factors. Such TSPECs are said to be *admissible*. In the table, S means specified, X means unspecified, and DC means "do not care."

Table K.1—Admissible TSPECs

TSPEC parameter	Continuous time QoS traffic (HCCA)	Controlled-access CBR traffic (HCCA)	Bursty traffic (HCCA)	Unspecified non-QoS traffic (HCCA)	Contention-based CBR traffic (EDCA)
Nominal MSDU Size	S	S	X	DC	S
Minimum Service Interval	S	Nominal MSDU size/mean data rate, if specified (VoIP typically uses this)	Mean data rate/ nominal MSDU size, if mean data rate specified	DC	DC
Maximum Service Interval	S	Delay bound/ number of retries (AV typically uses this)	Delay bound/ number of retries, if delay bound present	DC	DC
Inactivity Interval	Always specified				DC
Suspension Interval	DC				
Minimum Data Rate	Must be specified if peak data rate is specified	Equal to mean data rate	X	DC	DC
Mean Data Rate	S	S	DC	DC	S
Burst Size	X	X	S	DC	DC

Table K.1—Admissible TSPECs *(continued)*

TSPEC parameter	Continuous time QoS traffic (HCCA)	Controlled-access CBR traffic (HCCA)	Bursty traffic (HCCA)	Unspecified non-QoS traffic (HCCA)	Contention-based CBR traffic (EDCA)
Minimum PHY Rate	Always specified				
Peak Data Rate	Must be specified if Minimum Data Rate Specified DC	Equal to Mean Data Rate	DC	DC	DC
Delay Bound	S	S	DC	X	X
Surplus Bandwidth Allowance	Must be specified if the delay bound is present			DC	S
Medium Time	X (not specified by non-AP QSTA; only an output from the HC)				

K.2 Recommended practices for contention-based admission control

K.2.1 Use of ACM (admission control mandatory) subfield

It is recommended that admission control not be required for the access categories AC_BE and AC_BK. The ACM subfield for these categories should be set to 0. The AC parameters chosen by the QAP should account for unadmitted traffic in these ACs.

K.2.2 Deriving medium time

It is recommended that the QAP use the following procedure to derive medium time in its ADDTS Response frame:

There are two requirements to consider: the traffic requirements of the application and the expected error performance of the medium. The application requirements are captured by two TSPEC parameters: Nominal MSDU Size and Mean Data Rate. The medium requirements are captured by two TSPEC parameters: Surplus Bandwidth Allowance and Minimum PHY Rate. The following formula describes how medium time may be calculated (assuming RTS/CTS protection is not used):

Medium Time = Surplus Bandwidth Allowance * pps * MPDUExchangeTime

where:

pps = ceiling((Mean Data Rate / 8) / Nominal MSDU Size)

MPDUExchangeTime = duration(Nominal MSDU Size, Minimum PHY Rate) + SIFS + ACK duration (also see the definition of MPDUExchangeTime in 9.9.3.1.2)

duration() is the PLME-TXTIME primitive that returns the duration of a packet based on its payload size and the PHY data rate employed

K.3 Guidelines and reference design for sample scheduler and admission control unit

K.3.1 Guidelines for deriving service schedule parameters

The HC establishes the SI for each admitted TS for a non-AP QSTA to derive the aggregate minimum SI contained in the non-AP QSTA's service schedule. The SI for each TS is equal to a nonzero minimum SI contained in the TSPEC, if it exists; otherwise, it is the nominal MSDU size divided by the mean data rate. The SI contained in the service schedule is equal to the smallest SI for any TSPEC.

The HC may use an aggregate "token bucket specification" to police a non-AP QSTA's admitted flows. The HC must derive the aggregate mean data rate and aggregate burst size to establish the aggregate token bucket specification. The aggregate mean data rate is equal to the sum of the mean data rates of all of the non-AP QSTA's admitted TSs. The aggregate burst size is equal to the sum of the burst size of all of the non-AP QSTA's admitted TSs. An aggregate token bucket is initialized with the aggregate burst size. Tokens are added to the token bucket at the aggregate mean data rate.

K.3.2 TSPEC construction

TSPECs are constructed at the SME from application requirements supplied via the SME and with information specific to the MAC layer. There are no normative requirements on how any TSPEC is to be generated. However, in this subclause a description is given of how and where certain parameters may be chosen. The following parameters typically arise from the application: Nominal MSDU Size, Maximum MSDU Size, Minimum Service Interval, Maximum Service Interval, Inactivity Interval, Minimum Data Rate, Mean Data Rate, Burst Size, Peak Data Rate, and Delay Bound. The following parameters are generated locally within the MAC: Minimum PHY Rate and Surplus Bandwidth Allowance, although the Maximum Service Interval and Minimum Service Intervals may be generated within the MLME as well. This subclause describes how the parameters that are typically generated within the MAC may be derived.

Note that a TSPEC may also be generated autonomously by the MAC without any initiation by the SME. However, if a TSPEC is generated subsequently by the SME, the TSPEC generated autonomously by the MAC shall be overridden. If one or more TSPECs are initiated by the SME, the autonomous TSPEC, containing the same TSID, shall be terminated.

Typically, TSPEC parameters not determined by the application are built upon the assumption that the following exist:

— A probability p of not transmitting the frame (because it would have exceeded its delay bound)
— An MSDU length (which can be considered fixed for constant-bit-rate applications)
— Application throughput and delay requirements
— A channel model of error, in particular a channel error probability for the (fixed) frame length
— Possibly country-specific limits on TXOP limits

The minimum service interval, if determined within the MAC, may typically be given as the nominal MSDU size/mean data rate.

The maximum service interval, if determined within the MAC, may be calculated as the delay bound/number of retries possible. This number should be greater than the minimum SI, when that is specified. The number of retries may be chosen (as below) to meet a particular probability of dropping a packet because it exceeds its delay bound. Note that for multiple streams, this SI should be the aggregate of all SIs requested, because the QSTA is assigned the TXOPs, not any particular stream.

Typically, it can be assumed that the scheduler would attempt to schedule TXOPs distributed throughout a small multiple of beacon intervals (if not a single beacon interval). In addition, TXOP limits would typically be chosen to be as short as possible (within the constraints of the minimum PHY rate, acknowledgment policy, and so forth), consistent with the goal of maximizing throughput. In other words, because of overhead, not to mention the requirements for transmitting a single Poll frame, MPDU, and possibly ACK frame, the TXOPs need to be at least of a certain duration.

The channel model implies an error rate and an assumption about dependency (joint probability distribution of channel errors sequentially, i.e., burst error probabilities).

For example, if the channel causes errors independently from frame to frame and the error probability is the same for all frames of the same length at all times, this channel would be said to be an independent, identically distributed error channel. With p as the probability of dropping the frame, and p_e as the probability of the frame not being transmitted successfully (i.e., either the data frame or the ACK frame associated with it is in error), let N_p be the number of retries required to maintain the probability of dropping the frame to be p.

The probability of any given packet being dropped in such a channel after N_p retries is given by

$$p_{\text{drop}} = (p_e)^{N_p + 1}$$

For example, in such a channel, if $p_e = 0.1$ and $p_{\text{drop}} = 10^{-8}$, then up to seven retries within the delay bound are required, and the scheduler should ensure that sufficient cumulative TXOP allocations are made to accommodate retransmissions.

The Surplus Bandwidth Allowance parameter ensures the requesting QSTA is allocated a minimum amount of excess time by the scheduler to so that application dropped packet rates are bounded. For example, this parameter can be chosen to ensure that when there is a 10% packet error rate (PER) for 1000-octet packets, that there is a dropped PER (i.e., packets that fail to be received within the delay bound) of 10^{-8}. This parameter may be chosen based on an initial assumption regarding channel/source characteristics and renegotiated by sending ADDTS Request frames if required, based on actual transmission behavior. To understand how this parameter may be specified, consider the case where there are only 100 PPDUs to be transmitted and delay is not an issue. The PER p_e is 10%, with the errors happening independently from packet to packet. To accomplish this, the number of packets transmitted in each beacon interval must exceed the 100 PPDUs by N_{excess} in order to avoid dropping packets with some fixed probability (denoted as p_{drop}). For example, if $p_{\text{drop}} = 10^{-8}$, then the number of retries N_{excess} must satisfy to send *only* 100 packets successfully (based on Bernoulli distributed error probabilities):

$$p_{\text{drop}} = \sum_{k = N_{\text{excess}}}^{N_{\text{excess}} + 100} \binom{N_{\text{excess}} + 100}{k} p_e^k (1 - p_e)^{100 + N_{\text{excess}} - k}$$

where $p_e = 0.1$. Solution of an equation such as this yields the total number of additional retries. This may be found, for this example, using the fact that

$$\sum_{k = a}^{b} \binom{b}{k} p_e^k (1 - p_e)^{b - k} = I_{p_e}(a, b - a + 1)$$

where $I_{p_e}(a, b)$ is the Incomplete Beta function, and taking n sufficiently large (or invoking the law of large numbers). Solving this yields that, on the average, 38 additional MPDUs are required to keep the probability of dropping a packet to less than 10^{-8} to send only 100 packets. For this case, the

Surplus bandwidth allowance = $\frac{100 + N_{\text{excess}}}{100}$, which for this example would be 1.38.

This might represent an upper bound for the excess bandwidth for many applications: it presumes that the observation interval is 100 + N_{excess} frames. When the observation interval (or delay bound) is longer than the time it takes to transmit 100 frames, then it can be shown that the excess bandwidth required decreases. For example, if it were desired to send 100 000 frames with 12 000 frames excess, then the probability of a dropped frame becomes 1.6×10^{-15}.

On the other hand, suppose that an infinitely long stream was to be transmitted without any constraints on delay. In such a case, with an infinite number of retries and with a 10% PER, 1.111 times the bandwidth required to send MPDUs without error is required (because the probability that *n* retries are required for any packet is given by 0.1^n).

In fact, assuming a finite delay bound, the above result (1.111= surplus bandwidth allowance) represents a lower bound on what the surplus bandwidth allowance would be.

Typically, then the excess bandwidth required would be between a "send only *N* + excess packets" scenario within a given delay and "send an infinite number of packets with an infinite number of retries with no delay" scenario.

A more exact calculation can be done via simulation as follows: Suppose there are $N_{\text{allocated}}$ constant length PPDUs per beacon interval (these are actually transmitted on the air), and suppose there are N_{payload} constant length PPDUs to be transmitted. Suppose further that these transmitted and payload PPDUs arrive in a uniformly distributed manner in each beacon interval, then the delay incurred in waiting for a packet to be transmitted may be inferred from examining the transmit queue length and statistically may be inferred from examining how many retries within a certain period of time are required to keep the probability of a dropped packet below a certain amount. The number of retries (and consideration of IFS, polls, etc.) then determines the surplus bandwidth allowance.

In general, then the surplus bandwidth allowance may be given by $\frac{N_{\text{allocated}}}{N_{\text{payload}}}$.

Note that for the case of a Block Ack, a similar calculation applies, although the calculations for the excess bandwidth need to take into account the probability of failing to receive a Block Ack, and so forth.

K.3.3 Reference design for sample scheduler and admission control unit

This subclause provides the guidelines for the design of a simple scheduler and admission control unit (the unit that administers admission policy in the HC SME) that meet the minimum performance requirements as specified in 9.9.3.2. The scheduler and admission control unit use the minimum set of mandatory TSPEC parameters as specified in 9.9.3.2.

K.3.3.1 Sample scheduler

This subclause includes the reference design for a sample scheduler. This scheduler uses the mandatory set of TSPEC parameters to generate a schedule: Mean Data Rate, Nominal MSDU Size, and Maximum Service Interval or Delay Bound. If both Maximum Service Interval and Delay Bound parameters are specified by the non-AP QSTA in the TSPEC, the scheduler uses the Maximum Service Interval parameter for the calculation of the schedule.

The schedule generated by the scheduler meets the normative behavior as specified in 9.9.3.2. The schedule for an admitted stream is calculated in two steps. The first step is the calculation of the scheduled SI. In the second step, the TXOP duration for a given SI is calculated for the stream.

The calculation of the scheduled service interval is done as follows: First, the scheduler calculates the minimum of all maximum SIs for all admitted streams. Let this minimum be *m*. Second, the scheduler chooses a number lower than *m* that is a submultiple of the beacon interval. This value is the scheduled SI for all non-AP QSTAs with admitted streams. See Figure K.1.

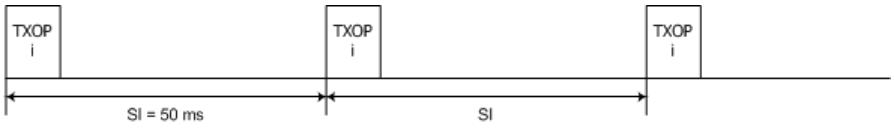


Figure K.1—Schedule for stream from QSTA *i*

For the calculation of the TXOP duration for an admitted stream, the scheduler uses the following parameters: Mean Data Rate (ρ) and Nominal MSDU Size (*L*) from the negotiated TSPEC, the Scheduled Service Interval (*SI*) calculated above, Physical Transmission Rate (*R*), Maximum Allowable Size of MSDU, i.e., 2304 bytes (*M*), and Overheads in time units (*O*). The physical transmission rate is the minimum PHY rate negotiated in the TSPEC. If the minimum PHY rate is not committed in the ADDTS Response frame, the scheduler can use the observed PHY rate as *R*. The overheads in time includes IFSs, ACK frames and CF-Poll frames. For simplicity, details for the overhead calculations are omitted in this description. The TXOP duration is calculated as follows: First, the scheduler calculates the number of MSDUs that arrived at the mean data rate during the SI:

$$N_i = \left\lceil \frac{SI \times \rho_i}{L_i} \right\rceil$$

Then the scheduler calculates the TXOP duration as the maximum of

— Time to transmit N_i frames at Ri and

— Time to transmit one maximum size MSDU at R_i (plus overheads):

$$TXOP_i = max(\frac{N_i \times L_i}{R_i} + O, \frac{M}{R_i} + O)$$

An example is shown in Figure K.1. Stream from QSTA *i* is admitted. The beacon interval is 100 ms and the maximum SI for the stream is 60 ms. The scheduler calculates a scheduled SI (SI) equal to 50 ms using the steps explained above is this annex.

The same process is repeated continuously while the maximum SI for the admitted stream is larger than the current SI. An example is shown in Figure K.2.

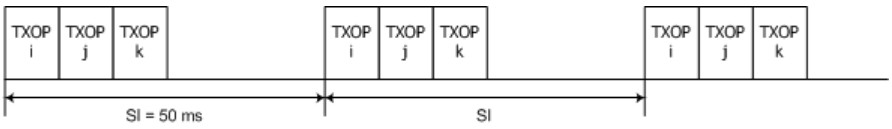


Figure K.2—Schedule for streams from QSTAs *i* to *k*

If a new stream is admitted with a maximum SI smaller than the current SI, the scheduler needs to change the current SI to a smaller number than the maximum SI of the newly admitted stream. Therefore, the TXOP duration for the current admitted streams needs also to be recalculated with the new SI.

If a stream is dropped, the scheduler might use the time available to resume contention. The scheduler might also choose to move the TXOPs for the QSTAs following the QSTA dropped to use the unused time. An example is shown in Figure K.3, when the stream for QSTA *j* is removed. However, this option might require the announcement of a new schedule to all QSTAs.

Figure K.3—Reallocation of TXOPs when a stream is dropped

Different modifications can be implemented to improve the performance of the minimum scheduler. For example, a scheduler might generate different scheduled SIs (SI) for different QSTAs, and/or a scheduler might consider accommodating retransmissions while allocating TXOP durations.

K.3.3.2 Admission control unit

This subclause describes a reference design for an admission control unit (ACU) that administers admission of TS. The ACU uses the same set of parameters that the scheduler uses in K.3.3.1.

When a new stream requests admission, the admission control process is done in three steps. First, the ACU calculates the number of MSDUs that arrive at the mean data rate during the scheduled SI. The scheduled SI (SI) is the one that the scheduler calculates for the stream as specified in K.3.3.1. For the calculation of the number of MSDUs, the ACU uses the equation for N_i shown in K.3.3.1. Second, the ACU calculates the TXOP duration that needs to be allocated for the stream. The ACU uses the equation for $TXOP_i$ shown in K.3.3.1. Finally, the ACU determines that the stream can be admitted when the following inequality is satisfied:

$$\frac{TXOP_{k+1}}{SI} + \sum_{i=1}^{k} \frac{TXOP_i}{SI} \leq \frac{T - T_{CP}}{T}$$

where

k is the number of existing streams
$k + 1$ is used as index for the newly arriving stream
T indicates the beacon interval
T_{CP} is the time used for EDCA traffic

The ACU needs to ensure that it complies with the dot11CAPlimit, i.e., the scheduler does not allocate TXOPs that exceed dot11CAPlimit. The ACU might also consider additional time to allow for retransmissions.

The ACU ensures that all admitted streams have guaranteed access to the channel. Any modification can be implemented for the design of the ACU. For example, UP-based ACU is possible by examining the UP field in TSPEC to decide whether to admit, retain, or drop a stream. If the UP is not specified, a default value of 0 is used. If a higher UP stream needs to be serviced, an ACU might drop lower UP streams.